Doctoral Dissertation
Doctoral Program in Management and Production Engineering (35th Cycle)

A Deep Learning and Augmented Reality Framework for 3D human-machine interaction

By

Leonardo Tanzi

Supervisor(s):
Prof.ssa Federica Marcolin, Supervisor

Doctoral Examination Committee:
Prof. Motyl Barbara, Università degli Studi di Udine
Prof. Speranza Domenico, Università degli Studi di Cassino
Prof. Baronio Gabriele, Università di Brescia
Prof. Gloria Antonio, Università degli Studi di Napoli Federico II
Prof. Montrucchio Bartolomeo, Politecnico di Torino

Politecnico di Torino
2022

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.



Leonardo Tanzi

2022

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgment

And I would like to acknowledge the wonderful research group I’ve had the honor to work with in Turin and Prof. Grandvalet for hosting me in its lab in Paris

Abstract

In the context of Industry 4.0, human-machine interaction (HMI) is a multidisciplinary field of study focusing on the design of computer technology and the interaction between humans (the users) and computers. In recent years, Artificial Intelligence (AI) and particularly Deep Learning (DL) has introduced huge improvements in the field. The same applies to Augmented Reality (AR), a technology that offers the possibility to build intuitive and immersive interfaces between the user and the machine, creating tools and methods for the approach for augmented humans. In different complex situations, a user, both expert and beginner, could improve his performance with the help of an assistance application. Following these needs, this study focuses on building an adaptable and general-purpose framework for assistance that could be applied in different areas, such as the biomedical and the industrial field, to underline the scalability of this approach. The proposed framework is composed of two layers: the first one focuses on the DL analysis of the data, while the second on the AR assistance.

After a brief overview, we focused on a literature review of augmented human approaches to critically analyze the state-of-the-art for identifying the baseline of DL and AR frameworks, discussed in Chapter 1.

In Chapter 2, we described the first layer, which comprises two projects related to classification. Concerning the biomedical area, we worked on classification DL algorithms to analyze images related to anatomical human regions, i.e., bone X-Rays. This research was carried out in collaboration with the A.O.U Città della Salute e della Scienza of Turin. In the industrial setting, we worked on DL algorithms to exploit 2D/3D data to support the classification of human emotions in collaboration with the Polytechnic University of Milan.

In Chapter 3, we presented the results of these three years of research, i.e., the second layer of the framework and the consequent two fully working pipelines. The first project focuses on computer-assisted prostatectomy, while the second on computer-assisted troubleshooting. The resulting final framework composes the

two layers to build a computer-assistance tool that analyzes the video stream and returns 3D information to augment it.
Finally, Chapter 4 presents a discussion and concludes this research.

Contents

Introduction 1

1. Literature Review 2

1.1 Deep Learning in Computer Assisted Healthcare..... 2

1.1.1. Fracture Classification..... 2

1.1.2 Computer assisted surgery in the IOR (Intelligent Operating Room) 12

1.2 Deep Learning in Computer Assisted Troubleshooting .. 25

2. First Layer 29

2.1 Femur Fracture Classification: Overview..... 29

2.1.1 First approach: Cascade Classifier..... 31

2.1.2 Second approach: Vision Transformer..... 41

2.1.3 Third approach: Hierarchies..... 54

2.2 Emotion Classification..... 69

3. Second Layer and Full Framework 75

3.1 Assisted Surgery: overview 75

3.1.1 First Approach: Segmentation CNN 77

3.1.2 Second Approach: RotationCNN..... 90

3.1.3 Blood Accumulation 106

3.2 Assisted 3D Printing 114

4. Discussion and Conclusions 129

References 131

List of Figures

Figure 1 Project 1 (left) and 3 (right) brief graphical description.....3

Figure 2 Project 2 and 4 divided into first and second layer.....4

Figure 3 Pipeline of the architecture proposed by Dimililer [12]. 4

Figure 4 Faster-RCNN architecture [21].....6

Figure 5 Pre-processing phases in [22] 8

Figure 6 Heatmap generated by Grad-CAM [24] 9

Figure 7 Neer's classification for proximal humerus [27] 9

Figure 8 Examples of Pre-Operative and Post-Operative phases. In the Pre-Op phase (a), Electronic Medical Record (EMR) and imaging can predict the risk score. In the Post-Op phase (b), EMR, vital information and video analysis can improve care management. 13

Figure 9 PRISMA Flowchart 14

Figure 10 Review Pipeline (only the principal objectives are shown).....15

Figure 11 Tool Presence (a), Tool Localization (b) and Tool Segmentation (c). 17

Figure 12 An example of pose estimation.....18

Figure 13 The Da Vinci robot manipulators. 21

Figure 14 Conceptualization of an IOR. 25

Figure 15 AO/OTA classification. 30

Figure 16 Some samples of real X-Rays images. These samples were used for training the neural network after the cleaning and cropping phase 31

Figure 17 STARD 2015 Flow Diagram to define how the dataset is decomposed. 32

Figure 18 Flow chart for the three classes classification case. Two approaches were presented: a classic CNN for classification used as baseline and a multistage one characterized by subsequent binary networks.....33

Figure 19 ROC curve with associated AUC (A, C) and confusion matrix (B, D) are shown for both three and five classes classification. The number of images used for testing was: 226 Unbroken images, 150 B, 114 A, 56 A1, 36 A2 and 21 A3. 36

Figure 20 Full pipeline adding a new step of classification. 38

Figure 21 Grad-CAM output for A and B type fractures classification. 39

Figure 22 The STARD 2015 Flow Diagram, which shows the dataset processing workflow 44

Figure 23 Confusion matrices and distributions of the different clustering approaches. They were obtained with a Convolutional Autoencoder applied directly on the images (a) or with an Autoencoder applied to the feature vector extracted from the CNN (b) and extracted from the ViT Encoder (c). In the right images, the cluster labels are shown with the colors presented in the sidebar 50

Figure 24 Full pipeline of this work. The images were first passed through a YOLOv3 network which cropped the areas related to the left and right femur. A CNN and a hierarchical CNN were then applied to this dataset and the results were used as baselines. Then, a modified ViT encoder was used for 7-class classification and the attention maps were analyzed. Finally, clustering was performed to evaluate the ability of the ViT to extract features 52

Figure 25 Some examples of original images and corresponding attention map for each class. The area in which ViT is focusing during inference is highlighted in white 53

Figure 26 Example of a class structure with parameter decomposition 59

Figure 27 Aggregation matrices computed during the topdown and bottom-up scans of the tree 60

Figure 28 Misclassification cost (top) and hierarchical misclassification cost (bottom), in %, for the three benchmarks and the three network architectures; the x-axis represents the number of images per class used for training. Each line style corresponds to a network architecture (dashed: InceptionV3; solid: ResNet50; dotted: ViT-B16). The baseline algorithm, trained with standard cross-entropy loss, is in red; the algorithm trained with hierarchical loss is in blue. 66

Figure 29 Principal plane of the Linear Discriminant Analysis performed on the fine-tuned representations learned by cross-entropy (top) and the hierarchical loss (bottom) for InceptionV3, ResNet50 and ViT-B16. For readability and fidelity of the 2D representations, the analyses focus on the 10 classes of two superclasses of CIFAR100. Each symbol and color corresponds to a superclass (green circle for “fish” and blue triangle for “household devices”), and each hue corresponds to a class. 68

Figure 30 Scheme of the overall architecture, composed of a 2D RGB-Net and a 3D Depth-Net. The two networks can work independently (2D only and 3D only), or be concatenated by an additional layer and work together (Multimodal 2D+3D) 72

Figure 31 Confusion matrix and corresponding precision, recall, and f1-score values for each class, where precision is the ratio of correctly predicted positive observations to the total predicted positive observations, recall is the ratio of correctly predicted positive observations to the all observations in the actual class, and f1-score is the weighted average of precision and recall. 73

Figure 32 Samples taken from video A, B, C, D and E videos.....80

Figure 33 The bounding polygon of the catheter. The upper black dot marks the anchor point, while the white one marks the centroid. The direction arrow was used to calculate the Z rotation and the two sections of the area divided by the centroid to compute the X rotation 82

Figure 34 The full pipeline of our method. The original image (a) is segmented (b) and, after the shape detection (c), the 3D model is projected over the 2D image (d).....83

Figure 35 Comparison between the pipeline of the CV based approach from our previous work (178) and the method proposed on this paper based on DL. It is possible to notice that the approach based on CV predicted a wrong anchor point and slightly worse rotations X and Z.....88

Figure 36 Some segmentation masks returned by our network and the consequent error of the 3D overlay. It is possible to notice how the network performs good even in borderline situations: when the catheter is partially occluded (a), when a tool is featured in the foreground (b) and when the camera is far from the scene (c) 89

Figure 37 Schematic illustration of the system workflow. The video frame is fed to a SegmentationCNN and a RotationCNN, depending on the Case Study. Target's position and scale are obtained through the SegmentationCNN for every Case Study. Target's orientation is retrieved in different ways depending on the Case Study: RotationCNN output the predicted rotation for each frame (Case Study 1); RotationCNN for registration in an initial frame and OF tracking for next frames (Case Study 2); Manual Registration in an initial frame and OF tracking for next frames (Case Study 3).....94

Figure 38 Rotation Neural Network overall architecture, which involves a modified version of the ResNet50 model, with a branch for each axis rotation....98

Figure 39 Sample images referring to original endoscopic images, segmentation masks and mask overlay (columns) for each case study (rows): catheter (C), prostate (P) and kidney (K).....102

Figure 40 Trends representation of organ orientation on X, Y and Z axes comparing tagged (blue) and OF estimated (red) rotation values for each sample frame. 103

Figure 41 Blood residue on tissues (a) vs. Active bleeding (b) 108

Figure 42 Multi-tasking neural network architecture (example for 160x240 input image). Each box represents a multi-channel feature map. The number of channels is provided on top of the box, while the x-y size is shown at the lower left edge. Blue boxes belong to the backbone, green boxes refer to the first branch for semantic segmentation, grey boxes denote copied feature maps (skip connections), and purple boxes represent the second branch for event detection. The arrows refer to the different operations. 110

Figure 43 . Sample imaged referring to the multi-tasking neural network outputs, namely the semantic segmentation masks for tools (colored in blue) and the percentage of accumulated blood if greater than 50%.....113

Figure 44 Full pipeline of the proposed approach. The frame is passed to an object detection algorithm which returns the cropped area related to the detected 3D printer and the position and scale information. Its output is passed to a CNN, which predicts the rotation values, according to the coordinates system shown in the figure. This information is then combined to correctly overlay the 3D model to the 2D stream with specific instructions and highlighted components. However, for clarity, in this example is shown the whole 3D model instead of a specific highlighted part.....118

Figure 45 Architecture of the RotationCNN. The network takes as input the RGB images, which are passed through the first layers of ResNet50. The last layer was substituted with three branches composed of a Dense layer, a Batch Normalization layer, a Dropout layer, and a final Dense layer with *n* neurons, where *n* is the range of the rotation along the specific axis.....119

Figure 46 Precision/recall curve with IoU thresholds of 0.5 (a), 0.6 (b) and 0.7(c). The AP is shown in light blue as the area under the curve 122

Figure 47 In the specific operation of filament replacement four components of the Prusa i3 MKS are involved: the Screen and the Button for steps 1, 2, 3, 4, and 8, the Filament for steps 5 and 7, the Coil for step 6 and the Extruder for steps 9 and 10. These components are highlighted during the corresponding step.....125

Figure 48 The ten steps of the filament substitution procedure. In each step, the specific component involved in the current precision was highlighted in orange, while the other components were kept in grey. The text instructions were the same defined above.....126

List of Tables

Table 1 Projects we conducted divided into two different layers 2

Table 2 Summary of objectives and related input, neural network and output. 24

Table 3 Comparison between three and five classes classification for InceptionV3, VGG16 and ResNet50. 34

Table 4 Precision, Recall and F1-score for three and five classes classification using the InceptionV3 network. The number of images for testing is proportioned to the initial dataset unbalance.....35

Table 5 Accuracy using 5-fold cross validation for the three networks used in the different stages.....36

Table 6 Precision, Recall and F1-score for three and five classes classification using the multistage approach. The number of images is proportioned to the initial dataset unbalance. Improvements for five classes classification are shown in parenthesis. 37

Table 7 Precision, Recall and F1-score for three and five classes classification using the multistage approach with 4 stages. Values for five classes classification increased, especially for A3, compared to the three stages approach, as shown in parenthesis. 38

Table 8 Cohen's Kappa scores to measure inter-agreement. Values between 0.41 and 0.60 indicate a moderate agreement, between 0.81 and 0.80 a substantial agreement and between 0.81 and 0.99 an almost perfect agreement.....40

Table 9 Baseline characteristics. Median computed with Interquartile Range (IQR) 43

Table 10 Values of precision, recall, and F1-score for the Compact Convolution Transformer and the base-16, base-32, large-16, large-32 configurations. The values are shown with related confidence intervals. 45

Table 11 Values of precision, recall, and F1-score for the two baselines (a) and (b) and for the ViT (c). The values are shown with related confidence intervals. In (c), the improvement given by the ViT compared to the highest value among the two baselines is shown in parenthesis 48

Table 12 Values of accuracy, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and loss for the three clustering approaches 49

Table 13 Values of accuracy for the 11 specialists who performed the evaluation with and without the CAD (Computer Assisted Diagnosis) system. The average values are shown with related Confidence Interval (CI) 51

Table 14 Macro-averaged F1-score (with standard deviation) with standard fine-tuning using cross-entropy and weight decay (Flat FT) and with fine-tuning using the hierarchical loss and regularization (Hierarchical FT), on the femur fracture dataset. 67

Table 15 Results training with A and testing with A, B, C, and D. Catheter IoU (Cat IoU) and Mean IoU for the available combination of architecture and base network, training with 235 frames taken from A video and testing with 50 frames taken from A, B, C, D. It is also shown the number of iterations per second for each combination. One iteration consists in predicting the segmentation output given an input image. 84

Table 16 Training and testing of U-Net architecture with MobileNet as the base network to generalize the results. We trained the network with A video and two videos among B, C, D, and tested it with the remaining one. In the first row, the results for the baseline are reported again to make the comparison valid and improvements are shown in parenthesis. In the last row are shown results given from training and testing the network with all the videos. 85

Table 17 Comparison between the proposed approach based on Deep Learning (DL) and the former approach based on Computer Vision (CV) techniques for segmentation. Standard deviation σ is also shown in the rightmost column. 85

Table 18 Comparison between the proposed approach based on Deep Learning (DL) and the former approach based on Computer Vision (CV) techniques for the 3D overlay performances with standard deviation σ 86

Table 19 Values referring to the images used for datasets creation, both synthetic and real, for segmentation and rotation CNNs. For the latter, considered rotation ranges are also shown. For the catheter, the RotationCNN was trained to predict only the X rotation, as Y was derived directly from the segmentation map and Z was considered irrelevant. 101

Table 20 IoU scores referring to the mean value of the actor classes (background, tool and target) and the target class only: catheter (C), prostate (P) and kidney (K). 101

Table 21 Values of mean accuracies when testing with a synthetic dataset, a real dataset and the same real dataset after fine-tuning the network. Each value is computed for the considered target classes: catheter (C), prostate (P) and kidney (K). The accuracies for X, Y and Z, were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: prediction with an error in the range [-5, +5] and in the range [-10, +10]. *Y-Axis rotation values for C were retrieved from the semantic segmentation CNN. 103

Table 22 Values of mean accuracies together geodesic distance. All values are shown with related standard deviation. The accuracies were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: exact prediction, with an error in the range [-5, +5] and in the range [-10, +10], while the geodesic distance is computed as the positive amount corresponding to the length of the geodesic arc connecting the two rotation's angles. It/s are iterations per second, where one iteration is considered as a forward pass in the full pipeline 122

Table 23 Values of accuracies for each axes together with mean accuracy and geodesic distance. All values are shown with related standard deviation for tests performed with a synthetic dataset, a real dataset, and the same real dataset after fine-tuning the network. The accuracies for X, Y, and Z, were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: exact prediction, with an error in the range [-5, +5] and in the range [-10, +10], while the geodesic distance is computed as the positive amount corresponding to the length of the geodesic arc connecting the two rotation's angles 123

Table 24 Values of speed (in minutes) and a number of actions performed by the ten non-expert users with the help of the user manual or with the help of our CAD system. In the two colored columns are showed the means and standard deviations for each distribution.....126

Introduction

In the context of Industry 4.0, human-machine interaction (HMI) is a multidisciplinary research topic concentrating on the design of computer technology and, in particular, the interaction between humans and automated systems. This no longer just applies to conventional industrial machinery but also to computers, digital systems, or anything connected to the Internet of Things (IoT). Among the many characteristics an HMI system should have, the most important two are undoubtely automation and intuitiveness [1].

1) **Automation:** Concerning the first one, doing things automatically is not a trivial task. Fortunately, in recent years, Artificial Intelligence [2] and, in particular, Deep Learning (DL) [3], a subsection of Machine Learning, have introduced huge improvements in the field. Machine learning is the study of algorithms that use real-world experience to identify and extract patterns from large amounts of unstructured data. Through experience, these algorithms naturally get better. More precisely, machine's performance increases over time by learning a particular task, without being explicitly programmed to do so. Deep learning is a technique of machine learning able to solve intricate problems, subdividing complex concepts into simpler ones and representing them with a nested hierarchy of tasks with different levels of abstraction the computer can efficiently perform. This philosophy makes these systems able to operate in complicated real-world environments. Deep learning exploits artificial neural networks with multiple layers between the input and the output layer for learning. A neural network is composed of input, hidden, and output layers, all of which are composed of nodes. All these subsequent layers define a function with thousands or even millions of parameters. In a typical classification problem, the network is fed with the input images and tries to assign a specific pre-defined class to each of them. After the prediction, it calculates a loss function (how much the predicted classes differ from the original) and adjusts its parameters with gradient descent and backpropagation. The images are fed at different times through the network back and forth until the network gets a certain accuracy [4].

2) **Intuitiveness:** The second necessary quality of an HMI system is intuitiveness. Indeed, these tools must be simple to use and shouldn't put too much strain on the user. This could be obtained with the use of Augmented Reality (AR), a technology that offers the possibility to build an intuitive and immersive interface between the user and the machine, creating tools and methods for the approach for augmented humans. AR enhances the real world by affixing layers of digital elements onto it. These elements could include haptic feedback, computer-generated visuals, sound or video effects, or sensory projects. The goal of including this digital data is to give customers a dynamic and engaging experience that is made possible by inputs from various technology, such as smartphones, smart glasses, and smart lenses. It is worth mentioning that AR differs from Virtual reality (VR). The primary distinction between the two is that, as opposed to AR, which is used in direct view of an existing actual environment and adds components like audio, films, or graphics, VR completely replaces the real environment with an artificial one.

By leveraging these characteristics given by DL and AR, in different complex situations, a user, both expert and beginner, could improve his performance with the help of an assistance application.

For this reason, we wanted to build an adaptable and general-purpose framework that could be applied in different areas, such as the biomedical and the industrial field, to underline the scalability of this approach.

First of all, in Chapter 1 we worked on a literature review of augmented human approaches to critically analyze the state-of-the-art for identifying the baseline of DL and AR frameworks. After understanding the principal aspects of the technology, we defined a general framework.

Table 1 Projects we conducted divided into two different layers

	Biomedical	**Industrial**
Layer 1	**1.** X-ray classification **2A.** Endoscopic video segmentation	**3.** Emotion classification **4A.** Solid object detection
Layer 2	**2B.** 3D augmented surgery	**4B.** 3D augmented troubleshooting

The proposed framework is composed of two layers: the first one focuses on the DL analysis of the data, while the second on the AR assistance. In total. We tackled four projects: two of them comprises just the first layer (project 1 and 3 in Table 1, discussed in Chapter 2), while the other two (project 2A and 4A in Table 1) were extended to the full framework by adding a second layer (project 2B and 4B in Table 1, discussed in Chapter 3).

Layer 1

Project 1 (Figure 1, left) discusses the classification of femur fractures, which follows a hierarchical structure. Three different steps are presented as we evolved in the research. First, we developed a cascade of Convolutional Neural Network (CNN), each tackling one level of classification, described in Section 2.1.1. We then decided to change the architecture and apply a Vision Transformer to the same task in Section 2.1.2. Finally, we focused on the general concept of hierarchy to improve the results of our work, explained in Section 2.1.3. In order to have more degree of freedom in the management of the most critical issue of this layer, we involved the Sorbonne UTC AI laboratory, managed by Prof. Y. Grandvalet, the deputy director of Heudiasyc, specialized in DL for computer vision, spending a 7-month period in the research group in France. The whole work was carried out in collaboration with the CTO of Turin.

Concerning the industrial setting, Project 3 (Figure 1, right), in collaboration with the Polytechnic University of Milan, deals with the implementation of a classifier able to recognize the 6 basic emotions (anger, disgust, fear, happiness, sadness, and surprise). The classifier was trained in a multimodal modality with 2D and 3D data.

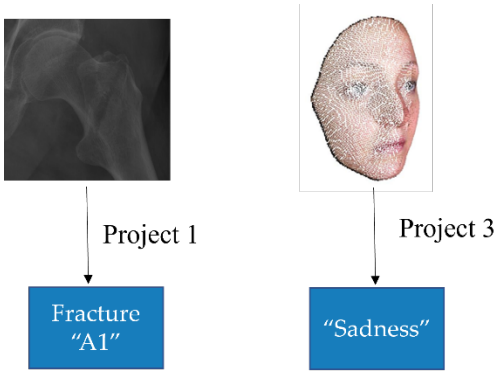


Figure 1 Project 1 (left) and 3 (right) brief graphical description

Layer 2

If the first two projects helped us to develop a general knowledge of DL algorithms, in the last two projects (Projects 2 and 4), we presented the full framework. The first is related to computer-assisted surgery, while the second is related to computer-assisted troubleshooting. Each of those was divided into two layers, shown in Figure 2.

In the first layer, a segmentation network (Project2A) or an object detection network (Project4A) was used to recognize the different actors in the video stream, while the second layer leveraged AR to offer an intuitive interface.

More specifically, Project 2B overlays a 3D model of the organ of the patient into the video stream together with rotation parameter. In Project 4B, the model is overlayed together with textual instruction of the operations to be performed.

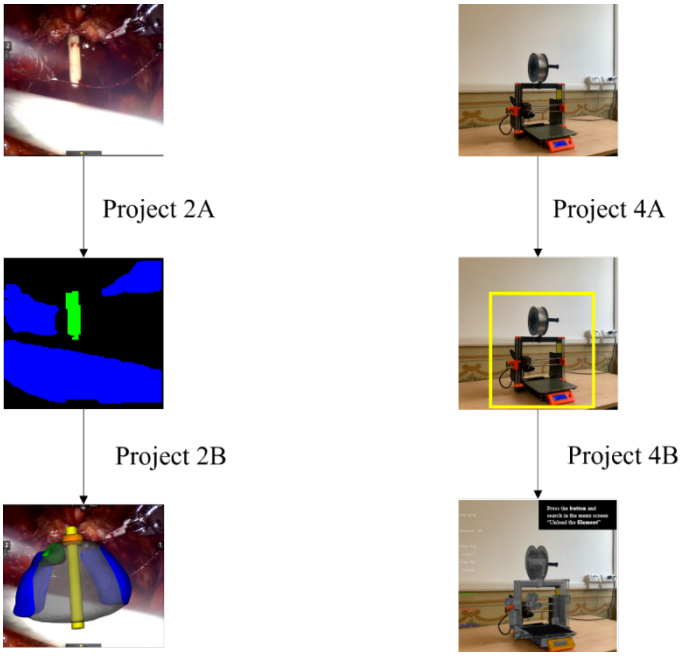


Figure 2 Project 2 and 4 divided into first and second layer

Finally, in Chapter 4, we discussed and concluded our work, describing also the limitations and the possible future works.

Chapter 1

Literature Review

1.1 Deep Learning in Computer Assisted Healthcare

Concerning the medical part of this work, we first performed a literature review on the two main projects we carried out. The first review is related to fracture classification while the second on computer assisted surgery.

Part of these works has already been published in [5,6].

1.1.1. Fracture Classification

Introduction

In the last decades, several medical procedures have improved year by year with the help of technology [7,8]; among these, great progress has been made in the orthopedic field. Bones fractures are one of the most common injuries nowadays. Every year 2.7 million fractures occur across the EU6 nations, France, Germany, Italy, Spain, Sweden, and the UK [9], an incredible number of people suffer from this disorder, and the implications of an untreated fracture may lead to permanent damages or even death. A great responsibility in this lies with the doctors, who have to evaluate tens of X-Ray images a day. The technology utilized for first diagnosis is mostly X-Ray, which is a modality used for more than one hundred years and is still frequently utilized. It is challenging for doctors to evaluate X-Ray images: firstly, X-Ray could hide certain particularities of bone; secondly, a long experience is needed to correctly classify different types of fractures; thirdly, doctors have often to act in emergency situations and may be constrained by fatigue. Actually, it has been shown that the performance of radiologists in the interpretation of musculoskeletal radiographs decreases in fracture detection at the end of the work day compared to the beginning of the work day [10]. In addition, radiographic interpretation often takes place in environments without the availability of qualified colleagues for second opinions [11]. The success of the treatment and prognosis strongly depends on an accurate classification of the fracture among standard types,

such as those defined by the Arbeitsgemeinschaft fur Osteosynthesefragen (AO-foundation). In that context, computed-aided diagnosing system able to help doctors might have a direct impact on the outcome of the patients. In this section, we review some selected papers concerning this topic, starting from basic approaches to the main advanced solutions. To the best of our knowledge, a work that tries to define an ideal method to classify fractures valid for each bone in the human body does not exist. In our opinion, the best way to pursue this is to evaluate different papers and select the strengths which could be mixed together to define a baseline approach.

Methods

A total of 237 records were identified through database searching and other sources. 107 records were screened and 63 of them excluded, resulting in 44 full-text articles assessed for eligibility. Among these papers, we selected 10 records for analysis. The majority of them pursue the classification between fractured and not fractured bones, while just two of them tried to classify the different types of fractures. We have chosen papers which, in our personal opinion, contain strengths given by a Deep Learning approach that should be used in order to develop a generic tool able to classify every type of fracture in each bone of the human body.

Review

Before the advent of CNN, the pre-processing phase was a fundamental part of the work. For example, in the work of Dimililer [12], the author's aim was to classify whether a bone in an X-Ray image is fractured or not. The system is composed by a neural network following a pre-processing phase. The tool has been trained with 30 images and tested with 70. The images contain different fractures in size and illumination conditions for each subject. In the pre-processing phase, the images are processed using techniques such as Haar Wavelet and Scale-Invariant Feature Transform (SIFT). Haar Wavelet transform is needed to pre-process images in order to compress them and save memory space, SIFT is a powerful method to detect feature points with high resilience to several issues like rotation, compression, and scaling. In the classification phase the author implements a 3-layers neural network with 1024 input neurons. The whole pipeline is shown in Figure 3.

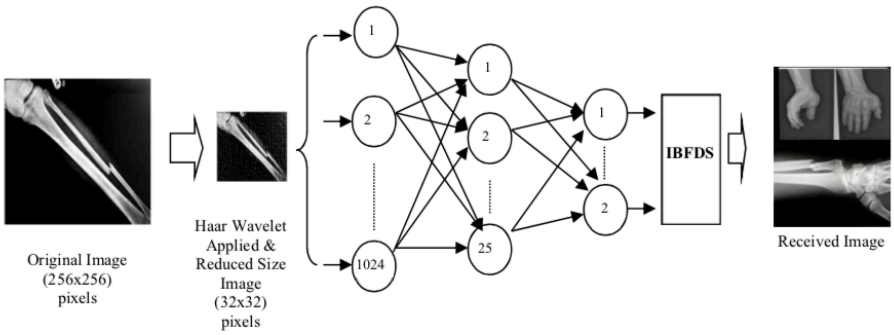


Figure 3 Pipeline of the architecture proposed by Dimililer [12].

This paper describes the technology used before the advent of the convolutional layers. The pre-processing phase was a fundamental part of the process in order to feed the fully connected layers with the correct information. All the remaining papers use convolutional layers to extract features before feeding the fully connected layers.

A dataset of 100 images is quite scant for Deep Learning applications: to solve this problem, Kim and MacKinnon [13] proposed the implementation of transfer learning techniques to classify wrist fractures into two classes: broken and unbroken. According to the authors, this was the first work where transfer learning from pre-trained CNNs has been successfully applied to the problem of fracture detection on plain radiographs. The final dataset was composed of 1.389 images, 695 wrist radiographs showing a fracture, and 694 showing no fracture. Subsequently, data augmentation was applied: horizontal flip, rotation (between 0° and 25°), width and height shift (by a factor of 0-15%), shearing (between 0-10%), and zoom (between 0-15%). This resulted in an overall amplification by a factor of 8 and 5.560 images in the fracture group and 5.552 images in the no-fracture group. This classification was checked and verified by a radiology registrar with 3 years of radiology experience. The network used for this purpose was InceptionV3 [14], originally trained with ImageNet 15) dataset and then adapted and re-trained for the broken/unbroken classification, modifying the top-layer of the network. In this case, the authors demonstrate how transfer learning could be applied to a dataset of images completely non-relatable, i.e., ImageNet.

On the contrary, Lindsey et al. [16] pre-trained their network with a large dataset of bones images. The aim of their work was to implement a tool that could help doctors in diagnosis, to distinguish if a wrist bone is fractured or not and which part of the

bone is fractured. The dataset consisted of 135.845 radiographs of a variety of body parts. Of these, 34.990 radiographs (Training Set) were posterior-anterior or lateral wrist views. The remaining 100.855 radiographs (Pre-Training Set) belonged to 11 other body parts: foot, elbow, shoulder, knee, spine, femur, ankle, humerus, pelvis, hip, and tibia. Every train image was labeled with a bounding box drawn by a group of senior orthopedic surgeons specialized in fractures. The model was a deep CNN, whose architecture extends the common U-Net [17] model. The CNN has two outputs: the probability that the radiograph has a visible fracture and a heat map indicating for each location in the image the probability that the fracture is present in that location. The training of the model can be divided into two stages. In the first stage, the model was pre-trained on the Pre-Training Set. In the second stage, the obtained model was fine-tuned using the Training Set to specialize it in detecting and localizing wrist fractures. After the training and testing phases of the CNN, the authors ran a controlled experiment with 40 emergency medicine clinicians to evaluate each clinician's ability to detect fractures in wrist radiographs both with and without the help of the system. With the use of the proposed system, the specialists' average sensitivities and specificity improved. This study showed that clinicians' evaluation might be improved with this system. This procedure should be applied in each work aiming at showing that a CAD system could help humans in evaluation. Pre-training the model before training it with the wrist bones images seems like a good procedure to adjust the parameters for the task instead of using weights taken from a network that has been trained with a completely different dataset, e.g., ImageNet.g. ImageNet.

Unfortunately, not always you can count on large dataset for pre-train your network: Rajpurkar et al. [18] solved this problem with the introduction of MURA, one of the largest public radiographic images datasets. MURA dataset contains 14.863 musculoskeletal studies of the upper extremity. Each study contains one or more images taken from different views, with a total of 40.561 images, and was manually labelled by radiologists as normal or abnormal. The studies are divided in 9.045 normal and 5.818 abnormal for 7 different extremities including the shoulder, humerus, elbow, forearm, wrist, hand, and finger. The total number of multi-view images is 40.561. The dataset used by the authors is freely available, which implies that may be used by other researchers to pre-train a model for bones classification that can rely only on a small dataset.

Another good practice is to test different networks before choosing the one that gave the best performance, for example, in the work of Olczak et al. [19]. The aim of this study was to assess if standard Deep Learning networks can be trained to

identify if a bone is fractured or not in orthopedic radiographs. The dataset comprised 256.458 hand, wrist, and ankle radiographs, with associated radiologist reports. The authors selected 5 common deep networks for this task: BVLC Reference CaffeNet network (8 layers), VGG CNN S network (8 layers), VGG CNN (16 and 19 layers networks), and Network-in-network (14 layers). The networks were pre-trained on the ImageNet dataset, and the last fully connected layer was replaced to adapt the network for this specific task. As VGG16 had the best performance in the fracture class, the authors selected it for manual review. When comparing the network with the two senior orthopedic surgeons, they found that it performed similarly to humans. Testing different existing networks and choosing the one that performs best is a good practice in the field of neural networks. The dataset contained a really high number of images, the highest among the datasets used in the papers we reviewed. This is obviously one of the most fundamental aspects when working with Deep Learning. With a huge dataset is not easy to label the images manually, and that is why the authors decided to label them automatically from the hospital information. This procedure is subject to errors, and a second review may be useful.

Since now, we have discussed works that obtain good results working with large datasets: Yahalomi et al. [20] proposed a new technique to avoid this. The authors trained a Faster R-CNN [21], a machine vision neural network for object detection, to identify and locate distal radius fractures.

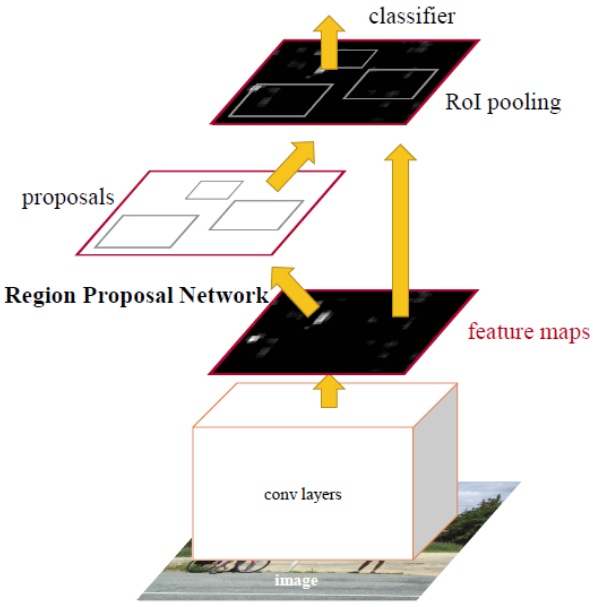


Figure 4 Faster-RCNN architecture [21].

The initial dataset was composed of just 55 images of distal radius fractures and 40 images of hands without fractures. In addition, 25 images not showing hand bones are used for the negative test set. Each image was labeled with bounding boxes around the fractured area. The authors used a Faster R-CNN to achieve two different tasks: classifying whether the fracture is present or not and finding the fracture's location. Faster R-CNN is an evolution of R-CNN and Fast R-CNN, where region proposals are generated by CNNs rather than selective search. Faster R-CNN has 3 different phases. At first, the input images go through a CNN that extracts feature maps. Secondly, an RPN (Region Proposal Network) is used for generating region proposals, i.e., to pre-check which location contains an object without classifying the entity of the object. The output is then passed through a ROI (Region Of Interest) pooling to perform max pooling on inputs of non-uniform sizes and obtain fixed-size feature maps. Finally, the pooled area goes through CNN and two fully connected branches for class softmax and bounding box regressor to detect the object class and return the bounding box of that object. The neural network used in this work was VGG16. This is the only work we found that implements the technology of R-CNN to not only classify fracture but also detect the exact region of the fracture with high accuracy (the results are demonstrated to be significantly more accurate than the detection achieved by physicians and radiologists).

Another improvement that can be applied is to select some specific region of the images to improve the network performance, as in Thurston *et al.* [22] The aim of this paper was to improve the performance of the system described and already discussed by Kim & MacKinnon n (1. The improvement of this work is given by removing unnecessary parts of the image with a semi-automated cropping process. The region of interest was defined using the Python OpenCV [23] *matchTemplate()* function. The method takes a template image and slides it across every position in the subject image (the wrist radiograph), returning the position where the closest match was calculated. In this study, the region of interest was the distal radius. The template was, therefore, an anatomical representation of the distal radius. The template was produced using a representative lateral wrist radiograph and a smoothing algorithm followed by a binary threshold to segment the bone. The process is showed in Figure 5.

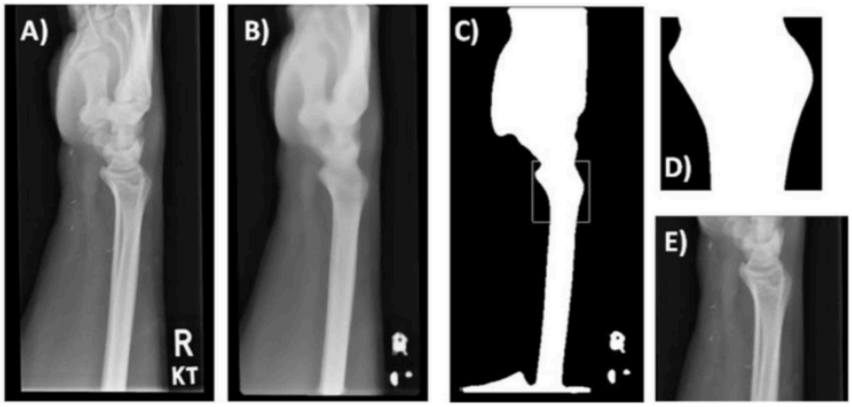


Figure 5 Pre-processing phases in [22]

A scaled template matching approach was adopted to account for different wrist sizes. The model's accuracy was improved when the region of interest focus was applied. This extension study has demonstrated that the accuracy of the network to predict fractures can be increased by removing surplus imaging data. The process is still semi-automated, it should be possible to make it fully automated using different techniques, for example, using feature matching that tries to match features between the template and the image.

Neural networks are often defined as “black box”, as it’s really hard to understand the learning process. This can be avoided with some visualization techniques, such as the one used by Cheng *et al*. [24]. The aim of this work is to use a CNN to classify and localize hip fractures on plain frontal pelvic radiographs. The localization phase is implemented by the use of gradient-weighted class activation mapping or Grad-CAM [25] to confirm the validity of the model. The authors used DenseNet [26] network for the classification task. To demonstrate that the CNN is actually focusing on the right area of the images, the authors implemented Grad-CAM to generate a heat map (Figure 6) in the images that the network classified as fractured. The heat maps computed with Grad-CAM were reviewed: after analyzing 49 heat map images, only two images identified the wrong activation site. Using Grad-CAM can confirm that the network is actually focusing on the correct area of images.

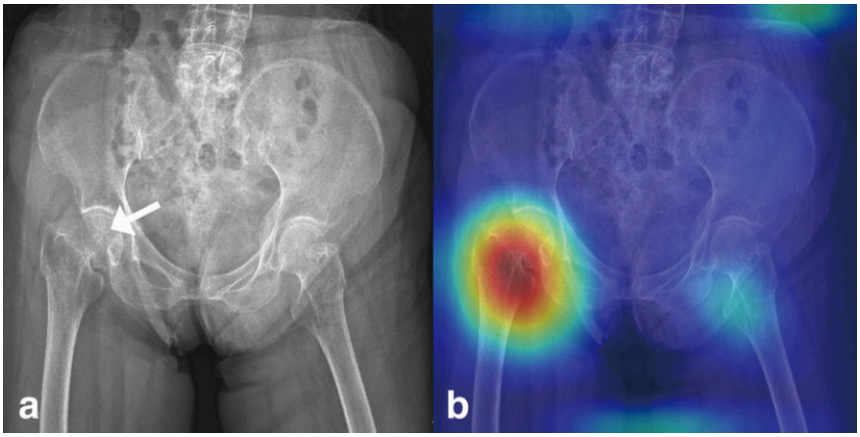


Figure 6 Heatmap generated by Grad-CAM [24]

Since now, we have dealt with papers tackling binary classification between broken and unbroken bones. The first work concerning the classification of different fractures was proposed by Chung et al. [27]. This work addressed the classification problem in different types of fractures in the proximal humerus bone. To evaluate the performance of fracture classification, the authors refer to Neer's classification, which is the most commonly used classification for the proximal humerus fracture, and distinguish 4 different types of fracture: greater tuberosity (B), surgical neck (C), 3-part (D), and 4-part (E) (Figure 7).

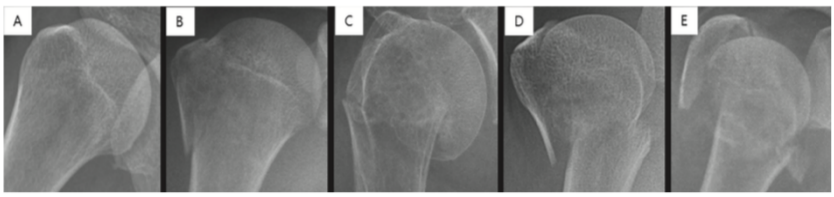


Figure 7 Neer's classification for proximal humerus [27]

The healthy humerus group is named A. Fracture classification was performed by two shoulder orthopedic specialists with 14 and 17 years of experience and one radiologist with expertise in musculoskeletal diseases and 15 years of experience. 515 cases were labeled as A, 346 cases as B, 514 cases as C, 269 cases as D, and 247 cases as E. The dataset of the 1.891 images was divided into 10 partitions without overlapping images: 1 partition was used as a test dataset, while all other images were used as training datasets. The authors used the open-source pre-trained ResNet-152 [28] as a deep CNN model. As the dataset was divided into 10 partitions, 10 experiments were performed in order to obtain an average performance. The ResNet-152 showed superior performance to that of general

physicians and general orthopedists and similar performance to that of shoulder orthopedists. The authors also used 10-fold cross-validation, which is a good practice for reducing ealuation biases. The labeling phase should include different specialists, as this fracture level is really complex to classify..

A limitation of this work is the use of a specific classification for the proximal humerus fracture: to define a generalized approach, we should use a classification structure that can be applied to different bones. In the work of Jiménez-Sánchez et al. [29], the authors proposed a fully automatic CAD tool able to identify, localize and finally classify proximal femur fractures on X-Rays images according to the AO classification. Following AO classification, proximal femur fractures are divided into three main groups: A, B, and C, depending on the area that is involved. Each of these classes is subsequently divided into sub-groups. The dataset is composed of a total of 1.347 X-Ray images. For the two classes problem, 780 fracture images and 567 normal images were considered. The same dataset was used for the three-class problem considering 327 images of type A fractures, 453 of type B fractures, and 567 normal X-Rays. Three clinical experts participated in the evaluation: one 5th-year resident trauma surgeon, one trauma surgery attendant, and one senior radiologist. Using AO foundation classification is a perfect approach to define a generalized method. The main obstacle broached by the authors is the large imbalance in the frequency of appearance of the classes in the fine-grained classification. A method to deal with unbalanced data must be defined.

Discussion

Most of the work focused on the classification between broken and unbroken bones without extending the task to different types of fractures. We think that a generalized tool able to distinguish different types of fractures should follow the classification stated by the AO foundation. The AO classification is hierarchical and is determined by the localization and configurations of the fracture lines, where each bone is divided into subsequent sub-groups of fracture. In the common literature, the AO classification was claimed to present a beer reproducibility compared to other classification systems [30], and its configuration made it optimal for a classification task. Plus, the structure is the same for different bones in the human body, so the approach could be easily extended. Once defined the correct classification system, an adequate dataset is certainly one of the most important aspects for a Deep Learning based application to operate efficiently. Even if in some work, good results have been obtained for the fracture/no fracture classification

without using a large dataset 12,20) a correct number of images is suggested when the network has to distinguish different sub-groups of fractures.

The dataset could be increased and balanced with data augmentation techniques if needed, but without adding useless or misleading information. For example, using shear, strain or spot noise augmentation could cause a normal bone image to be classified as a bone with a fracture [20]. Thus, data augmentation is not always enough to balance irregular datasets. One idea to tackle this could be to assign different weights to different classes when computing the cost function - classes with few images will be associated with higher weights. Also, technologies such as Generative Adversarial Networks) might be used to generate "fake" fractured bones, but this could probably be unfeasible as the fractures might result in unrealistic. Concerning the pre-processing phase, the dataset should be cleaned from images containing prostheses or other evident defects, and we recommend using no more than one image per person to decrease the overperformance by the inclusion of a very similar image of the same patient. It is also demonstrated that selecting the fractured area and feeding the neural network with cropped images instead of the full image improves the network performances [22,29]. For this reason, a fully automated tool to select the fractured regions should be designed for the pre-processing phase. In addition, it is suggested to use lossless formats such as PNG and TIFF and to resize images to different sizes and see which oneks best [20]. Unfortunately, this is not feasible when using an existing network with transfer learning because the input images must have a fixed size. Transfer learning allows using a network, pre-trained on a different dataset, for your dataset. The most used network architectures were VGG, ResNet, DenseNet, and Inception, pre-trained with ImageNet dataset. As demonstrated in different papers, pre-training the network using a larger dataset of X-Ray bone images may improve the performance [16,18]. For example, MURA dataset [18] is one of the biggest bone datasets freely available. For this reason, we recommend trying different networks pre-trained with MURA dataset and testing which one works best for the specific problem at hand. Cross-validation should be used to demonstrate that the network correctly generalizes the dataset features [27]. Another improvement could be introduced by removing surplus imaging data. For example, if a network has to classify between the no fracture class and three different types of fractures A, B, and C, it works at its best if trained excluding no fracture images. Following these results and the AO foundation classification, one idea should be to apply a hierarchical approach. To be clearer, a first network that classifies between fracture and no fracture and a subsequent one that takes the images predicted as fracture and classifies them in A, B, and C. Finally, Class Activation Mapping or similar technologies should be used

to see where the network is focusing [18,24,29]. Last but not least, also specialists have a fundamental role: the dataset must be correctly labeled, and different years of experience are needed to properly classify the types of fractures following the AO classification, especially the subgroups. Both for labeling and evaluation, if possible, more than one expert coming from different specializations should be enrolled in order to have multiple opinions. As the final aim of this tool would be to prove that the CAD system effectively helps doctors in diagnosis, it would be also important to evaluate the performance of the specialists with and without the help [16].

All these discoveries will be used to develop our own framework, presented in Section 2.1.

1.1.2 Computer assisted surgery in the IOR (Intelligent Operating Room)

Introduction

In the last few years, DL is becoming more and more widely used in the world of computer vision technologies, giving astonishing results in different fields of application. Recently DL has gained a fundamental role in the medical environment, and many different tremendous applications have been implemented, from orthopedics [5] to oncology [32] and many more. One of the sub-field where DL could give a fundamental contribution is certainly surgery. Due to its increasing digitalization, the operating room (OR) promises to directly benefit from this progress in the form of new assistance tools that can enhance the abilities and performance of surgical teams. The concept of "Intelligent Operating Room" [33], a collaborative OR based on highly intuitive, natural, and multimodal interaction, will be the final product of this progress. The potential of DL is formidable: by learning from the data from the past hundreds or thousands of procedures, it will be possible to monitor the progress of a surgery operation, detect potential anomalies, provide context-aware decision support, compare best practices, make objective and data-driven analyses of the workflow and provide new ways to interact with the multi-modal data during and after a procedure, to name a few examples. As stated in Hashimoto *et al.* [34], AI applications in surgery could be divided into three groups: preoperative (Pre-Op), intraoperative (Intra-Op) and postoperative (Post-Op). The Pre-Op phase (Figure 8a) is used to perform tests or prior diagnosis, one

example of AI application could be the already discussed bone fracture classification to adapt the surgery to the specific type of fracture. The Post-Op phase (Figure 8b) takes place after the operation: an example could be found in [35], where the authors implement a neural network (NN) to detect post-surgery patient mobilization activities. The term Intra-Op refers to the time during surgery. Intra-Op care is patient care during an operation, and activities such as monitoring the patient's vital signs, fluid therapy, medication transfusion, blood oxygenation levels, anesthesia, and radiography are examples of Intra-Op care. Intraoperative care is provided by surgeons, nurses, anesthesiologists, nurse anesthetists, surgical technicians, and residents, all working as a team. During the Intra-Op phase, different impediments could arise (38), and each team member should be able to act promptly [36].

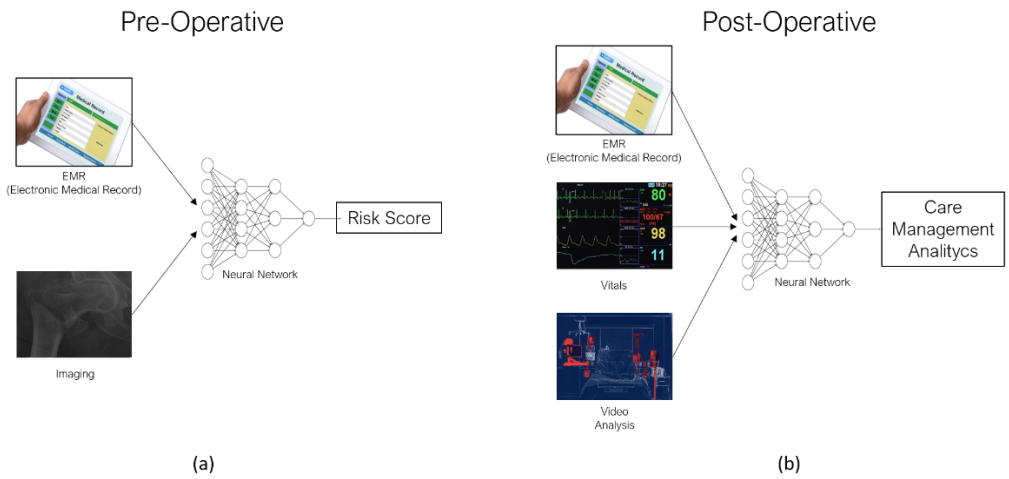


Figure 8 Examples of Pre-Operative and Post-Operative phases. In the Pre-Op phase (a), Electronic Medical Record (EMR) and imaging can predict the risk score. In the Post-Op phase (b), EMR, vital information and video analysis can improve care management.

Methods

We collected on Google Scholar, Springer and Science Direct with keywords "Deep Learning", "Operating Room", "Real Time", "Surgical Workflow", "Intra Operative" a total of n=996 papers published after January 2015. After review of titles and abstracts, n=83 articles were selected for a full review to determine inclusion status. Subsequent full-text review led to the exclusion off n=31 studies for failing to satisfy the eligibility criteria: we excluded n=7 studies using surpassed technologies, n=8 studies which obtained low results, *and* n=16 studies tackling

off-topic arguments. This process has led to the inclusion of n=52 studies for effective analysis. The PRISMA [37] flowchart is shown in Figure 9.

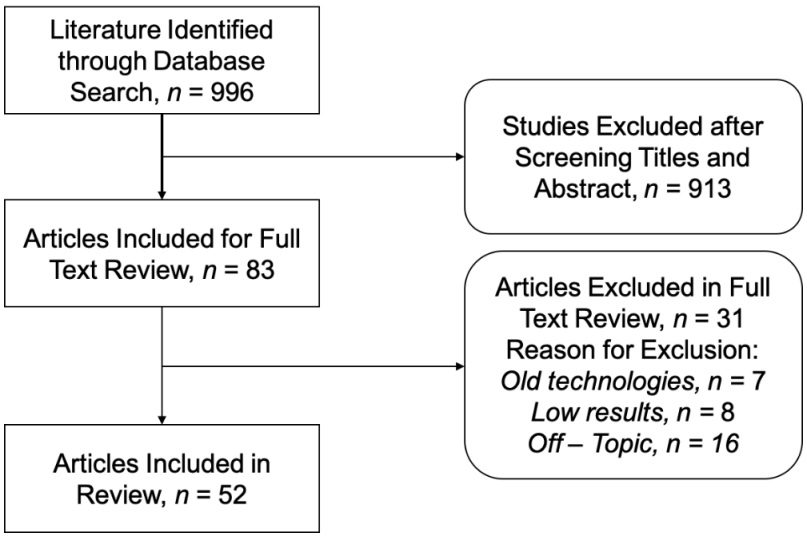


Figure 9 PRISMA Flowchart

Review

The review is organized as follows: firstly, we selected and discussed the principal objectives of this application, then, we focused on how the data is collected, i.e. we outlined how we can elaborate input from different types of medium and, finally, we studied how DL was applied to extract information. This whole pipeline is shown in Figure 10.

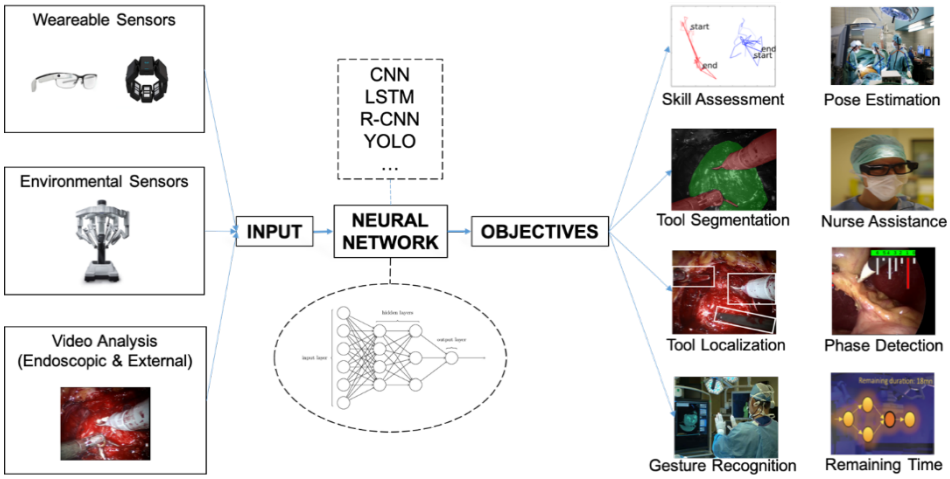


Figure 10 Review Pipeline (only the principal objectives are shown).

A. Objectives

There are several applications of DL in Intra-Op surgery. We grouped them into three categories which we considered fundamental: surgical skill assessment, monitoring of surgery progress, and room/team management. We chose these three groups due to the large number of related studies that made it meaningful to perform a comparison between them.

Surgical skill assessment

Currently, surgical training programs aim to objectively evaluate surgical trainees' basic surgical skills using methods such as the Objective Structured Assessment of Technical Skills (OSATS) [38] and the Ottawa Surgical Competency Operating Room Evaluation (O-SCORE) [39]. Despite this, these tools rely on the observations and judgments of an individual, inevitably associated with subjectivity and bias. Plus, manual feedback from senior surgeons observing less experienced trainees is a laborious task that is very expensive and time-consuming. As such, automated and analytically objective ways to evaluate surgical technical skills are needed. The main aim of this task is to extract features from a surgical process that could be utilized to define the level of expertise. In [40] and [41] the authors used a Convolutional Neural Network (CNN) to predict three different levels of expertise (*novice, intermediate, and expert).* In both cases the input analyzed was taken by external sensors to distinguish the *suturing*, *needle-passing*, and *knot-tying* phases. The dataset used for training was the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [42]. In the first work, the network was also adapted to predict the OSATS level and the CAM [43] technology was used to see which part

of the trajectory the network was focusing on predicting the correct level of expertise. A different metric was used in [44]: the Global Operative Assessment of Laparoscopic Skills (GOALS) rating system. Five values - *depth perception*, *bimanual dexterity, efficiency*, *tissue handling*, and *autonomy* – are considered to evaluate surgeons on laparoscopic surgery performance on a scale of 1 to 5, a more precise evaluation compared to just three levels of expertise.

Monitoring of surgery progresses

Recognizing the surgical workflow is one of the fundamental tasks required to design truly intelligent assistant systems for the OR. Surgical procedures are characterized by pre-defined phases and identifying the current phase is crucial to monitor the surgical workflow. Many studies have been carried out in the past few years, consisting of a NN trained with videos labeled with the corresponding phase for each frame [45–51]. The task of surgical phase recognition consists of identifying automatically at each time *t* the current phase of the surgery directly from the videos. Another technique is to deduct the phase from the tool's presence or localization. The tools are very informative of the actions taking place during the surgery. Plus, they can be used to recognize more subtle activities than the phases described above, such as which tool is performing what action on which part of the anatomy. Detecting surgical tools is particularly challenging due to their similarity, their occlusions by blood or tissues, and the large changes of their sizes in the images due to perspective foreshortening. DL approaches have again been shown to be very successful in recognizing and localizing the surgical tools present in images. The most basic method to tackle this is tool presence detection shown in Figure 11 (a) i.e., try to predict what tools are present without any localization of them [52]. In this context, Twinanda *et al.* [53] proposed EndoNet, a network able to detect tool and concatenate this result to predict the phase, which became a really used network in this field. The next step is tool detection and localization, shown in Figure 11 (b), that has been applied in [44,54–57] and more in particular for needle detection in [58]. Understanding not only the presence but also the position gives great improvement in phase recognition. This can evolve in tracking [59,60], and segmentation [61–65], a technique to define the contours of all the different tools in the current frame, shown in Figure 11 (c).

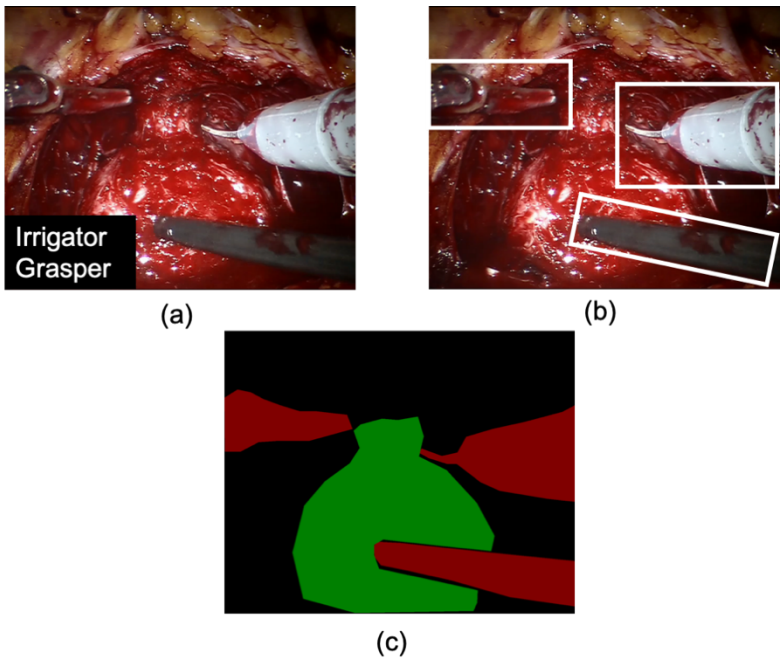


Figure 11 Tool Presence (a), Tool Localization (b) and Tool Segmentation (c).

All these discussed studies:

- Used supervised learning; conversely, in [66] and [67] the authors applied unsupervised learning to assign phases to specific frames and avoid the time consuming labeling phase.
- Focused on endoscopic videos; the same techniques have been applied to external cataracts video [68,69]. A freely dataset containing videos of cataracts operation (CATARACTS, [70]) has also been published.
- Had to handle unbalanced dataset; the common problem of imbalance in surgical tool detection was discussed by Sahu *et al.* in [71].

Room/Team Management and Surgeon Assistance

The progress of technology is moving toward the direction of creating an Intelligent OR, and different aspects must be considered to realize this concept. Firstly, during a surgery operation, many people are involved and being able to localize the medical staff could give a greater organization to the team and can also provide important clues for surgical workflow analysis. This could be achieved with pose estimation (Figure 12) or face detection [72]. Clinical settings include severe occlusions, clutter and difficult lighting conditions that make pose estimation harder than the standard. In [73] and [74] the authors fused 3D and 2D camera views data to obtain trustable results in pose estimation. Secondly, technology could be used

to support the medical staff: for example, Nishio *et al.* [75] proposed a computer-aided navigation system that indicates the current procedure for nurses, and also instructs nurses to prepare surgical instruments to be used in the next procedure using smart glasses. Zhou and Wachs [76] studied a system to substitute human nurses with robotic nurses, that were trained for turn-taking events (when the turn switch from surgeon to nurse and vice versa) prediction. Results demonstrated that the robotic nurse surpassed human performance for early turn-taking prediction: 10% of completion of the actual action was enough for the robot to understand that a turn-taking event was happening. Thirdly, surgeon assistance is a focal aspect of IOR. For every hospital, it is desirable to fully utilize its operating room capacity. Wrong scheduling often derives from the incorrect estimation of the surgery duration, which may vary significantly due to the diversity of patient conditions, surgeon skills, and intraoperative situations. To solve this, Aksamentov *et al.* [77] proposed real-time DL-based tool to estimate the remaining surgery time. Another problem that has to be solved is the smoke presence during surgery, which can considerably impede a patient's treatment while additionally entailing serious deleterious health effects. Hence, state-of-the-art surgical procedures employ smoke evacuation systems utilizing industrial sensors. As an alternate approach, CNN and video analysis can be used to detect the smoke presence [78]. Finally, in sterile environments such as ORs where surgeons need to interact with images on screens, contamination issues may happen if the surgeon touches a keyboard or a mouse. To reduce contamination, different touchless gesture recognition algorithms have been proposed in [79–81]. In these works, the authors implemented an algorithm for the surgeons to control the screens just with gestures. The same idea was applied in [82] using gaze estimation.



Figure 12 An example of pose estimation.

B. Data collection

As modern ORs have seen a surge in the use of digital devices and sensors in the last decades, such data are nowadays increasingly available, for instance, in the form of images, device signals, electronic reports, and videos. Considering the current challenge to collect data from all equipment in the OR, most existing works analyzing the temporal workflow have usually relied on videos or data manually derived from videos. This is because cameras not only capture a large amount of information but are also naturally present in many minimally invasive procedures or can be installed without disrupting the surgical workflow. It is however noteworthy that there also exist efforts to recognize activities from the signals of other devices, such as from sensors that can be integrated into the OR to capture a person's position or tool/device usage. Working with data in the medical field presents different obstacles. Firstly, captured video innately carries sensitive information that should not be completely visible in order to preserve the patient's and the clinical teams' identities. To solve this, in [83] the authors proposed a solution to anonymize faces in the operating room. Another issue is that everything in the operating room should be sterilized, for this reason, wearable sensors are not a suitable solution in the majority of the case. Usually, using video (endoscopic or external) as input is the easiest and most common solution. Otherwise, external sensors avoid the problem of sterilization as well.

Wearable sensors

As already said, sterilization in the surgery room is a huge obstacle for wearable sensors, for this reason, just two studies were found addressing this solution. In the first [76], the authors implemented a network that receives in input 50 different types of signals. To collect the multimodal input, the authors utilized:

- MYO armband sensors, a gesture-capturing device worn on the forearm, to capture the arm orientation, arm acceleration, arm gyroscope, arm muscle Electromyograph (EMG) signals.
- Epoc headset sensor, a brain–computer interface used to capture a surgeon's head motions and Electroencephalography (EEG) signals. It collects head EEG, head gyro and emotion classification (*engagement, frustration, meditation, excitement* and *valence*).
- Kinect sensor, a motion sensing and acoustic recording device, to record face orientation, body postures, left hand extension, right hand extension and acoustic amplitude confidence.

In the second study [55] the author proposed the use of smart glasses to collect data in real time and see essential information about tools during surgery that may reduce the complication of the procedures.

Endoscopic video

The majority of the works used endoscopic videos to solve specific tasks, as they are really easy to implement and don't require ulterior technology [44,46]. Abdominal surgery (prostatectomy, sigmoidectomy, etc) was the most studied application and public dataset were made available for experimentation, such as *Cholec80* [53], consisting on a collection of cholecystectomy video. Apart from abdominal surgery, in [84] the authors proposed OffsetNet, a network for localization in the lung, and in [85] Bouget *et al.* worked on images from neurosurgical microscopes.

External videos

While endoscopic videos are a very useful source of data in minimally invasive surgical procedures, they do not capture the activities present in the whole room, which are complementary to the ones performed inside the patient. The solution to this is external videos. Ceiling cameras are mainly used for room and team management as the already discussed pose estimation [74] and face detection [72]. In [82] the authors placed a web camera on top of the laparoscopic video pointing to the surgeon for gaze estimation. External cameras could also be used for force feedback, as in [86], thus avoiding the use of add-on sensors, and phase recognition with RGB-D cameras [87].

External sensors

External sensors are another solution to the problem of surgeons' sterilization of the surgeon. Recent progress in surgical robotics, such as the da Vinci surgical system, enabled the recording of kinematic data from various surgical tasks. The da Vinci robot is comprised of two master tool manipulators (MTMs) on the left and right sides and two patient-sides slave manipulators (PSMs) (Figure 13). In [40,41] the authors utilized as input 19 measures for each end-effector (Cartesian positions, rotations, linear velocities, angular velocities, and the gripper angle), for a total of 76 input values collected and elaborated to evaluate surgeon skills.

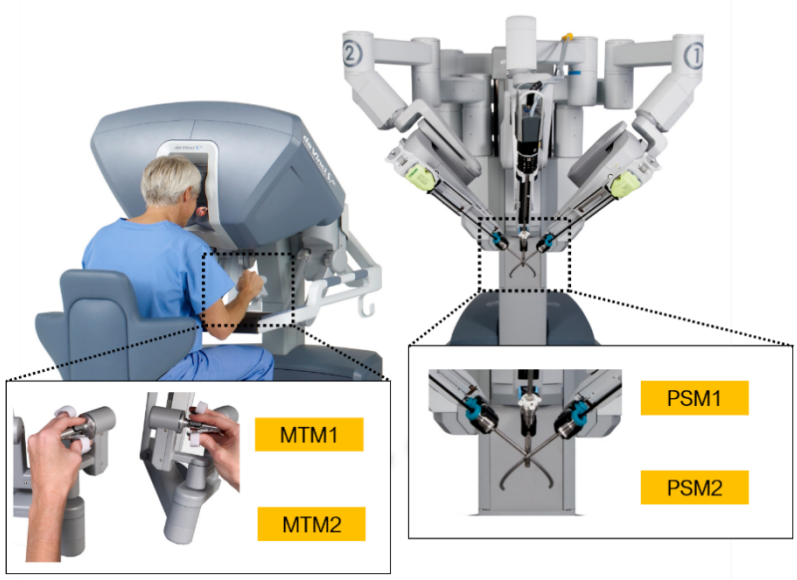


Figure 13 The Da Vinci robot manipulators.

C. Deep Learning methods

Before the advent of machine learning, different feature extraction techniques were used [88,89], such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Principal Component Analysis (PCA), and Codebook. Lately, with the advent of DL, some new interesting solutions have been proposed by the community.

Convolutional Neural Networks (CNN)

CNN are the most used technology in computer vision as they're able to successfully capture the spatial and temporal dependencies in an image through the application of relevant filters, saving memory space and limiting the number of parameters to compute. For this reason, CNN are perfectly suitable for videos. In the case of laparoscopic video for workflow analysis, one of the most popular choices is EndoNet [53], a novel CNN architecture designed to carry out the phase recognition and tool presence detection tasks in a multi-task manner that has been subsequently utilized and improved in different studies. Other networks used go from *ad hoc* architectures [41] to modifications of popular networks such as ResNet in [64], VGG and InceptionV3 in [51], U-Net, TernausNet, LinkNet in [65], AlexNet in [66], and Inception-ResNet in [47]. CNN are often also used for gesture

recognition with external videos, such as in [80] and [74]. The biggest limitation of CNNs is that the temporal dependencies are not considered.

CNN + LSTM (Long Short Term Memory)

LSTMs are recurrent neural networks (RNN) that can learn and remember the information previously extracted and are known as suited for time series data analysis. Or this reason, RNN are the solution to CNN for time dependencies. Compared to a conventional RNN, which can only learn short-term representation, LSTM can also learn the long-term representation of data. RNN could be used alone for computation as in [45], but using LSTM layers stack on CNN layers has some advantages and has been demonstrated to confer high accuracy in many domains. Deep-Onto [48], SV-RCNet [46], and MTRCNet-CL [90] are examples of an ensemble of CNN and LSTM models for workflow recognition. Other examples of solutions based on LSTM can be found in [59] and [49].

Object Detection Algorithm

The main difference between object detection and classification algorithms is that a detection algorithm tries to draw a bounding box around the object of interest to locate it inside the image. This cannot be achieved with a standard CNN followed by a fully connected layer because the length of the output layer is variable, not constant, as the number of occurrences of the objects of interest is not fixed. A simple approach to solve this problem would be to take various regions of interest from the image and use CNN to classify the presence of the object within the specific region. Hence, it would be needed to select a large number of regions and this could be computationally limiting. The evolution of R-CNNs (Region CNNs) and Fast R-CNNs, Faster R-CNNs [21], are a solution to this problem. These networks usually consist of different steps. The first is a region proposal algorithm to generate bounding boxes of possible objects in the image; the second is a feature generation stage to obtain features of these objects, usually using a CNN; the third is a classification layer to predict which class the object belongs to; the fourth and last is a regression layer to make the coordinates of the object bounding box more precise. Faster R-CNN were used in [44], [55], [57] and [58] for tools detection. Faster R-CNN used regions to localize the object within the image and the network does not look at the complete image. Instead, just parts of the image which have high probabilities of containing the object. YOLO (You Only Look Once) [91] is an object detection algorithm much different from the region-based algorithms seen above. In YOLO a single CNN predicts the bounding boxes and the class probabilities for these boxes. YOLO and YOLO9000 were used for surgical tool detection in [54] and [56], respectively.

Other techniques

- Generative Adversarial Networks (GANs) [31] in [67] for feature extraction in surgical videos. GANs is a technology that, given, for example, a set of faces, is able to produce fake faces. In GANs, two networks compete using unsupervised machine learning. The first network, the generator, generates a data instance (for example, an image) that mimics real-world data. This is fed to the second network, the discriminator, along with authentic examples from the real world. The generator aims to convince the discriminator that the generated data is authentic, while the discriminator has to discern between which data is real and which is replicated.
- Spiking Neural Network (SNN) [92] in [76] for turn-taking early prediction. Conventional neural network models enforce synchronous firing of neurons of the same layer. The connections between consecutive layers are forced to have the same conduction delay, thus, all the neurons of the same layer can fire at the same time. This rigid structure poses difficulties when modeling multimodal temporal sequences, since the delays between layers are fixed and cannot adapt to different temporal resolutions associated with multimodal signals. SNN, however, can model the variability of axonal conduction delays between neurons. Because of the not uniform conduction delays, the times for the firings to traverse the network will be different. This way, the asynchronous effect of multimodal signals can be modeled.
- Reinforcement Learning (RL) in [93] for gesture segmentation and classification. RL is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize some notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.

Discussion

Table 2 Summary of objectives and related input, neural network and output.

Objective	Input	Neural Network	Output
Skill Evaluation	Kinematic Data	CNN	Level of Expertise
Phase Detection	Endoscopic Video	CNN + LSTM	Current Phase
Tool Localization	Endoscopic Video	Faster R-CNN / YOLO	Tools Bounding Boxes
Tool Segmentation	Endoscopic Video	CNN / AutoEncoder	Class for each pixel
Time Remaining	Endoscopic Video	CNN + LSTM	Predicted remaining time
Gesture Recognition	External Video	CNN	Type of gesture
Pose Estimation	External Video	CNN	Human pose structure estimation

From this overview, it is clear that different aspects, summarized in Table 2, should be considered to realize an IOR. Many useful data could be derived from different types of input: as we already stated, most wearable sensors are not a practical solution because they have to be constantly sterilized and they can also be uncomfortable for surgeons. We asked a senior surgeon, and he confirmed that smart glasses could be the only wearable sensor he could use without decreasing his performance. With smart glasses, for example, skill evaluation could include not only kinematics data, obtained with external sensors, but also the movement of the surgeon's pupil. In this context, video analysis is fundamental. Firstly, an IOR should have external cameras to implement pose estimation in order to localize the different actors in the IOR. Plus, cameras should be used to develop gesture recognition software that avoids contact with an unsterilized mouse or other types of hardware. Secondly, endoscopic videos are an invaluable source of information to monitor the surgical workflow: tool detection, localization, and segmentation could be really helpful to understand the correct phase, which is a prerequisite for different aims. Knowing the phase, nurses (both robotic and human) can be easily assisted and be more reactive to stimuli. Phase recognition made also possible to predict the remaining time. The majority of the most common obstacles during surgery, according to the specialists we consulted, could be solved with video analysis, for example, blood accumulation. Algorithmically speaking, to elaborate

real-time video the correct solution could be an ensemble of CNN and LSTM to preserve both spatial and temporal dependencies. CNN should be used for skill evaluation, tool segmentation, gesture recognition, and pose estimation. For tool localization, the most correct solution would be Faster-RCNN or YOLO. A graphic conceptualization of IOR is shown in Figure 14. This review outlined how AI and, in particular, DL could help the surgery field in revolutionizing itself, with numerous, and different applications. This process is evolving years by years into the realization of an IOR, equipped with technologies perfectly suited to drastically improve the surgical workflow and lower the number of obstacles that normally occur during surgery.

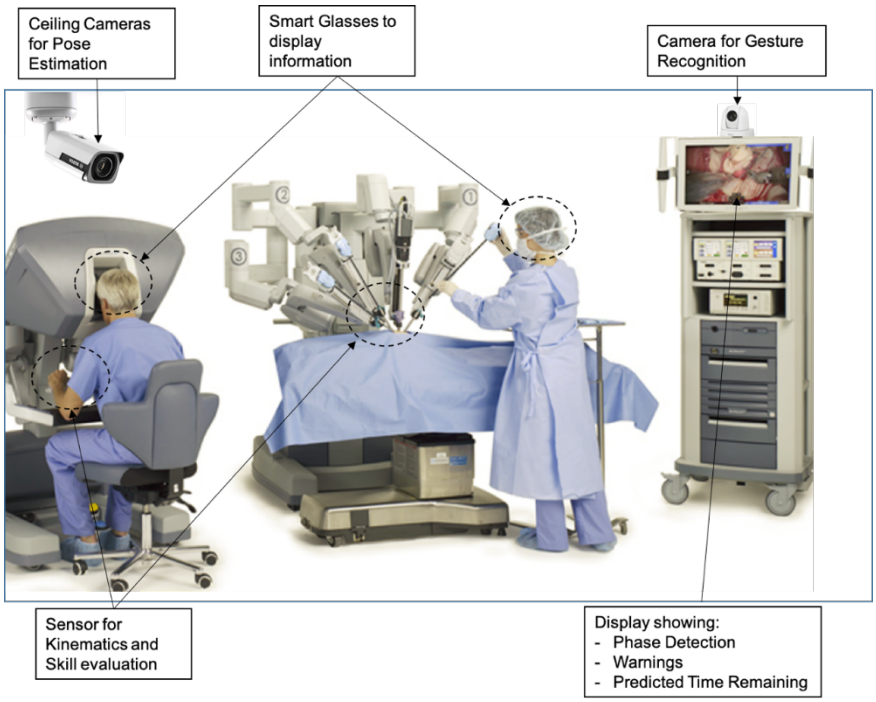


Figure 14 Conceptualization of an IOR.

This review aims to collect all the knowledge relative to this sub-field in a way to select the best omni-comprehensive approach, which we then started to develop and present in Section 3.1.

1.2 Deep Learning in Computer Assisted Troubleshooting

In the same way, we performed an overview of the state-of-the-art in computer assisted troubleshooting, in order to define a baseline to develop the project discussed in Section 3.2.

Overview

As stated in [94], the use of AR in maintenance ranges from dis/assembly to repair, diagnosis, and training. Repair operations are defined as actions aimed at restoring the functional properties of a device [95]. Diagnosis refers to maintenance activities that aim to assess the current state of the product and analyze the causality of deterioration and functional degradation [96]. Training refers to processes that aim to transfer maintenance skills to technicians [97]. Regarding dis/assembly, which is the area where our application resides, as early as 1997, Azuma [98] stated that overlaying a 3D animated drawing could facilitate assembly processes compared to traditional user manuals. In [99], the authors demonstrate a straightforward AR approach that overlays virtual arrows and text on top of the real environment. In [100], the authors used the Hand Held Display (HHD) to perform maintenance tasks on consumer devices by showing the task description at the bottom of the display and providing a few buttons to navigate through the procedure. In [97] the authors showed an effort to provide different levels of instructions. They proposed two levels of guidance: a strong one that supports the user through each step, and a soft one that offers more high-level information and is designed for more experienced users. In [101], the authors incorporated into the AR procedure the ability to provide real-time feedback on the operation. Through the position and orientation of the components, they were able to show warning messages to correct the assembly procedure. Finally, a slightly different approach was proposed in [102], where the authors developed an AR application to simulate the assembly procedure during the initial component design phase. They also estimated the forces involved in assembly by considering the stiffness, shapes, and contact surfaces of both the real component and the virtual prototype.

In all these applications, when AR is applied to a specific situation, the goal is fusing 3D objects with real-time images taken by the camera. The challenge was how to correctly align the virtual objects with their real-world equivalent. Estimating the 6D position of an object from an image is a central problem in Computer Vision (CV). It affects many domains, such as robotics, autonomous driving, medicine, industrial inspection, and virtual/augmented reality applications,

widely used in the entertainment and medical care industries [103–105]. The problem consists of determining the 3D rotation and translation of an object whose shape is known with respect to the camera, using observable details from the reference image. However, solving this problem is not trivial. Due to self-occlusions or symmetries, objects cannot be clearly and unambiguously identifiable, assuming an ambiguous position. In addition, image conditions are not always optimal in terms of illumination and occlusions between depicted objects [106]. In these situations, it is often necessary to add an earlier semantic segmentation or object detection step to identify the area of the image that contains the object before estimating its position. Although researchers have studied this problem for many years, it has experienced something of a renaissance with the advent of DL. Older pose estimation methods were based on geometric approaches, trying to establish correspondences between 3D models and corresponding 2D images of objects using manually annotated local features. With untextured or geometrically complex objects, it is not easy to select local features. In these cases, although matching is time-consuming, it can fail and provide a result that is not always accurate [107]. In opposition to these methods, researchers have introduced other strategies, relying on representations of 2D objects from different viewpoints and comparing them with the original image to determine location and orientation. These methods are very susceptible to variations in illumination and occlusions even though they can handle untextured objects and require many comparisons to achieve a certain level of accuracy, increasing the runtime [105]. With the spread of DL, researchers have introduced new strategies to achieve this goal, improving the traditional methods, and making them more efficient and performant. The basic idea of systems involving CNN is to learn a mapping function between the image, and the 6D position of the object, from images that have three-dimensional position annotations. These methods can achieve very high levels of accuracy but need a lot of data to accurately train the network to work well in real-world cases. One particular approach based on DL relies on a synthetic dataset to train a CNN to predict the object's rotation, such as in [108,109].

After this literature review, the approach proposed in these two papers seemed the most suitable for a general framework, due to its flexibility and extendibility to new objects or procedures.

Nevertheless, the first relies on Faster-CNN for object detection, an algorithm that is 8 times slower than modern object detectors, such as YOLO, and can't be thus used for a real-time application. The second utilized for training both synthetic data and ~22K images real images from PASCAL 3D+, which made the algorithm once again dependent on real data, which are costly to collect.

In Section 3.2, we proposed an approach that overcomes these two obstacles by implementing a real-time detection and an overlay phase dependent on a very small amount of real-images for fine-tuning. The main contribution is in presenting a novel approach that combines two different neural networks to predict the rotation and position of a generic machine in real-world space, leveraging only RGB data from a live video stream. The predicted data can then be used for AR applications like, for example, those that support nonskilled people with maintenance operations.

Chapter 2

First Layer

In the following section, we present the methods we developed after the literature review concerning the first layer. First of all, we discuss the classification of femur fracture. Part of these works has already been published in two different papers [110,111] and one conference paper still under review, which are an evolution of the approach. The work on emotion classification presented in the next section is still under review.

2.1 Femur Fracture Classification: Overview

Introduction

Musculoskeletal diseases represent the most common cause of severe, long-term disability worldwide [112]. Due to the progressive aging of the population, the prevalence and incidence of fragility fractures are increasing and will continue so in the future [113]. In 2010 the estimated incidence of hip fractures was 2.7 million patients per year globally. In the Emergency Department, a pelvis radiograph is performed when a hip fracture is suspected [114]. The correct evaluation and classification of proximal femur fractures by radiologists strongly affects future patients' treatment and prognosis. The AO/OTA (Arbeitsgemeinschaft für Osteosynthesefragen Foundation/Orthopaedic Trauma Association) classification is hierarchical and provides a well-defined methodology for assessing fractures correctly, enabling physicians to guide treatment and communicate with a standardized language [115]. However, the correct classification of hip fractures can be demanding for osteopenia, superimposition of soft tissues in obese patients, and difficult patients' positioning resulting in poorer image quality [116]. The above-mentioned difficulties, the stressful working environment of Emergency Departments, and the perceived complexity of the classification might have affected its widespread utilization, limiting evidence-based fracture management and data collection for research.

In this context, implementing a CAD (Computer Assisted Diagnosis) system in doctors' workflow might have a direct impact on patients' outcomes.

AO/OTA Proximal Femur Classification

Proximal femur fractures were first taken into consideration as they represent the most common reason for admission to an acute orthopaedic ward especially for the elderly population [117]. The proximal femur is labeled as "31", being the first number related to the femur and the second to the proximal region. Then the intertrochanteric region is coded as "A", the neck of femur "B" and the femoral head as "C". Then the fracture is further described in groups and subgroups depending on the complexity and degree of displacement. The classification process adopted in this study is shown in Figure 15.

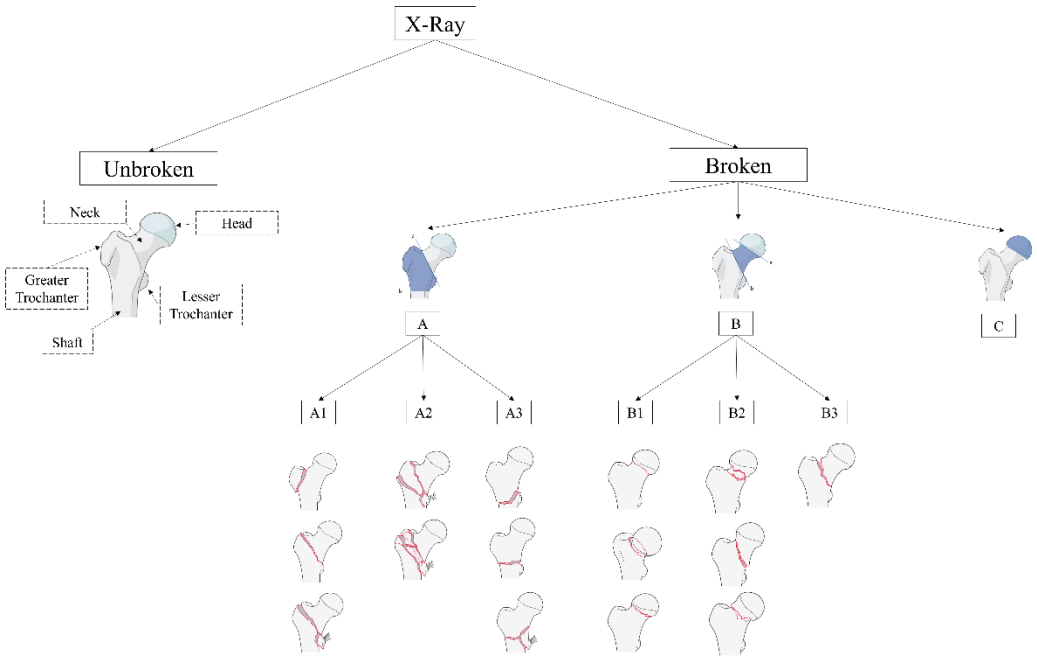


Figure 15 AO/OTA classification.

This classification is hierarchical and determined by the localization and configuration of the fracture lines. Type *A,* type *B,* and type *C* fractures concern the trochanteric region, the femoral neck, and the femoral head, respectively. Each group is then subsequently divided into different levels of subgroups. Even if in the common literature the AO/OTA classification was claimed to present a better reproducibility compared to other classification systems [118], both for intertrochanteric and trochanteric fractures AO/OTA classification present a low reproducibility [119,120]. Some samples are shown in Figure 16.

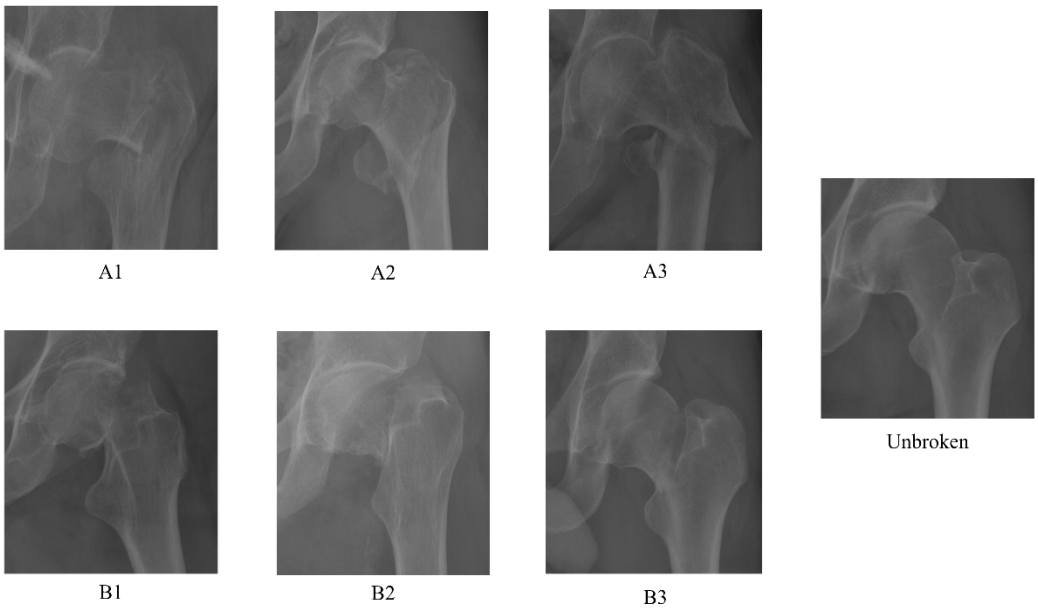


Figure 16 Some samples of real X-Rays images. These samples were used for training the neural network after the cleaning and cropping phase

2.1.1 First approach: Cascade Classifier

The first approach was based on a very simple classifier of CNN in cascade. The work was done in collaboration with KTH Stockholm and the CTO of Turin,

Dataset

The initial dataset was provided and labeled by Dr. Aprato, a senior trauma surgeon with 17 years of experience, and Dr. Audisio, an orthopedic specialist who has worked specifically on femur fractures in the past 5 years, from the CTO (Center for Orthopaedic Trauma) of Turin and was composed by 1.787 images of the entire or half hip bone. After a cropping phase, where the areas related to the right and left femur were selected through a semi-automated cropping method and resized to 299 x 299, and a cleaning phase, where images containing prosthesis (n=15), with a too low resolution (n=47), with the area around the femur partially hidden (n=23) or showing a lateral view (n=98) were excluded (n=183), the dataset was reviewed a second time by our specialists. The final number of images was: 1.133 *Unbroken* femurs, 570 type *A*, 750 type *B,* and 4 type *C*. Because of the low number of *C* fractures (n=4), we decided to start excluding this class. Among the *A* type, 280 fractures were labeled as *A1*, 183 as *A2,* and 107 as *A3*. This process is shown in Figure 17 following the STARD 2015 flow diagram [121]. We choose to start with

A1, *A2,* and *A3* classification as the treatments related to these sub-classes are highly

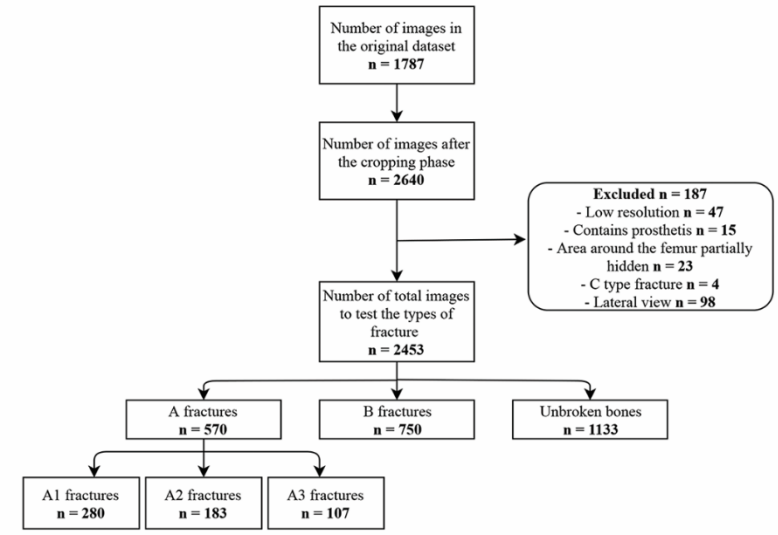


Figure 17 STARD 2015 Flow Diagram to define how the dataset is decomposed.

dependent on the type of fracture. *B* fractures, on the other hand, present the same treatment for different subgroups. The mean age of patients was 83 (63–91) years.

Methods

Two different methods were developed, shown in the flow chart in Figure 18 for the three classes' classification case. After trying different architectures and methods, the model which gave the best test accuracy was a simple InceptionV3 [122] network, with the last layer replaced with a Softmax layer for three and five classes classification and pre-trained on ImageNet [123]. We wrote a function to assign different weights to each class when computing the loss, in order to compensate for the unbalanced dataset, i.e., higher weights have been assigned to classes with fewer images. After trying different configurations, we obtained the best results using a batch size of 32 and Adam optimizer [124] with a learning rate of 0.0001 and beta values of 0.9 and 0.999, respectively. The function to calculate the loss was the *sparse categorical crossentropy*. We also implemented data augmentation in the training set using a rotation range of 10 degrees, horizontal flip, and both height and width shift from 0.0 to 0.1 fraction of total height or width. We did not use any more complex transformations as shearing in order not to create "fake" fractures. We carried on the same computation with VGG16 [125] and ResNet50 [126] before choosing InceptionV3 as the best architecture for the problem at hand, keeping every parameter the same (except the images input size

that is 224 × 224 pixels for both of them). After selecting the best network and using its performance as a baseline, we switched to a second approach training our InceptionV3 network defined above to tackle a binary classification between *Broken* and *Unbroken* and *A* and *B* and a three classes classification between *A1, A2,* and *A3*. We then tested the networks as a cascade of three stages: the first recognizes between *Unbroken* and *Broken*, the second classifies between *A* and *B* the images labeled as *Broken* from the first network and the third classifies between *A1, A2,* and *A3* the ones resulting in *A* class. We then improved one more time the performances for *A1, A2,* and *A3* classes substituting the third stage with two sub-stages. This approach tried to emulate the method used by the specialists to classify fractures and its hierarchical structure is particularly appropriate for AO/OTA Classification. Lastly, we used Grad-CAM [127] to visualize where the network was focusing for the different classifications.

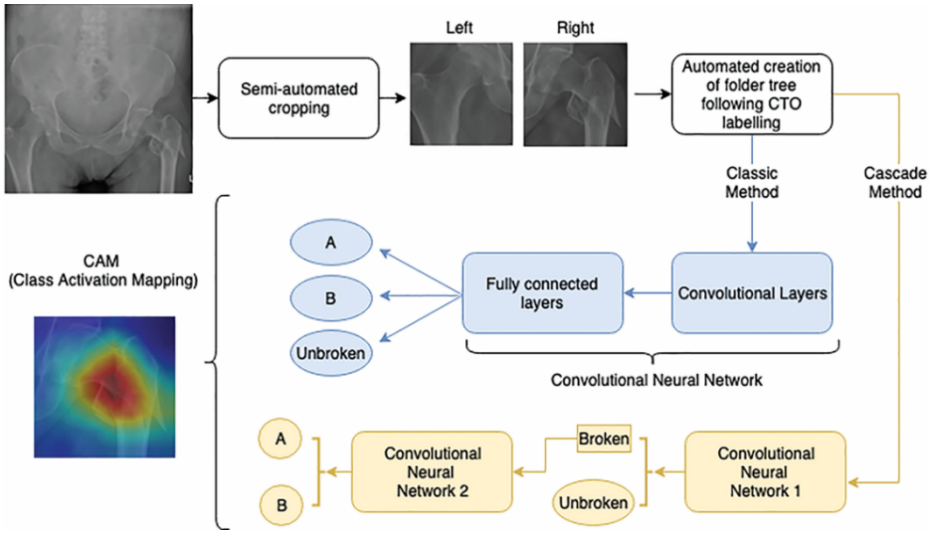


Figure 18 Flow chart for the three classes classification case. Two approaches were presented: a classic CNN for classification used as baseline and a multistage one characterized by subsequent binary networks.

Training, Framework and Evaluation

The networks were trained using 5-fold cross-validation with 20% images for validation, and 20% of images for each class were kept apart for testing. We ran the model for 150 epochs using *EarlyStopping* with a patience of 10 epochs. We used Keras [128], an open-source neural-network library written in Python, running on top of TensorFlow [129], on Ubuntu 16.04.5 LTS with NVIDIA GeForce GTX 1080 Ti. For each network, we computed the top-1 accuracy, the conventional accuracy for the deep CNN answer (top-1) being precisely the expected answer,

among choices of *A*, *B,* and *Unbroken* or *A1*, *A2*, *A3*, *B*, *Unbroken*. Then, the performances for single classes were measured using the area under the receiver operating curve (AUC), precision, recall, and F1-score. Every value was averaged among the 5 folds. Performances of orthopedists with and without the CAD system were measured using accuracy and Cohen's Kappa scores [130].

Results

Baseline Method

Running the aforementioned InceptionV3 model, we obtained, using 5-fold cross-validation, an average test accuracy of 0.87 (CI 0.85-0.88) for three classes classification and 0.78 (CI 0.76-0.79) for five classes classification. We choose InceptionV3 after running ResNet50 and VGG16 with the same parameters and obtaining lower test accuracies. ResNet50 gave an average test accuracy of 0.85 (CI 0.84-0.87) for three classes' classification and 0.75 (CI 0.73-0.77) for five classes' classification. VGG16 gave an average test accuracy of 0.82 (CI 0.79-0.85) for three classes' classification and 0.77 (CI 0.76-0.78) for five classes' classification. These values are shown in Table 3.

Table 3 Comparison between three and five classes classification for InceptionV3, VGG16 and ResNet50.

K-Fold	InceptionV3		VGG16		ResNet50	
	3 classes	5 classes	3 classes	5 classes	3 classes	5 classes
Fold1	0.87	0.76	0.84	0.78	0.88	0.75
Fold2	0.88	0.77	0.82	0.76	0.85	0.74
Fold3	0.85	0.77	0.80	0.76	0.82	0.76
Fold4	0.87	0.78	0.84	0.78	0.84	0.74
Fold5	0.86	0.79	0.79	0.77	0.86	0.77
Average	**0.87**	**0.78**	**0.82**	**0.77**	**0.85**	**0.75**
Standard Deviation	±0.01	±0.01	±0.02	±0.01	±0.02	±0.01

We then computed precision, recall, and F1-score for each class using the InceptionV3 network (Table 4). The values are shown with related confidence intervals. Lastly, we computed the confusion matrix and ROC curves for each fold. ROC and confusion matrix for the model that gave the best results both for three classes and five classes are shown in Figure 19**.**

Table 4 Precision, Recall and F1-score for three and five classes classification using the InceptionV3 network. The number of images for testing is proportioned to the initial dataset unbalance.

Class	Precision		Recall		F1-score		# of images for testing
	3 classes	5 classes	3 classes	5 classes	3 classes	5 classes	
Unbroken	**0.90** (0.86-0.93)	**0.91** (0.89-0.92)	**0.92** (0.89-0.95)	**0.92** (0.91-0.93)	**0.91** (0.89-0.92)	**0.91** (0.90-0.92)	226
B	**0.83** (0.80-0.87)	**0.78** (0.72-0.84)	**0.81** (0.76-0.86)	**0.83** (0.81-0.86)	**0.82** (0.80-0.84)	**0.81** (0.77-0.84)	150
A	**0.86** (0.80-0.91)		**0.84** (0.79-0.89)		**0.85** (0.84-0.86)		114
A1		**0.43** (0.36-0.50)		**0.29** (0.16-0.42)		**0.34** (0.25-0.43)	56
A2		**0.42** (0.38-0.46)		**0.50** (0.27-0.73)		**0.44** (0.35-0.53)	36
A3		**0.74** (0.56-0.93)		**0.57** (0.49-0.66)		**0.64** (0.55-0.72)	21

The average AUC among 5 folds for three classes classification was 0.95 (CI 0.93-0.97) for class *A*, 0.93 (CI 0.92-0.95) for class *B,,* and 0.96 (CI 0.95-0.97) for class *Unbroken*. The average AUC among the the 5 folds for five classes'' classification was 0.84 (CI 0.78-0.90) for class *A1*, 0.93 (CI 0.92-0.93) for class *A2,* 0.97 (CI 0.96-0.98) for class *A3,* 0.93 (CI 0.91-0.95) for class *B* and 0.97 (CI 0.96-0.98) for class *Unbroken*. Clearly, as noticed in Table 4 and Figure 19 (D), the network is not learning to recognize features of *A1, A2,,* and *A3* fractures. This is why we decided to implement the already mentioned multistage approach.

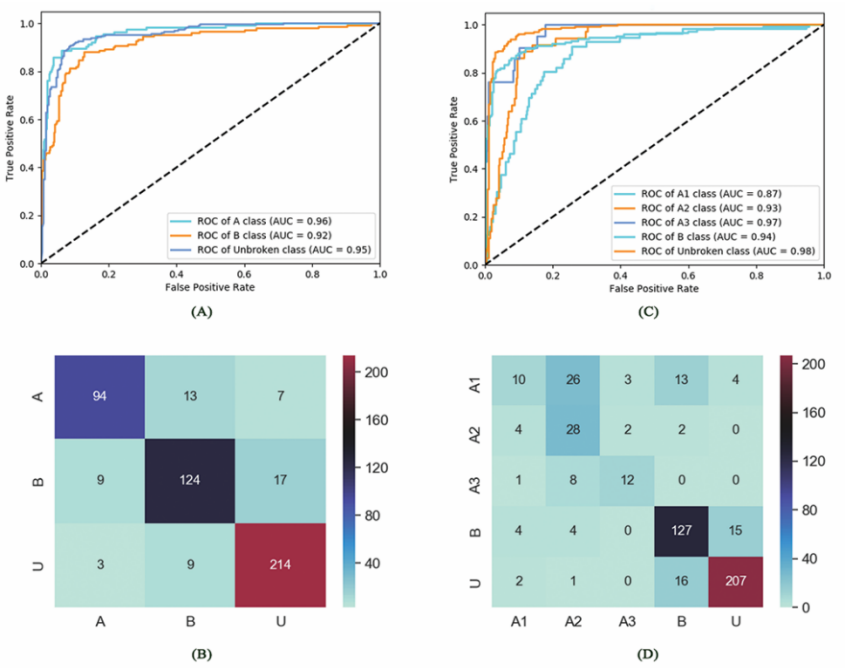


Figure 19 ROC curve with associated AUC (A, C) and confusion matrix (B, D) are shown for both three and five classes classification. The number of images used for testing was: 226 Unbroken images, 150 B, 114 A, 56 A1, 36 A2 and 21 A3.

Multistage Method

Training three InceptionV3 networks for classification between *Broken-Unbroken*, *A-B,* and *A1-A2-A3* with 5-folds cross-validation, resulted in an average test accuracy of 0.91 (CI 0.89-0.93), 0.87 (CI 0.86-0.89) and 0.61 (CI 0.63-0.68) respectively. Values for each fold are shown in Table 5.

Table 5 Accuracy using 5-fold cross validation for the three networks used in the different stages.

K-Fold	*Broken-Unbroken*	*A-B*	*A1-A2-A3*
Fold1	0.90	0.88	0.58
Fold2	0.92	0.87	0.54
Fold3	0.90	**0.89**	0.62
Fold4	**0.93**	0.86	0.62
Fold5	0.90	0.86	**0.68**
Average	0.91	0.87	0.61
Standard Deviation	±0.01	±0.01	±0.04

Building the already described multi-stage architecture, we obtained an average accuracy of 0.86 (CI 0.84-0.88) for three classes' classification and 0.80 (CI 0.77-0.82) for five classes' classification. The other metrics are shown in Table 6**.**

Table 6 Precision, Recall and F1-score for three and five classes classification using the multistage approach. The number of images is proportioned to the initial dataset unbalance. Improvements for five classes classification are shown in parenthesis.

Class	Precision		Recall		F1-score		# of images for testing
	3 classes	5 classes	3 classes	5 classes	3 classes	5 classes	
Unbroken	**0.91** (0.89-0.92)	**0.93(↑0.02)** (0.92-0.94)	**0.90** (0.86-0.93)	**0.90(↑0.02)** (0.86-0.93)	**0.90** (0.88-0.92)	**0.91(↑0.02)** (0.89-0.93)	226
B	**0.80** (0.76-0.84)	**0.85(↑0.07)** (0.82-0.87)	**0.83** (0.80-0.86)	**0.83 (=)** (0.80-0.86)	**0.82** (0.81-0.83)	**0.84(↑0.03)** (0.83-0.85)	150
A	**0.84** (0.79-0.89)		**0.82** (0.73-0.90)		**0.83** (0.78-0.87)		114
A1		**0.50(↑0.07)** (0.46-0.53)		**0.53(↑0.24)** (0.45-0.61)		**0.51(↑0.17)** (0.47-0.55)	56
A2		**0.45(↑0.03)** (0.35-0.55)		**0.56(↑0.06)** (0.40-0.72)		**0.49(↑0.05)** (0.40-0.58)	36
A3		**0.70(↑0.04)** (0.51-0.88)		**0.56(↓0.01)** (0.45-0.68)		**0.62(↓0.02)** (0.50-0.73)	21

Notice that the performance for three classes does not change too much. On the other hand, we had important improvements for five classes'' classification. However, these improvements were still not optimal,, and we wanted to go further with *A1*, *A2,,* and *A3* subclasses. For this approach,, we trained the last two networks with an increased dataset composed of new images of *A1*, *A2,,* and *A3* fractures, resulting in a total of 495 *A1*, 293 *A2,,* and 170 *A3*. With the help of the specialists, we noticed that the main problem was distinguishing between *A1* and *A2*. This was confirmed by running a binary network to classify between *A1* and *A2* together against the *A3* class, resulting in an average accuracy of 0.92 (CI 0.89-0.95). Thus, we added this step and a successive binary network to distinguish between *A1* and *A2,* which reached an average accuracy of 0.68 (CI 0.65-0.70). The full pipeline is shown in Figure 20**.** Adding a new stage to the computation, we obtained an accuracy of 0.81 (CI 0.79-0.82), and values of precision, recall, and F1-score increased, as shown in Table 7**.**

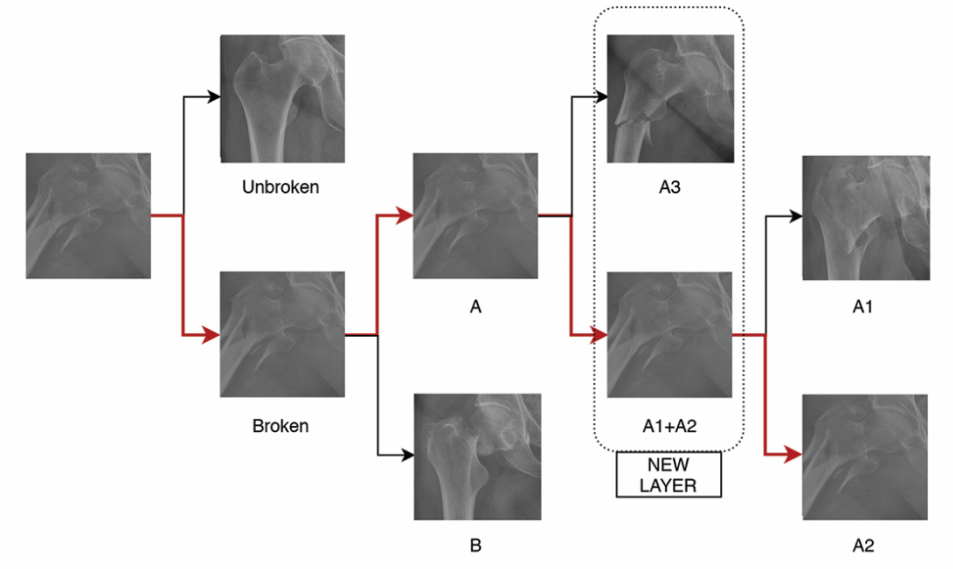


Figure 20 Full pipeline adding a new step of classification.

Table 7 Precision, Recall and F1-score for three and five classes classification using the multistage approach with 4 stages. Values for five classes classification increased, especially for A3, compared to the three stages approach, as shown in parenthesis.

Class	Precision	Recall	F1-score	# of images for testing
Unbroken	**0.93 (=)** (0.92-0.94)	**0.90 (=)** (0.86-0.93)	**0.91 (=)** (0.89-0.93)	226
B	**0.85 (=)** (0.82-0.87)	**0.83 (=)** (0.80-0.86)	**0.84 (=)** (0.83-0.85)	150
A1	**0.49 (↓0.01)** (0.45-0.54)	**0.54 (↑0.01)** (0.37-0.70)	**0.51 (=)** (0.42-0.60)	56
A2	**0.50 (↑0.05)** (0.41-0.58)	**0.55 (↓0.01)** (0.39-0.71)	**0.51 (↑0.02)** (0.43-0.60)	36
A3	**0.73 (↑0.03)** (0.54-0.93)	**0.73 (↑0.17)** (0.64-0.82)	**0.73 (↑0.11)** (0.62-0.84)	21

Visualization

In order to demonstrate that the network was actually learning, we implemented Grad-CAM, one variant of CAM [131] broadly applicable to any CNN-based architectures, which allowed us to visualize where the network was focusing for the different classifications. Some examples of Grad-CAM heat maps are shown in Figure 21 related to different types of fracture. This visualization confirmed that the network is focusing on the correct area of the bone: for *A* prediction the network focused on the lower part of the proximal femur, and for *B* prediction around the

neck. Unfortunately, for *A1, A2,* and *A3* subgroups we were not able to find a recurrent pattern, a clear sign that this classification still needs to be improved.

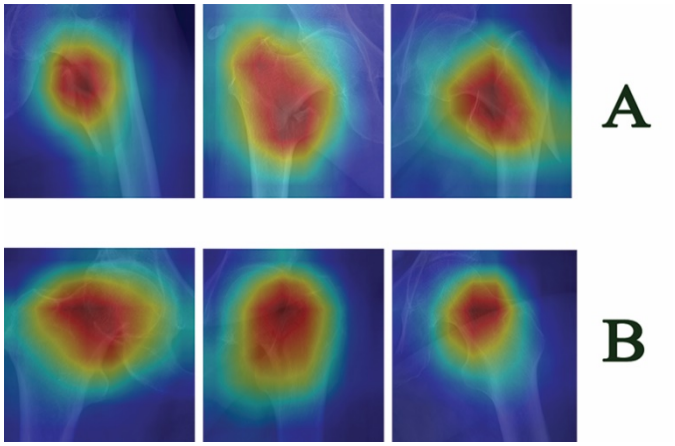


Figure 21 Grad-CAM output for A and B type fractures classification.

CAD-system

Firstly, one specialist with 5 years of experience and one orthopedic student with 1 year of experience evaluated the type of fracture of 150 bone images without the help of the neural network, resulting in an accuracy of 0.73 and 0.68, respectively. As already stated, we decided to use two raters with different levels of experience to demonstrate that this tool could also be useful for educational purposes. It should be mentioned that the specialist participated in the creation of the ground truth classification together with a senior surgeon, although the evaluation for this study was conducted one year after the ground truth was created. Fourteen days later, to produce unbiased results, we gave them the prediction of the neural networks in cascade, which obtained, with this particular set of images, an accuracy of 0.84, and the probability that the neural network assigned to each class. With the help of the CAD, the accuracy increased to 0.86 and 0.83, respectively, resulting in an average improvement of 14% in accuracy and a considerable reduction in time spent on evaluation. We then computed Cohen's Kappa scores, shown in Table 8, to observe the inter-agreement between the neural network and the raters with and without the CAD system's help. We could notice in the first row that the agreement between the neural network and the raters improved by 0.17 and 0.40 with and without the CAD system. This demonstrated that both of them changed their predictions accordingly to the neural network predictions, especially the student. Plus, we highlighted in red the agreement of the raters with and without the CAD system:

from these values, we could notice how the student changed his predictions accordingly to the neural network, resulting in a lower Cohen's Kappa (0.60) compared to the one of the specialists (0.73). In the end, from the values emphasized in green, we could see how the agreement between the specialist and the student increased using the CAD system.

Table 8 Cohen's Kappa scores to measure inter-agreement. Values between 0.41 and 0.60 indicate a moderate agreement, between 0.81 and 0.80 a substantial agreement and between 0.81 and 0.99 an almost perfect agreement.

Cohen's Kappa	Specialist No CAD	Specialist CAD	Student No CAD	Student CAD
Neural Network	**0.60**	**0.77 (↑0.17)**	**0.48**	**0.88 (↑0.40)**
Specialist No CAD		0.73	0.50	0.66
Specialist CAD			0.57	0.76
Student No CAD				0.60

Discussion

We developed a system, based on the AO/OTA classification, for proximal femur fracture classification, easy to extend to other bones in the human body. Using transfer learning with the InceptionV3 architecture, we reached an accuracy of 0.87 (CI 0.85-0.88) for fracture classification in types *A, B,* and *Unbroken* and 0.78 (CI 0.76-0.79) for types *A1, A2, A3, B, Unbroken*. We used these results as a baseline and proposed a new method to increase the accuracy, especially for sub-classes, using different subsequent networks in cascade. This second approach is perfectly suitable for AO/OTA hierarchical classification and also allowed us to work on sub-fractures, such as *A1, A2,* and *A3* and we obtained a 0.80 (CI 0.77-0.82) test accuracy for five classes classification, 2% more than the previous approach. We then add a new stage of classification to increase the performance to 0.81 (CI 0.79-0.82). We finally showed the potential of this tool for helping in diagnosis with an improvement of 14%. For the specialist, the accuracy with the CAD system's help also surpassed the accuracy of the neural network alone, demonstrating the usefulness of a combined work between specialists and CAD system. For the students, we had an important improvement in accuracy, and the help of the CAD system was also demonstrated by the growth of Cohen's Kappa coefficient. These results underlined the utility of this tool both for practical and educational purposes.

Nonetheless, many challenges need to be undertaken in order to overcome the limits of this approach.

First of all, the process is still semi-automated. An automatic detection phase of the area related to the femur would solve this issue. Secondly, given the intrinsic cascade structure, the error propagates through the subsequent layer. Thirdly, the dataset is still scarce, and we could not extend the work for B sub-fractures. In the next approach, we solved the first two problems by introducing a Vision Transformer and a YOLO network for object detection. Moreover, we extended the dataset and introduced the B sub-fractures into the classification.

2.1.2 Second approach: Vision Transformer

Introduction

In the vision domain, after the introduction of AlexNet [132] in the ImageNet competition in 2017, the applications of Convolutional Neural Networks (CNNs) have been increasing their ability to capture the spatial dependencies in an image. Nevertheless, CNNs have different limitations.

In recent times, a new paradigm called Transformer, introduced formerly for Natural Language Processing (NLP) [133], has demonstrated exemplary performance on a broad range of language tasks, by means of BERT (Bidirectional Encoder Representations from Transformers) [134] or GPT (Generative Pre-trained Transformer) [135]. Transformer architectures are based on a self-attention mechanism that learns the relationships between the elements of a sequence and 1) can deal with complete sequences, thus learning long-range relationships, 2) can be easily parallelized, 3) can be scaled to high-complexity models and large-scale datasets. The discovery of Transformer networks in the NLP domain has aroused great interest in the computer vision community. However, visual data follow a typical structure, thus requiring new network designs and training schemes. As a result, different authors have proposed their implementation of a Transformer model applied to vision, obtaining great results in object detection [136], segmentation [137], video analysis [138], image generation [139], and many more.

As already discussed in Section 1.1.1, recently, in the field of femur fracture classification, impressive results have been obtained using CNNs. Unfortunately, the majority of the existing works regarding fractures classification focused mainly

on the binary classification between *Broken* and *Unbroken* bones [16,18,19], which has a low impact on doctor diagnosis. To the best of our knowledge, pure CNNs have been applied to classify different types of fractures just in two previous papers [27,29]. More complex architectures were applied in [140], where an encoder with an LSTM (Long Short Term Memory) layer obtained very good results even in sub-fracture classification, but using also radiology reports which are not always easy to obtain, and in [141], where the authors used spatial transformer and a CNN to localize and classify the fractures. Nevertheless, results are still non-optimal, especially for complex fractures, and a generalized approach does not exist yet.

The novelties introduced by this work compared to the cascade approach are four-fold:

1) the largest and richest labeled dataset ever for femur fractures classification was used, with 4207 images divided into 7 different classes;
2) the Vision Transformer (ViT) implementation by Google [142] was applied for the classification task, surpassing the two baselines of a classic InceptionV3 [14] network and a hierarchical network proposed in [110]. This is the first work where CNNs were not involved in the classification pipeline;
3) the attention maps of ViT were visualized and the output of the Transformer's encoder was clustered in order to understand the potentiality of this architecture;
4) a final evaluation was carried out, asking 11 specialists to classify 150 images using an online survey, with and without the help of our system.

Materials and Methods

Patients selection and Dataset

This retrospective study was conducted in a Level-I trauma center and was approved by the appropriate Ethics Review Board. All >18-year-old patients with proximal femur fractures in the Emergency Department between January 2013 and December 2020 were eligible for enrollment. Exclusion criteria included missing pelvic anteroposterior radiographs documenting the hip fracture on the hospital's PACS and pathologic fractures. Demographic data, pelvic anteroposterior radiographs with related radiological referral, and intraoperative diagnosis were recorded in a computerized dataset. All data were collected anonymously using Synapse 3D (FUJIFILM Corporation) and each image was carefully examined for removal of all labels. Table 9 describes patients' baseline characteristics. The initial

dataset was labeled by a senior trauma surgeon with 18 years of experience, a specialist who has worked specifically on femur fractures in the past 6 years and was composed of *n=2645* images of the entire or half hip bone. The dataset was then reviewed using the included radiological referrals and intraoperative diagnosis. All images with discordant classification were analyzed and removed if univocal interpretation by the whole team was not attained. Then segmentation of the images into left and right hips was performed. The second step was a cleaning phase, where a total of *n=242* images were excluded because they contained prosthesis (*n=97*), had poor lighting condition (*n=47*), showed the area around the femur partially hidden (*n=23*), or presented a lateral view (*n=72*). Moreover, because of the low number of *C* fractures (*n=3*) detectable on pelvic radiographs, this class was excluded. The third step was a cropping phase, where the areas related to the right and left femur were selected through a fully automated cropping method and resized to $224x224$, since some images present only one between the right and left femur. This technique concerned the use of the YOLOv3 (You Only Look Once) [143] algorithm for the detection of left and right femurs. The fourth step was the revision of the YOLOv3 errors: *n=25* femurs were not detected, *n=1845* right femurs were correctly detected while *n=208* wrongly detected as left, *n=1874* left femurs were correctly detected while *n=241* wrongly detected as right. After the correction and the manual cropping of the not detected images, the final dataset was composed of *n=2152* left femurs and *n=2055* right femurs, which have been flipped horizontally. The fractures were then divided into different types, and, afterward, the dataset was reviewed by two radiologists from our medical team to confirm the validity of the ground truth. The final number of images was *n=4207* manually annotated images divided into different fracture types: 2003 *Unbroken* femurs, 631 type *A1*, 329 type *A2*, 174 type *A3*, 625 type *B1*, 339 type *B2*, 106 type *B3*. This process is shown in Figure 22 following the STARD 2015 flow diagram [144].

Table 9 Baseline characteristics. Median computed with Interquartile Range (IQR)

		Total (n=2645)
Age (*yr*)	Median (IQR)	81 (73-86)
Sex	F	1785
	M	860
	%F	67.5%

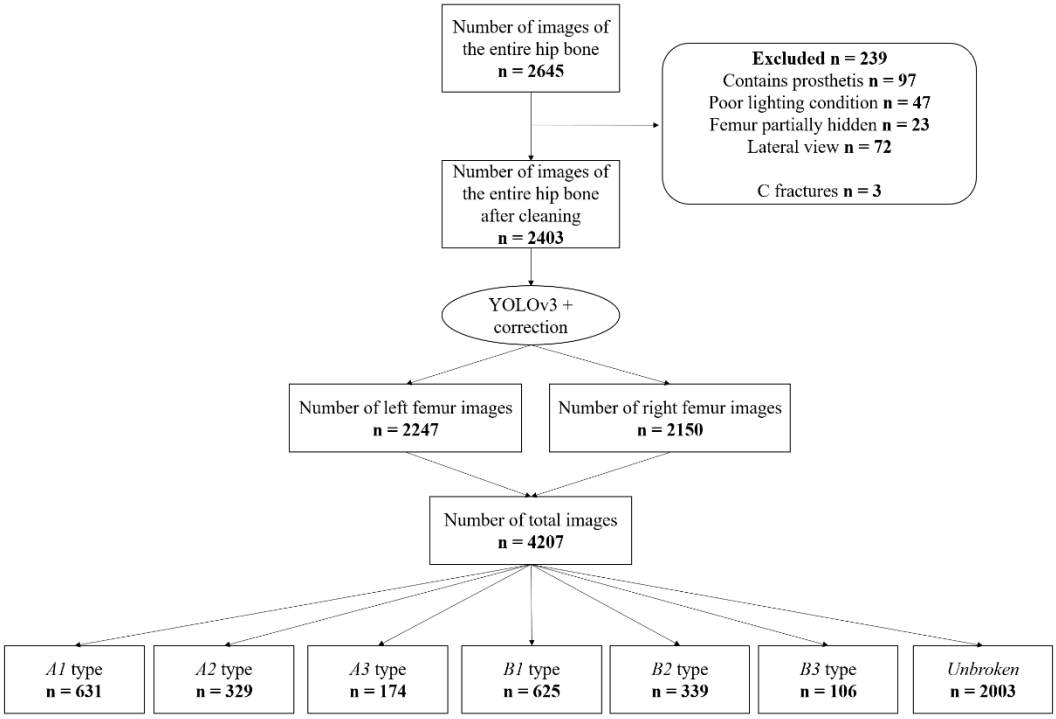


Figure 22 The STARD 2015 Flow Diagram, which shows the dataset processing workflow

ViT Configuration

In this work, the ViT proposed in [142] was applied, one of the first vision solutions leveraging self-attention. To handle image data, this architecture focuses on small patches which are flattened using a linear projection matrix, and a positional embedding is added to it. The Transformer encoder, similar to the original 2017 version [133], consists of multiple blocks of self-attention, normalization, and fully connected layers with residual connections. In each attention block, multiple heads can capture different connectivity patterns. The fully connected Multi-Layer Perceptron (MLP, [4]) head at the classification output provides the desired class prediction.

As stated in the original ViT paper, this network typically requires a larger dataset than usual, as well as a longer pre-training schedule. For this reason, with only 4207 images it would be unfeasible to train ViT from scratch. As a first solution, a Compact Convolutional Transformer [145] was applied, a very recent architecture based on ViT, where the patch extraction phase is substituted by a CNN, which took care of the feature extraction. This solution is proven to usually overcome the big data problem, but, unfortunately, it did not apply to this case. For this reason, the

four pre-trained ViT architectures were tested. In [142], the proposed configurations depend on several network parameters (such as the number of neurons of a specific layer) and patch number. After experimenting with the so-called *base-16*, *base-32*, *large-16,* and *large-32* configurations, the *large-16* ViT block was selected, which has a multilayer perceptron of 4096 units, 16 heads, 24 layers, a hidden size of 1024, and operates with $16x16$ patches. The comparison between these four configurations and the CCT is shown for completeness in Table 10.

Table 10 Values of precision, recall, and F1-score for the Compact Convolution Transformer and the base-16, base-32, large-16, large-32 configurations. The values are shown with related confidence intervals.

Architecture	Precision	Recall	F1-score
B16	0.77 (CI 0.67-0.88)	0.74 (CI 0.59-0.88)	0.75 (CI 0.63-0.87)
B32	0.67 (CI 0.51-0.83)	0.65 (CI 0.48-0.83)	0.65 (CI 0.49-0.81)
L16	0.77 (CI 0.64-0.90)	0.76 (CI 0.62-0.91)	0.77 (CI 0.64–0.89)
L32	0.71 (CI 0.59-0.83)	0.65 (CI 0.48-0.82)	0.66 (CI 0.53-0.80)
CCT [145]	0.39 (CI 0.18-0.59)	0.38 (CI 0.12-0.65)	0.38 (CI 0.15-0.60)

A dense layer with 4096 neurons was then added to this block, with a GELU (Gaussian Error Linear Units) activation followed by a batch normalization layer and a dropout layer with 0.5 as *keep-in* parameter. Finally, a Softmax layer for 7-class classification was attached. The learning rate was initially set to $1e^{-4}$ and reduced by a factor of 0.2 after 4 epochs of plateau until a minimum of $1e^{-6}$. The optimizer used was the Rectified Adam, and the loss function categorical crossentropy.

To cushion the problem of class imbalance, three diverse methods have been tried: the first assigned different weights to each class during training, where the weights are inversely proportional to the number of samples in the respective class; the other ones applied oversampling or data augmentation, with a rotation range of 10 degrees, and both height and width shift from 0.0 to 0.1 fraction of total height or width, in order to obtain the same number of samples for each class. After testing, oversampling resulted in the most performing choice. The problem with data

augmentation is the fact that more complex transformations as shearing, cannot be used, as it may lead to the generation of "fake" fractures.

To compare the results with the state of the art, two other approaches were used as baselines. The first is an InceptionV3 network, which has been chosen for the results obtained in Section 2.1.1. The second is the cascade approach, proposed again in Section 2.1.1. Even if we already showed the results in the previous section, we re-trained each approach with this updated dataset to make a fair comparison.

Visualization and Clustering

To understand where the ViT network was focusing during inference, a function was implemented in order to visualize the attention map. Then, unsupervised learning was used to evaluate the ability of the ViT encoder to extract features. Three clustering approaches were tested: firstly, the initial dataset of images resized to $224x224$ was clustered using a Convolutional Autoencoder, and the results were used as a baseline. Secondly, the Convolutional Autoencoder was substituted by an Autoencoder, which took as input a vector of 1024 values, extracted in one case from the InceptionV3 network and the other from the ViT encoder. In all three cases, the Autoencoder was pre-trained for 200 epochs. After this, the encoders were extracted from the three architectures, and a clustering layer was added at the end. This layer was initialized with the centers found by the *kmeans++* function and trained until convergence. It outputs a vector that represents the probability, calculated with Student's t-distribution, of the sample belonging to each cluster. The clustering performance was measured with *accuracy, Normalized Mutual Information (NMI)* between the ground truth and the predicted distribution, where 0 means no mutual information and 1 means a perfect correlation, *Adjusted Rand Index (ARI),* which computes a similarity measure by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clustering, and *loss*.

Training, Framework, and Evaluation

From the initial dataset, 15% of images for each class were kept apart for testing, resulting in a test set of 91 images of type *A1*, 94 type *A2*, 25 type *A3*, 90 type *B1*, 49 type *B2*, 16 type *B3*, and 282 *Unbroken* femurs, and 15% for validation. The remaining 70% of the images were used for training. The networks were trained for 40 epochs using *Early Stopping* with a patience of 10 epochs. We used Keras [128], an open-source neural-network library written in Python, running on top of TensorFlow [129], on Windows 10 Pro with NVIDIA Quadro RTX 6000. For each

network, the macro accuracy was computed. Then, the performance for single classes was measured using precision, recall, and F1-score. The performance of the specialists with and without the system was computed using an online survey (Forms, Microsoft Corporation, Redmond USA) and was measured using accuracy. In this case, the accuracy was a reliable metric as the dataset used was balanced. Firstly, 11 specialists (7 residents and 4 radiologists) evaluated 150 hips without the help of the neural network. The CAD tool was designed to suggest the fracture classification for the proposed image, the confidence level in percentage points, and the attention map. This set of images was taken from the test dataset and, therefore not involved in the training process, for obtaining comparable results. Fourteen days later, to produce unbiased results, the evaluation was made again, but this time specialists could consult the prediction of the ViT and the probabilities that the network assigned to each class.

Results

Baseline Methods

The aforementioned InceptionV3 model was able to correctly classify 67% of the images and obtained an average precision of 0.57 (CI 0.42-0.72), recall of 0.53 (CI 0.33-0.72), and F1-score of 0.54 (CI 0.36-0.71). The hierarchical network composed of different InceptionV3 networks in cascade was able to correctly classify 61% of the images and obtained an average precision of 0.44 (CI 0.21-0.68), recall of 0.41 (CI 0.14-0.69), and F1-score of 0.40 (CI 0.15–0.64). These values are shown in Table 11 (a) and (b).

Table 11 Values of precision, recall, and F1-score for the two baselines (a) and (b) and for the ViT (c). The values are shown with related confidence intervals. In (c), the improvement given by the ViT compared to the highest value among the two baselines is shown in parenthesis

CNN	Precision	Recall	F1-score	# Images
A1	0.42	0.53	0.47	91
A2	0.65	0.43	0.51	94
A3	0.65	0.60	0.63	25
B1	0.59	0.61	0.60	90
B2	0.43	0.45	0.44	49
B3	0.43	0.19	0.26	16
Unbroken	0.85	0.89	0.87	282
Macro AVG	0.57 (CI 0.42-0.72)	0.53 (CI 0.33-0.72)	0.54 (CI 0.36-0.71)	

(a)

Hierarchical CNN	Precision	Recall	F1-score	# Images
A1	0.36	0.51	0.42	91
A2	0.50	0.21	0.30	94
A3	0.20	0.32	0.24	25
B1	0.51	0.70	0.59	90
B2	0.59	0.20	0.30	49
B3	0.11	0.06	0.08	16
Unbroken	0.87	0.88	0.87	282
Macro AVG	0.44 (CI 0.21-0.68)	0.41 (CI 0.14-0.69)	0.40 (CI 0.15–0.64)	

(b)

ViT	Precision	Recall	F1-score	# Images
A1	0.66 (↑24%)	0.66 (↑23%)	0.66 (↑19%)	91
A2	0.77(↑12%)	0.66 (↑23%)	0.71 (↑20%)	94
A3	0.92 (↑30%)	0.92 (↑32%)	0.92 (↑29%)	25
B1	0.74 (↑15%)	0.93 (↑23%)	0.82 (↑22%)	90
B2	0.79 (↑ 20%)	0.69 (↑24%)	0.74 (↑30%)	49
B3	0.56 (↑13%)	0.56 (↑37%)	0.56 (↑30%)	16

Unbroken	0.95 (↑8%)	0.94 (↑5%)	0.95 (↑8%)	282
Macro AVG	0.77 **(↑20%)** (CI 0.64-0.90)	0.76 **(↑23%)** (CI 0.62-0.91)	0.77 **(↑23%)** (CI 0.64–0.89)	

(c)

ViT

Our configuration of the ViT was able to correctly classify 83% of the entire test dataset. The average values of precision, recall, and F1-score obtained were 0.77 (CI 0.64-0.90), 0.76 (CI 0.62-0.91), and 0.77 (CI 0.64–0.89), respectively. These values are shown in Table 11 (c).

Clustering

The results for the three clustering in terms of accuracy, NMI, ARI, and loss are shown in Table 12Errore. L'origine riferimento non è stata trovata., while a graphical representation of the confusion matrix and distribution of each approach is shown in Figure 23.

Table 12 Values of accuracy, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and loss for the three clustering approaches

Input	Architecture	Accuracy	NMI	ARI	Loss
Images	Convolutional Autoencoder + Clustering Layer	0.22	0.01	0.01	0.11
Feature vector from CNN	Autoencoder + Clustering Layer	0.36	0.22	0.22	0.17
Feature vector from ViT encoder	Autoencoder + Clustering Layer	0.78	0.61	0.72	0.04

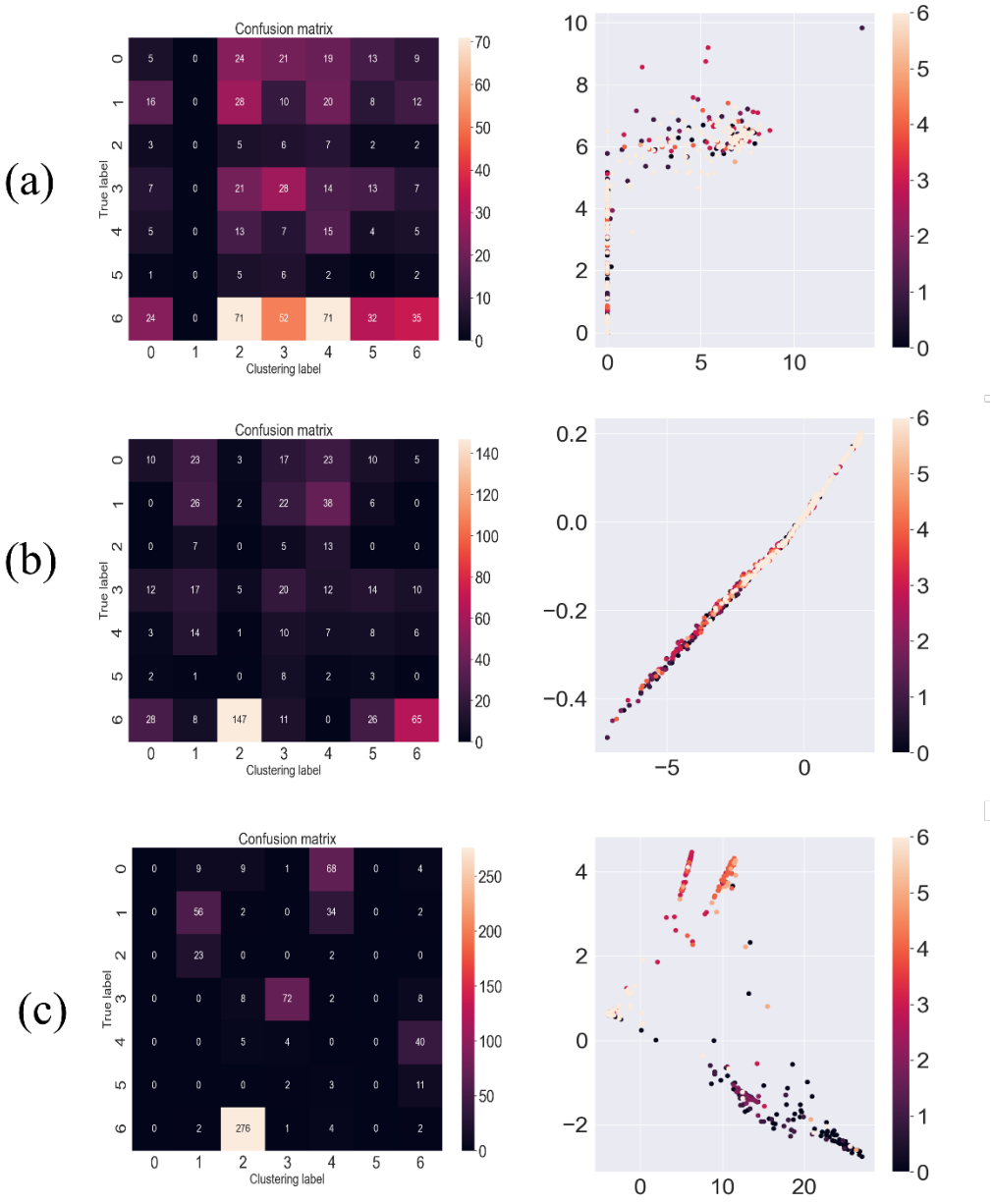


Figure 23 Confusion matrices and distributions of the different clustering approaches. They were obtained with a Convolutional Autoencoder applied directly on the images (a) or with an Autoencoder applied to the feature vector extracted from the CNN (b) and extracted from the ViT Encoder (c). In the right images, the cluster labels are shown with the colors presented in the sidebar

CAD-system

The 7-class evaluation of the type of fracture present in 150 images without the help of ViT, performed by 11 specialists and resumed in Table 13, resulted in an average accuracy of 0.58 (CI 0.53 – 0.65) for residents and 0.84 (CI 0.77 – 0.92) for

radiologists. Fourteen days later, the same test was performed with the help of ViT, which with this particular set of images, obtained an accuracy of 0.88 (CI 0.78-0.97). In this case, the accuracy of both residents and radiologists increased to 0.96 (CI 0.92 – 0.99) and 1.00, respectively. The result is an average improvement of 0.29 (CI 0.12 – 0.37) in accuracy.

Table 13 Values of accuracy for the 11 specialists who performed the evaluation with and without the CAD (Computer Assisted Diagnosis) system. The average values are shown with related Confidence Interval (CI)

Specialist	Years of experience	Accuracy without CAD	Accuracy with CAD	Accuracy Improvement
Resident #1	2	0.55	0.90	0.35
Resident #2	1	0.55	0.89	0.34
Resident #3	2	0.53	0.98	0.45
Resident #4	4	0.69	1.00	0.31
Resident #5	3	0.63	0.98	0.35
Resident #6	2	0.53	0.96	0.43
Resident #7	3	0.64	0.98	0.34
Radiologist #1	10	0.81	1.00	0.19
Radiologist #2	15	0.90	1.00	0.10
Radiologist #3	7	0.80	1.00	0.20
Radiologist #4	13	0.87	1.00	0.13
Residents' Average		0.58 (CI 0.53 – 0.65)	0.96 (CI 0.92 – 0.99)	0.37 (CI 0.32 – 0.42)
Radiologists' Average		0.84 (CI 0.77 – 0.92)	1.00	0.15 (CI 0.08 – 0.23)
Total Average		0.68 (CI 0.59 – 0.78)	0.97 (CI 0.94 – 1.00)	0.29 (CI 0.12 – 0.37)

Discussion & Conclusion

This work is summarized in Figure 24.

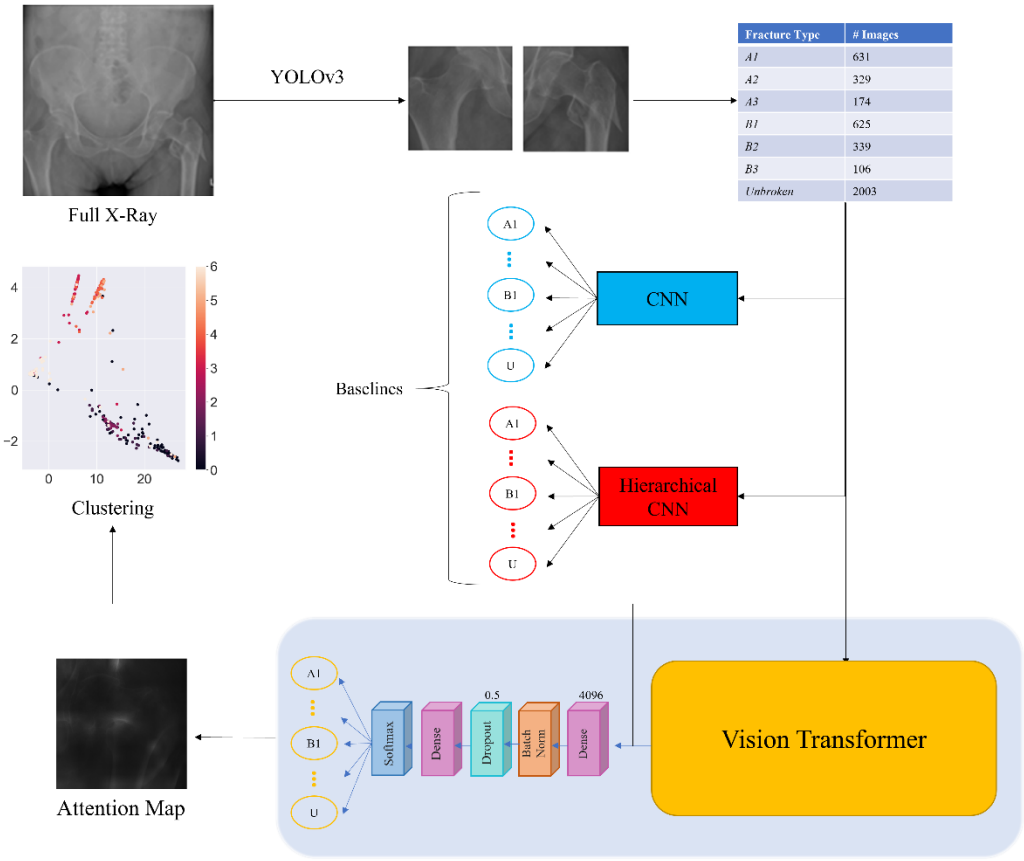


Figure 24 Full pipeline of this work. The images were first passed through a YOLOv3 network which cropped the areas related to the left and right femur. A CNN and a hierarchical CNN were then applied to this dataset and the results were used as baselines. Then, a modified ViT encoder was used for 7-class classification and the attention maps were analyzed. Finally, clustering was performed to evaluate the ability of the ViT to extract features

Firstly, the original images were cropped and ordered using a YOLOv3 algorithm, obtaining a vast dataset of 4207 images divided into 7 classes. Two baselines were defined using InceptionV3 and a hierarchical network which were able to correctly classify 67% and 61% of the test samples, respectively. A ViT-based architecture was then applied to correctly classify 83% of the test images. The average value of accuracy, precision, recall, and F1-score improved by a factor of 25%, 20%, 23%, and 23% compared to the best result among the two baselines. For the first time in literature, an architecture that does not involve CNN was able to obtain good performance in sub-fracture classification. The attention maps (Figure 25), after being analyzed by a team of specialists, showed how the network focuses on the correct areas during inference: the calcar and trochanteric area for the *A* class, the neck of the femur, and the cortex of the greater trochanter for the *B* class, and

alongside the whole cortex profile of the proximal femur for the *Unbroken* class. A clustering phase was also used to demonstrate how better the ViT extracts a feature vector compared to the other two approaches, even if it still struggles with some sub-fractures, as shown in Figure 23 (c), where *A*, *B,* and *Unbroken* class were correctly clustered, but *A1*, *A2*, *A3* and *B1*, *B2*, *B3* were often mismatched. Nevertheless, the accuracy of this clustering was 0.78 compared to 0.22 and 0.36 of the other two, underlying once again the improvement given by ViT. Finally, in Table 13**,** is shown the evaluation of 7 residents and 4 radiologists, with or without the ViT, which improved by a factor of 37% (accuracy: 0.96) and 15% (accuracy: 1.00), respectively. These values seem excessive, but it has to be considered the fact that the specialist also had access to the probabilities returned by ViT, related to each class. This information allowed them to focus much more on the images with uncertain probabilities and to also consider the second class predicted by ViT, in order of probability. On top of that, the best results were achieved through the synergic effect between physicians and the CAD system, resulting in even better than the physicians or the algorithm alone. As already stated in [146], the performance of algorithms should be complementary to that of doctors rather than comparable or better with respect to them.

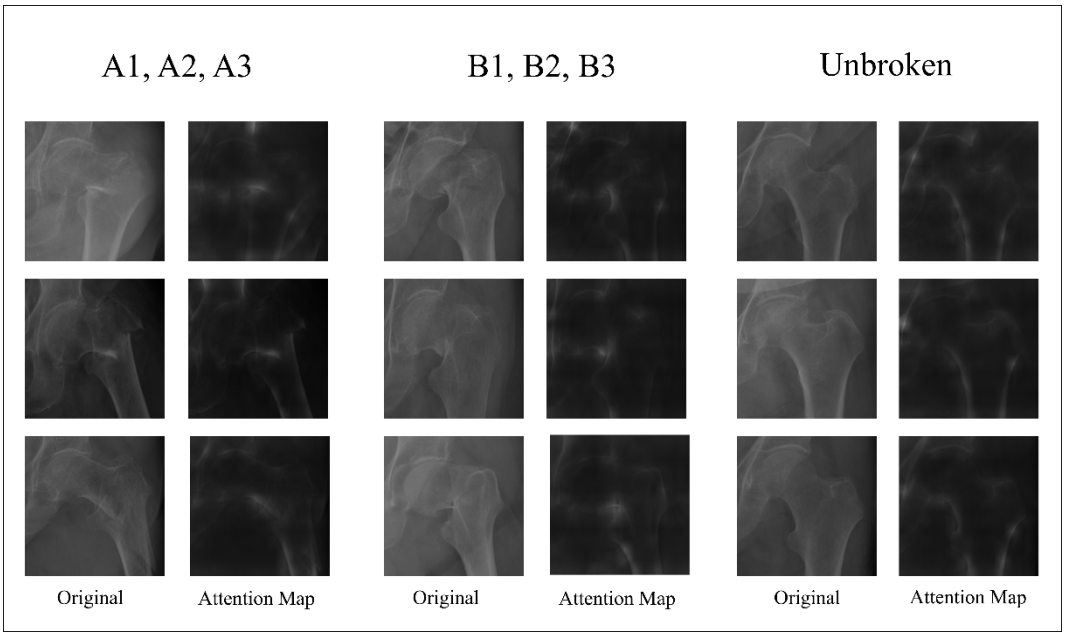


Figure 25 Some examples of original images and corresponding attention map for each class. The area in which ViT is focusing during inference is highlighted in white

Summarizing, the results achieved in this work using a Vision Transformer outperformed the previous approach based on a cascade of CNN, with a dataset

eight times and four times larger, respectively. This aspect may seem a negative side since it is always better to achieve certain results with as few images as possible. However, given the variety and diversity of fracture patterns, a larger dataset could bring a better generalization.

On the other hand, we ignored the intrinsic hierarchical structure of the dataset and the performance of the ViT could massively improve by adapting its architecture to the hierarchical structure of the AO/OTA classification.

In the next section, we discuss the next approach that deals in general with hierarchies and could also be applied to our task.

2.1.3 Third approach: Hierarchies

In this work, we focused on the general problem of hierarchies and utilized our dataset as a case study. This work was carried out under the supervision of Prof. Grandvalet, a senior researcher at UTC Sorbonne Universitè, during a visiting research period in his laboratory.

Introduction

“The richer the data, the better the outcomes” is one of the core tenets of the deep learning universe. Indeed, the proportional improvement of performances with the number of samples in a dataset has been demonstrated by a large number of studies [147–149]. On the other side of the medal, data scarcity can be a significant problem in many applications. Over the year, different solutions have been introduced to overcome this problem. For example, transfer learning [150], which allows extending the knowledge learned on a large dataset to a smaller one, is nowadays the de-facto solution for this type of issue.

Nevertheless, the exponential increase of data over the last few years and the efficiency of technique such as transfer learning has pushed the deep learning research community to shift its focus toward building architectures suited to harness this enormous amount of data, ignoring their performances in the low-data regime. However, 1) transfer learning is not always the most performing approach, as the features learned from a generic dataset might not be well suited for a specific fine-grained task, and 2) not all domains have seen an increase in the number of samples big enough to train a network from scratch. For example, the medical field continues to be a domain where data is very limited and valuable. One solution in

this context has been to adapt complex architectures to a reduced data regime, ignoring the potentiality of data proprieties. This approach can be extended to the majority of the most recent works on vision [137,142].

The idea of this work is different. Instead of adding complexity to architectures to make them suitable for low-data regimes, we focused on defining a straightforward technique that leverages the inherent characteristics of the data and can be applied to any architecture. Among the many intrinsic features that a dataset could have, the most common example is probably hierarchy. In hierarchical supervised classification (HSC), different classes are related on the basis of a tree hierarchy representing the relationships among them [151]. Thus, we can consider the classes at the finest level as the leaves of a tree. Each class belonging to the same group can be grouped into coarser super-classes (e.g., cats and tigers are both felines) until reaching the root node. We do this under the assumption that classes belonging to the same category are more similar, e.g., pears and apples can be grouped as fruits, while dogs and wolves as animals, and it is often easier to distinguish a wolf from an apple than a dog. This property indicates that the supervised classification can be performed hierarchically, instead of treating all classes in a flat way. When tackling HSC, a classifier should give importance not only to the finest level but also to the coarser categories. The majority of the existing approaches have proposed ad hoc architectures to leverage the hierarchy, e.g., networks with intermediate prediction [152] or cascades of classifiers trained separately [110]. These approaches can be effective, but they are mostly architecture-dependent.

In contrast, our approach is to redefine the objectives of learning, regardless of the model, by defining a learning criterion that can be applied to any architecture with a final softmax layer. This criterion includes a data-fitting term that is simply a weighted sum of losses at different granularity levels of classification, as well as a regularization term that aims to ensure that classes that are close in the class hierarchy are also close in the softmax layer's internal representation.

We experimented these ideas with InceptionV3, ResNet50, and ViT-B16 [122,126,142], to test the real efficiency of this approach on already over-tested architectures. To produce comparable results, we worked on the ImageNet-1K, CIFAR100, and FGVC-Aircraft [123,153,154] benchmarks, and we gradually decreased the number of samples for training, to demonstrate the effectiveness of our approach when data are scarce. We also performed Linear Discriminant Analysis (LDA) [155] on the representation extracted by the same network with and without our proposed add-ons, to visualize and demonstrate its efficiency in

extracting hierarchical information. We finally moved on to a specific domain where the data are often poor, the femur fracture dataset that we utilized in this thesis work.

Related Work

Hierarchical supervised classification (HSC) has received substantial attention in computer vision. In contrast to classic image classification, HSC attempts to classify the objects into a hierarchy of categories, with each level of the hierarchy corresponding to a level of conceptual abstraction. The standard multiclass approach ignores the class hierarchy and fits a ”flat” classifier, which implicitly considers that all classes are equidistant. However, hierarchies can convey information that could drastically improve the performances, both by using top-down strategies with different classification nodes or by using the taxonomy information for an appropriate design.

Many classical benchmarks, such as CIFAR100 and ImageNet-1K, are defined by a hierarchical class decomposition. This is also the case for real-world data such as bones or plants [156]. When not known, a possible hierarchy can be inferred in an automated way [125,157,158] or modified during training, for example with pruning [159].

Hierarchies can be very complex, especially in large-scale applications. In these situations, specific approaches are necessary to handle big amount of data [159–162]. However, because it is frequently expensive to gather many labeled images in fine-grained image classification tasks, many researchers concentrated on the small-data regime. Nevertheless, it might be challenging to avoid considerable overfitting when training a deep neural network on a limited dataset.

The two leading solutions to prevent overfitting are data augmentation, used in [163] to create new augmented samples by identifying inherent and simple to annotate hyper-classes in the fine-grained data, and regularization techniques, which once again rely on the hierarchy information to regularize the weights accordingly [161,163–165].

One additional factor to take into account is that real-world datasets have also frequently an imbalanced composition and under-represented classes. Writing designs that can be adjusted to the particular dataset is a viable solution to this problem [166,167]. Giving specific classes more importance or stopping the

prediction as soon as a predetermined threshold is not met [165] have been possible solutions to this issue.

Finally, various alternative approaches proposed theories for data scarcity and imbalance based on the observation that CNN's layers prioritize details by proceeding with levels, therefore, comparisons can be drawn with the definition of hierarchy. In [168] the authors used Recurrent Neural Network on the subsequent internal representation while in [169] experimented with pooling layers to benefit from this parallelism.

Hierarchy could be exploited also by intervening in the training stages [156,165,170], for example, with techniques based on alternating fine and coarse labels or pathbase backpropagation.

Many studies proposed instead specific loss functions to address the issue. Basic approaches employ a series of models in cascade, as the one presented in the first section of this chapter, to handle hierarchies. Nevertheless, this technique is troublesome as the last models only see a small portion of the dataset, and errors accumulate in the subsequent steps. Another popular strategy is to utilize intermediate prediction to specialize different layers in different levels of hierarchy [152] or to rely on matrices to compute dynamic losses at each step, considering the class of the current sample [165,171].

Approach

We describe here the general approach we devised. We first introduce the notation describing the class hierarchy. We then describe the loss and regularizer computed for learning to account for this hierarchy.

Notation

The class hierarchy is represented by a directed rooted tree $T = (V,E)$ whose nodes $v_j \in V$ represent subsets of classes, and where a branch $e_{jj'} = (v_j,v_{j'}) \in E$ indicates that the subset of labels j subsumes the labels j', that is, j is a superclass of j'. Let $p(v_j)$ be the parent node of $v_j \in V$ and $a(v_j)$ be the set of all its ascendants, up to the root of the tree. Similarly, $c(v_j)$ is the set of all children of v_j, and $l(v_j)$ are the descendant *leaves* of v_j and d_j is their number. We use the convention that the node indexing starts from the leaves, so that, in a multiclass problem with K classes, $\forall k \leq K$, v_k is the leaf node is a singleton representing the fine-grained class k.

Let $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ be the training set, where $\mathbf{x}_i \in X$ is an input and $y_i \in \{1,...,K\}$ is its label, and θ be the parameters of the model. The output of the model corresponding to class k is noted $f_k(\mathbf{x}_i;\theta)$. It is obtained by applying the softmax function to the representation of the input $\mathbf{x}_i$ on the last layer, denoted $\mathbf{z}_i$:

Equation 1

$$f_k(\mathbf{x}_i;\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\beta}_k^T \mathbf{z}_i + \alpha_k)}{\sum_{l=1}^K \exp(\boldsymbol{\beta}_l^T \mathbf{z}_i + \alpha_l)}$$

where β_k are the parameters corresponding to the (fine-grained) class k.

Hierarchical Loss

The class hierarchy is taken into account during learning by using a loss that rewards partial correct answers given by the model: if $y_i = k$, all the subsets of classes that include class k are partially correct. Hence, the simplest way to take into account the class hierarchy is to reward all correct partial answers given at each node of the class hierarchy, so that the ancestors of leaf k in the hierarchy could be rewarded to provide this partial answer:

Equation 2

$$\sum_{i=1}^{n} \sum_{v_j \in a(\{y_i\})} \log \left(\sum_{k \in v_j} f_k(\mathbf{x}_i;\boldsymbol{\theta}) \right)$$

As $f_k(\mathbf{x}_i;\theta)$ estimates $p(y_i = k|\mathbf{x}_i)$, $\sum_{k \in v_j} f_k(\mathbf{x}_i;\theta)$ estimates $p(y_i \in v_j|\mathbf{x}_i) = \sum_{k \in v_j} p(y_i = k|\mathbf{x}_i)$. Note that, since the root of the tree is the ancestor of all classes there is a vacuous term in Equation 2.

Since the depth of their labels in the class hierarchy impacts their number of occurrences in the loss, we do not utilize the loss function Equation 2 since it does not make balanced use of examples. By using this loss, the training criterion would be diverted from its original goal and minimize errors on a separate distribution with distorted prior probabilities $P(y = k)$. Instead, we consider a combination of losses that each operates on a partition of the set of classes, thus ensuring a balanced representation of all examples. These partitions are obtained by scanning the tree and producing equivalent trees in which edges represent inclusion instead of strict inclusion, and for which all classes are represented at all depths of the trees.

We consider two types of scanning. The first one scans the tree from the root to the leaves (top-down): a path is appended to each leaf positioned at a non-maximum

depth; the path goes to the maximum depth, with, at each depth, a new node representing the class of the leaf. The second scan goes from the leaves to the root (bottom-up): all leaves are positioned at the maximum depth, and the path to the root is obtained by aggregating the (super)classes as soon as possible, to recreate the superclasses of the original tree at a maximum depth (without creating superclasses that were not present in the original tree). These processes insert nodes between parent and child nodes, with the inserted nodes being the children in the top-down scan and the parent in the bottom-up scan. The set of all these partitions is denoted P = $\{P_1,...P_u,...,P_U\}$ and the loss function is then written as follows:

Equation 3

$$\mathcal{L}(\mathcal{D},\boldsymbol{\theta}) = \sum_{P_u}\sum_{i=1}^{n}\sum_{v_j\in P_u}\mathbf{1}_{v_j}(y_i)\log\left(\sum_{k\in v_j} f_k(\mathbf{x}_i;\boldsymbol{\theta})\right), \quad (3)$$

where $\mathbf{1}_{v_j}(y_i)$ is the indicator function that returns 1 if $y_i \in v_j$ and zero otherwise. For a "flat hierarchy", where each leaf is directly attached to the root, there is a single non-vacuous partition P_1 = {{1}...,{k},...{K}}, with v_k = {k}, and we recover the standard cross-entropy loss.

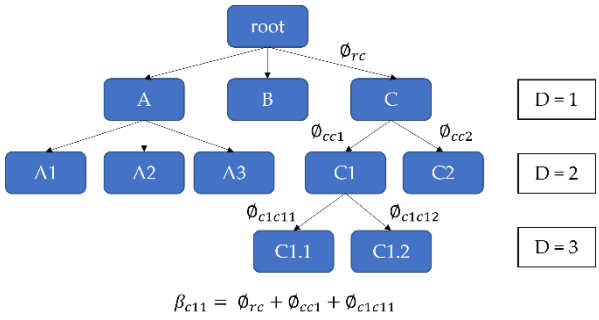


Figure 26 Example of a class structure with parameter decomposition

Practical Implementation

An illustrative example of a tree is shown in Figure 26, where the singleton leaves are labeled {$A1,A2,A3,B,C1.1,C1.2,C2$}, and where the subsets of classes are $C1$ = {$C1.1,C1.2$}, C = {$C1,C2$} and A = {$A1,A2,A3$}. Scanning the tree top-down provides the following partitions: (depth 1) P_1 = {A,B,C}, (depth 2) P_2={$A1,A2,A3,B,C1,C2$} and (depth 3) P_3={$A1,A2,A3,B,C1.1,C1.2,C2$}; scanning the tree bottom-up provides (depth 3) P_3={$A1,A2,A3,B,C1.1,C1.2,C2$},(depth 2) P_4 = {$A,B,C1,C2$} and (depth 1) P_1 = {A,B,C}: the two scans differ here at depth 2. The two trees are always identical at the leaf level and depth 1, but for a larger,

imbalanced original tree, there are more differences between the results of the two scans.

To continue with this example, each of the four partitions P = $\{P_1, P_2, P_3, P_4\}$ provides a component of the loss Equation 3 , each component being the cross-entropy calculated at the granularity corresponding to the class partition. Our approach to computing the loss function is based on matrix computations that make the procedure fast and adaptable to any tree, so that the code can handle any class hierarchy. These matrices, displayed in Figure 27 operate on the softmax outputs f_k in Equation 3 to aggregate the softmax predictions corresponding to each superclass of one of the partitions of P.

Hierarchical Regularization

The knowledge of the class hierarchy is also introduced in the learning process by a second means, through a regularization term applied to the parameters of the final softmax

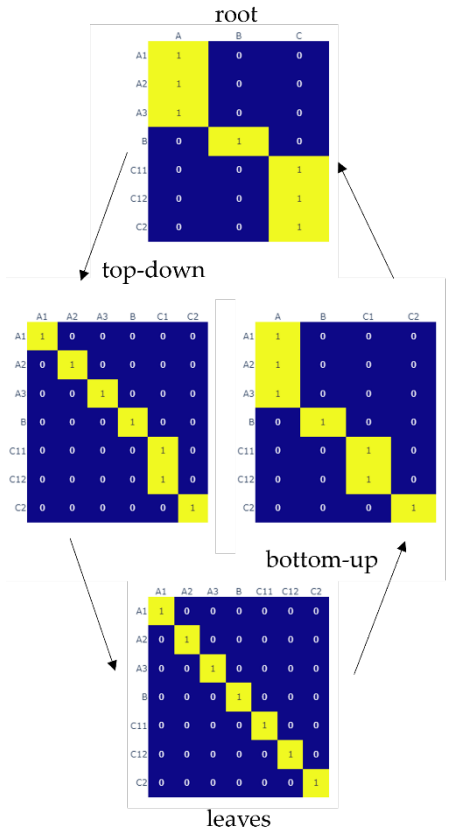


Figure 27 Aggregation matrices computed during the topdown and bottom-up scans of the tree

layer. This regularization aims at favoring the similarity of discriminant functions for similar classes (that is, sharing the same parent or, to a lesser extent, sharing a more distant common ascendant). We use a reparameterization of this softmax layer to introduce this regularization before reverting to the original, more compact, parametrization for the purposes of the actual implementation.

The form of the standard softmax layer, where the last internal representation is used to estimate the class probabilities, is recalled in Equation 1. Following [164], we consider a parameterization of the softmax layer that takes into account the class hierarchy. Each branch of the tree is provided with a $\phi_{jj'}$ parameter, leading to a reparameterization of the β_k coefficients in Equation 1 as the sum of all parameters along the path from the root to the corresponding leaf of the tree:

Equation 4

$$\beta_k = \sum_{j \in v_k \cup a(v_k)} \phi_{p(j)j}$$

where we recall that $a(v_k)$ is the set of ascendants of the leaf node corresponding to class k, and $c(j)$ is the parent of node j. The decomposition is illustrated in Figure 26. This decomposition is convenient because it allows to express the effect in terms of classes and superclasses. Still, it introduces as many new parameters as there are internal nodes in the class tree, which leads to indeterminacy. Neal and Shahbaba [164] use a Bayesian approach in which this indeterminacy is resolved by defining priors on the ϕ parameters. Here, we propose to use soft constraints, in the form of regularizers whose shape reflects the distance between the subsets of classes, as defined by the class tree. In other words, we want the siblings of any node to be similar, a condition that we formalize as a ℓ_2 penalty on the parameters of the branches leading to these siblings, in the manner of a local weight decay:

Equation 5

$$\forall j > K, \ \ \Omega_j = \sum_{j' \in c(v_j)} \|d_{j'} \phi_{jj'}\|^2$$

where we recall that $c(v_j)$ is the set of children of node v_j, d_j is the number of descendant leaves of v_j, and that the indices $j > K$ correspond to all non-terminal nodes of the tree. The proposal to weight each sibling by its number of descendant leaves is somewhat arbitrary, but it reflects a willingness to treat all (fine-grained) classes equally.

The overall training criterion then combines the data-fitting term Equation 3 and the sum of the regularization terms defined in Equation 5:

Equation 6

$$\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}) + \sum_{j>K} \lambda_j \Omega_j$$

where λ_j is the regularization parameter applied to node j. In all our experiments, we used the same value for all nodes to avoid a costly over-optimization of the hyperparameters.

From a computational viewpoint, we don't have to deal with the ϕ parameters, as the weight decay terms in Equation 5 can be rewritten in the β softmax parameters:

Equation 7

$$\Omega_j = \sum_{j' \in c(v_j)} \left\| d_{j'} \left(\frac{1}{d_{j'}} \sum_{k' \in l(v_{j'})} \boldsymbol{\beta}_{k'} - \frac{1}{d_j} \sum_{k \in l(v_j)} \boldsymbol{\beta}_k \right) \right\|^2$$

Note that this expression reflects the local view of the classes by each node v_j of the class hierarchy: in this form, Ω_j explicitly penalizes the deviations of the coefficients between the children superclasses and their parent superclass.

Hierarchical Misclassification Cost

A class hierarchy may call into question the validity of the misclassification cost as a performance criterion, since the classes are no longer considered equidistant (see for example [172,173]). Bertinetto et al. [174] propose using hierarchies to avoid gross errors and they measure the severity of errors by the distance to the deepest common ancestor. They average these numbers to summarize the overall performance of a model. Here, we propose to use a slightly different measure that also relies on the most standard metric between vertices of graphs/trees, that is, the length of the shortest path between two vertices. Note that there is no universal weighting scheme that would meet all needs. Here, we use a scheme consists in weighting an edge according to its depth d:

Equation 8

$$\begin{cases} 2^{-d} & \text{if the vertex at depth } d \text{ is a leaf} \\ 2^{-(d+1)} & \text{otherwise} \end{cases}$$

This weighting has the property that the distance between two vertices is equal to 2^{-d}, where d is the depth of their deepest common ancestor. In particular, this hierarchical misclassification cost (HMC) is 1 if the shortest path between two classes passes through the root of the tree, and it is lower otherwise. The HMC is therefore always lower than the standard misclassification cost (MC). For example, on the tree in Figure 26, if the actual leaf is C1.1 and the prediction is C1.2, the common ancestor is C1, which has a depth of 2, and so the HMC is $2^{-2} = 0.25$; if the prediction is C2, the common ancestor is C, which has a depth of 1, and so the HMC $2^{-1} = 0.5$; if the prediction is A, A1, A2, A3, or B, the common ancestor is the root and so the HMC $2^{0} = 1$.

Experimental Protocol

Architectures and Datasets

The proposed method has been evaluated with three different architectures with near state-of-the-art performance in vision tasks. The first two, InceptionV3 and ResNet50, have become the standard CNNs for transfer learning, and many works rely on fine-tuning these architectures on new problems [175–177]. In order to demonstrate the independence of the proposed approach from convolutional layers, we also tested vision transformers with ViT-B16 [142], so as to show the effectiveness of our approach on an architecture based on self-attention.

Regarding the datasets used, the primary experiments were conducted on portions of three traditional benchmarks, ImageNet-1K, CIFAR100, and FGVC-Aircraft, each presenting a hierarchy of balanced classes. For each benchmark, we vary the number of images per class, ranging from about 5 to about 100, in order to test our approach on a wider range of problems, while remaining in the low data regime where our approach can bring significant improvements. We also test our approach in a practical application using an imbalanced medical dataset with underrepresented classes. The aim is to show that a network equipped with hierarchical loss and regularization can learn to classify even classes with very few samples by using the knowledge from superclasses.

ImageNet-1K is the most used image classification dataset, consisting of 1,281,167 training images and 50,000 validation images. ImageNet-1K is divided into 1000 classes. The original images are downsampled to 64×64 to make classification more difficult and to mimic a domain adaptation problem with a deep class hierarchy.

CIFAR100 is divided into 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR100 are grouped into 20 superclasses, and a coarser hierarchy level was added to merge similar superclasses. The original image size is 32×32.

FGVC-Aircraft contains 10,200 images, with 100 images for each of 102 different aircraft model variants. The hierarchy of the dataset is also available from the official release. The original image size is 913×687.

Finally, the femur fracture dataset utilized in the previous work was used. As a reminder, it shows femur fractures taken from X-Rays and is composed of 4207 manually annotated images divided into different fracture types: 2003 *Unbroken* femur, 631 type *A1*, 329 type *A2*, 174 type *A3*, 625 type *B1*, 339 type *B2*, and 106 type *B3* fractures.

All hierarchies are of depth 3, except ImageNet-1K, which has a depth of 14, they are displayed in the supplementary materials. All original images were resized by bilinear interpolation to match the input sizes of the considered architectures, which are 224×224 for ViT-B16 and ResNet50 and 299×299 for InceptionV3.

Training Protocol

Each network was trained for 30 epochs with Adam optimizer. After a preliminary study, the learning rate and weight decay value were set to 0.001 and 0.1, respectively, for all datasets. Hierarchical regularization was used with the hierarchical loss, and the standard weight decay was used with the cross-entropy baselines. All the networks were pre-trained on ImageNet and thoroughly fine-tuned, that is, without freezing any layer. It seems inappropriate to fine-tune a model, previously pre-trained on ImageNet, on ImageNet, but the protocol used is that of domain adaptation. Indeed, the models are pre-trained on the original ImageNet images of size 224×224 or 299×299, while the fine-tuning is performed on degraded images, originally of size 64×64 and then resized to 224×224 or 299×299 by bilinear interpolation. This results in a change in the distribution of images that makes the protocol emulate domain adaptation.

Results and Analysis

Comparison on Benchmarks

Tested with three networks on three benchmarks, our approach results in an almost systematic improvement compared to the baseline where networks are trained with crossentropy. Figure 28 displays standard MC and HMC as a function of the number of images per class. In general, our approach improves all networks on all benchmarks. The improvement does not only concern the HMC, but also the MC. Moreover, the improvements are noticeable for all sample sizes tested, except for the smallest ones. The figure summarizes 45 pairwise comparisons for each of the two criteria, among which only four are not favorable to our approach in terms of MC and only one in terms of HMC. This demonstrates the ability of this technique to reduce the number of errors or at least gross errors (when only HMC is reduced) in low-data regime. On ImageNet, the increase of the depth of the class tree has a very clear impact on the HMC value, but it is also on this dataset that the relative improvements of HMC brought by our approach are the most important (division by 2, while MC decreases by about 20%), which implies that the coarse errors are much better controlled here.

Preliminary experiments (not shown) on networks with frozen layers did not show significant improvements in MC or HMC, suggesting that the effect on internal representation induced by training with the hierarchical loss is a key ingredient for its success. This effect is illustrated on the three models obtained after training on CIFAR100 with 30 samples per class and analyze their fine-tuned representations after training. We use the last inner representations, before softmax (that is, with the $\mathbf{z}_i$ of Equation 1) to perform Linear Discriminant Analysis (LDA). For readability, and to improve the fidelity of this LDA representation, we focus on two superclasses of CIFAR100 and their five (fine-grained) child classes. LDA being performed on these 10 classes, its 2-D summary is more faithful to the discriminant directions than the one that would have been performed on the 100 classes.

The principal planes of these LDA are displayed in Figure 29. The first observation is that superclasses are always more or less separated on these plots, except for the representation for ViT-B16 trained with cross-entropy, which is quite messy. The situation slightly improves with the hierarchical loss on this matter.

The standard fine-tuning applied to InceptionV3 and ResNet50 yields well-separated representations at the superclass level, but there is no visible structure within the superclasses, except that the fine-grained class “keyboard” is far from the others, so that there are three visible clusters corresponding to the superclass “fish”, the class “keyboard” and the superclass “household devices” deprived from

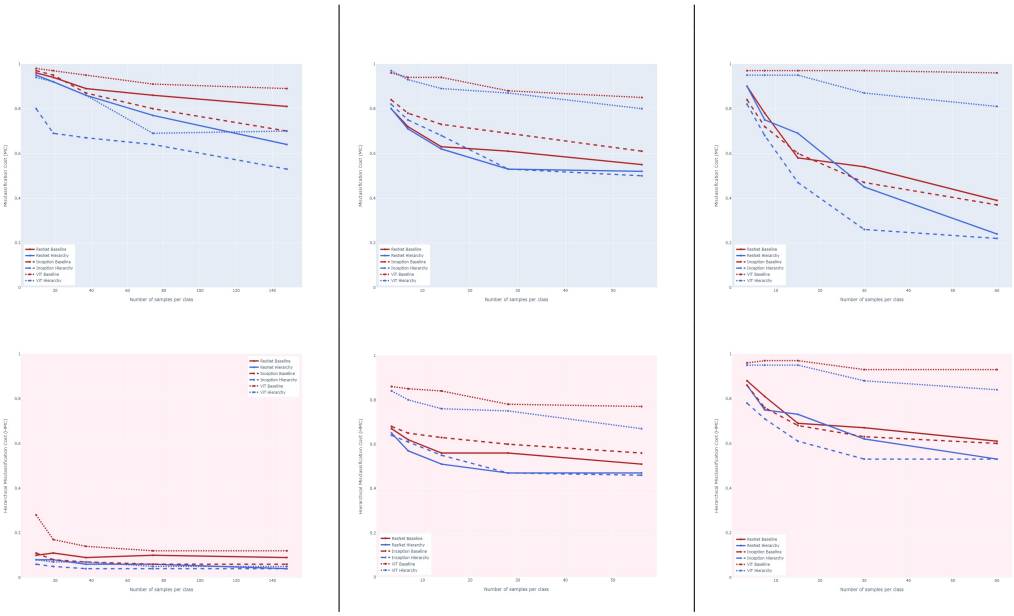


Figure 28 Misclassification cost (top) and hierarchical misclassification cost (bottom), in %, for the three benchmarks and the three network architectures; the x-axis represents the number of images per class used for training. Each line style corresponds to a network architecture (dashed: InceptionV3; solid: ResNet50; dotted: ViT-B16). The baseline algorithm, trained with standard cross-entropy loss, is in red; the algorithm trained with hierarchical loss is in blue.

“keyboard”. The representations obtained from the hierarchical loss are more balanced, with more visible clusters corresponding to classes identified within superclasses. In conclusion, LDA indicates that the minor benefit in hierarchical misclassification cost for ViT-B16 observed in Figure 28 can be explained by better superclass separation, without better class separation, and thus little improvement in classification cost. In contrast, the major benefit in misclassification cost for InceptionV3 and ResNet50 results from better class separation within superclasses, which also explains most of the improvement in hierarchical misclassification cost.

Comparison on Real Medical Data

As the dataset of femur fractures suffers from high imbalance, we measure performance with (macro-averaged) F1 scores. Table 14 compares the results with standard finetuning and with our approach that leads to improvements in F1-scores

of 4%, 13%, and 2% for InceptionV3, ResNet50, and ViT-B16, respectively. Also, the lower disparity in per-class F1-scores shows that our approach especially improves for the least well predicted classes. Through the hierarchy, well-represented classes can help recognize underrepresented classes by soft-sharing the characteristics of the superclass (see the form of the regularizer in Equation 7).

Table 14 Macro-averaged F1-score (with standard deviation) with standard fine-tuning using cross-entropy and weight decay (Flat FT) and with fine-tuning using the hierarchical loss and regularization (Hierarchical FT), on the femur fracture dataset.

Architecture	Flat FT	Hierarchical FT
InceptionV3	0.71 (0.19)	0.75 (0.14)
ResNet50	0.61 (0.20)	0.74 (0.15)
ViT-B16	0.71 (0.20)	0.73 (0.13)

Conclusion

Data collection is not always easy; a standard medical dataset is usually small, with potentially imbalanced classes corresponding to more or less rare conditions. In this situation, a priori knowledge can have a significant impact on machine learning. Many techniques to exploit the knowledge of a hierarchy on classes have been introduced [161,163,165,168], but there is a need for generic approaches that can be used on any standard multiclass classifier.

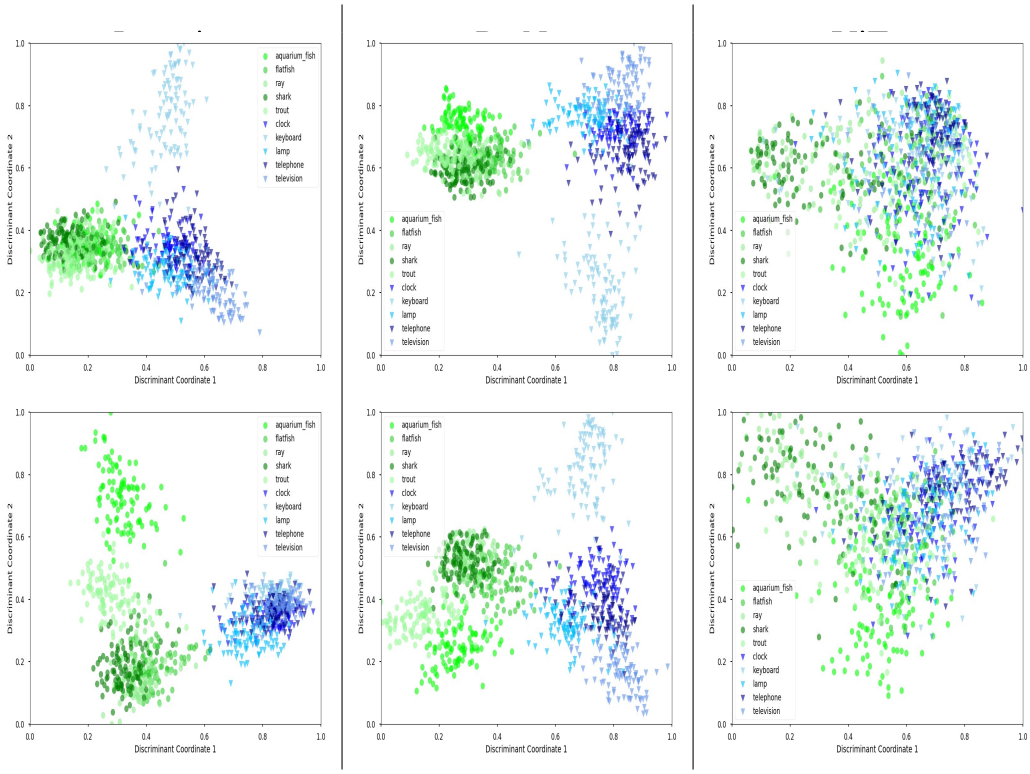


Figure 29 Principal plane of the Linear Discriminant Analysis performed on the fine-tuned representations learned by cross-entropy (top) and the hierarchical loss (bottom) for InceptionV3, ResNet50 and ViT-B16. For readability and fidelity of the 2D representations, the analyses focus on the 10 classes of two superclasses of CIFAR100. Each symbol and color corresponds to a superclass (green circle for "fish" and blue triangle for "household devices"), and each hue corresponds to a class.

Here we propose a very simple and generic approach for integrating the knowledge of a hierarchy over classes in the learning criterion, whose data-fitting term and regularization term are adapted to the class hierarchy. The approach is applicable to any model that is usually trained with crossentropy. As a by-product of our experiments, we introduce a hierarchical misclassification cost (HMC), in order to evaluate the performance on a hierarchical class structure, in which not all errors are equivalent. This generic metric is based on the distance between the predicted class and the actual class in the tree that represents the class hierarchy. As a caveat, note that our approach is based on the assumption that superclasses are homogeneous, comprising visually similar items. A dolphin and a lion are both mammals, but they are visually very different and live in different environments. It is unlikely that our method would benefit from a class hierarchy that is real but not consistent with visual similarities.

We evaluated our approach on three benchmarks with three SOTA architectures, on training sets of 5 different sizes. We also tested it on a real dataset with characteristics representative of those that are typically found in medical applications.

In the future, we would like to extend this approach to partial labels that may either represent a partial knowledge of the data-labeling expert or an intrinsic uncertainty in the data. For example, if we have an image of an unknown dog breed, we can still use the knowledge of the superclass “dog” for training. On the same line, but this time regarding predictions, it can be preferable to provide a cautious partial prediction rather than a bold precise one. However, deep learning is known to be subject to calibration problems: class posterior probabilities are often rather poorly estimated. Delivering cautious predictions may not be as easy as it first seems.

2.2 Emotion Classification

Introduction

Facial Expression Recognition (FER) has emerged in recent years as a challenging and interesting problem in computer vision with applications in several different areas, including the human-computer interaction (HCI) domain. While most of the algorithms in the literature are based on 2D images or video, the fast development of 3D imaging and scanning technologies is contributing to the increasing use of 3D face scans. As reported in previous works [178–180], the performance of 2D face recognition is negatively affected by the presence of illumination, pose variations and make-up changes.

On the other hand, 3D data naturally overcomes these demanding problems for being invariant to lightning conditions and geometric transformations. Furthermore, 3D models preserve the anatomical structure of the face and retain all the geometric and topological information, handling drawbacks of 2D face recognition and allowing an enhanced level understanding of the facial surface.

Despite the many advantages over 2D methods, 3D facial expression recognition is still not well studied due mainly to the limited availability of public databases. Thanks to the advancement in 3D data acquisition techniques and the availability of inexpensive sensors, the number of public 3D databases is constantly increasing. Nowadays, it is easily possible to create a large-scale 3D database as several cost

effective RGB-D cameras capture both RGB images and depth data. These sensors, also integrable into some mobile phones, can acquire 3D objects and produce point clouds (i.e. sets of 3D data points defined in 3D space, where each point is represented by using its position in terms of Cartesian coordinates) preserving all the geometric information.

Among the various deep learning architectures, during the last decade Convolutional Neural Networks (CNNs) have obtained interesting results in nearly everywhere applications. More recently, parallel to the improvement of convolutional networks, other architectures based on self-attention, in particular Transformers [133] have become the reference point for a wide range of tasks. The adoption of transformers in vision models is a recent and relevant trend, born with the introduction of a new paradigm to represent and process high-level concepts in images using visual tokens, the already discussed Vision Transformer. Its performance is still poor when trained on smaller datasets because Transformers need large amounts of data to well generalize on computer vision tasks. Nevertheless, when ViTs are pre-trained on a huge dataset and then extended to a small-sized one, they still achieve very competitive results.

This work presents a novel multimodal architecture based on two different branches: a 3D CNN inspired to the classic VGG structure and a 2D ViT encoder. These networks are trained in parallel and their top layers are concatenated for classification to address two current research questions in a multimodal approach. The major objectives of the present work are: to exploit the advantage of using three-dimensional data, like point clouds, together with two-dimensional data to improve the recognition rate; to test a cutting-edge architecture on RGB images to perform FER. Emotion databases are usually poor in terms of samples, and an architecture based solely on ViT could not be enough; furthermore, fine-tuning ViT (which was trained on 2D images) on a 3D dataset is not effective. In addition, CNNs lack the self-attention mechanism, which is useful to underline the relation between different facial features (e.g., eyes, mouth, and expression wrinkles). For this reason, the hypothesis of this work is that a multimodal approach based on ViT and 3D CNN could fulfill the deficiencies of both the architectures.

Datasets

The BU-3DFE (Binghamton University 3D Facial Expression) Database and the Bosphorus 3D Face Database [181,182] are the benchmarks for static 2D and 3D FER.

The BU-3DFE database consists of 100 subjects (56 females, and 44 males), aged between 18 and 70 years, of different populations. For each subject there are 25 samples, one for the neutral expression, and the other 24 samples for the six prototypical expressions, performed at four levels of intensity. As a result, the BU-3DFE is composed of 2500 2D texture images and 2500 3D models. The standard dataset used for 3D FER is a subset of the whole BU-3DFE, containing 1200 2D and 3D face pairs of 100 subjects with the 6 prototypical expressions at the two highest levels of intensity.

The Bosphorus 3D Face Database contains 105 subjects (45 females, and 60 males), most of them Caucasian. Also, 27 professional actors/actresses are incorporated in the database to obtain more emphasized expression data. Every subject has from 31 to 54 scans, performing not only the six basic emotions (anger, disgust, fear, happiness, sadness, and surprise), but also different action units, poses (yaw, pitch, and cross rotations), and occlusions (hand, hair, and eyeglasses) for a total amount of 4652 pairs of 2D face images and 3D face models. Within the dataset, a subset consisting of 65 subjects who have performed all the six prototypical expressions in frontal view, without occlusions, is mostly considered for FER, but the scans of only 49 subjects have been correctly processed for this study.

The CALD3R database [183] stores 4678 pairs of 2D RGB images and 3D point clouds and it has been realized by out research group by recording spontaneous facial expressions elicited by static visual stimuli.

Methods

A multimodal 2D+3D approach has been adopted in order to exploit all available data, combining the advantages of both 2D and 3D face data. Indeed, there are several limitations using only RGB information because the 3D world is projected into a 2D space leading to inevitable data loss.

In this work, the early level fusion adding a fusion classification layer, obtained by concatenating two fully-connected layers, has been proposed. A 3D convolutional neural network inspired by VGG architecture has been leveraged as 3D Depth-Net while a Vision Transformer has been used as 2D RGB-Net. Figure 30 shows a scheme of the overall architecture.

A 3D convolutional neural network has been chosen as 3D Depth-Net to perform facial expression recognition based on 3D coloured point clouds. The input of the network is fixed to a 112x112x56x3 size, and it is passed through a stack of four Conv3D layers.

The ViT has been applied to 2D RGB images. This architecture focuses on patches of the image, flattened using a linear projection matrix. Among the configurations proposed in the original paper, the so-called ViT-B/32 has been selected.

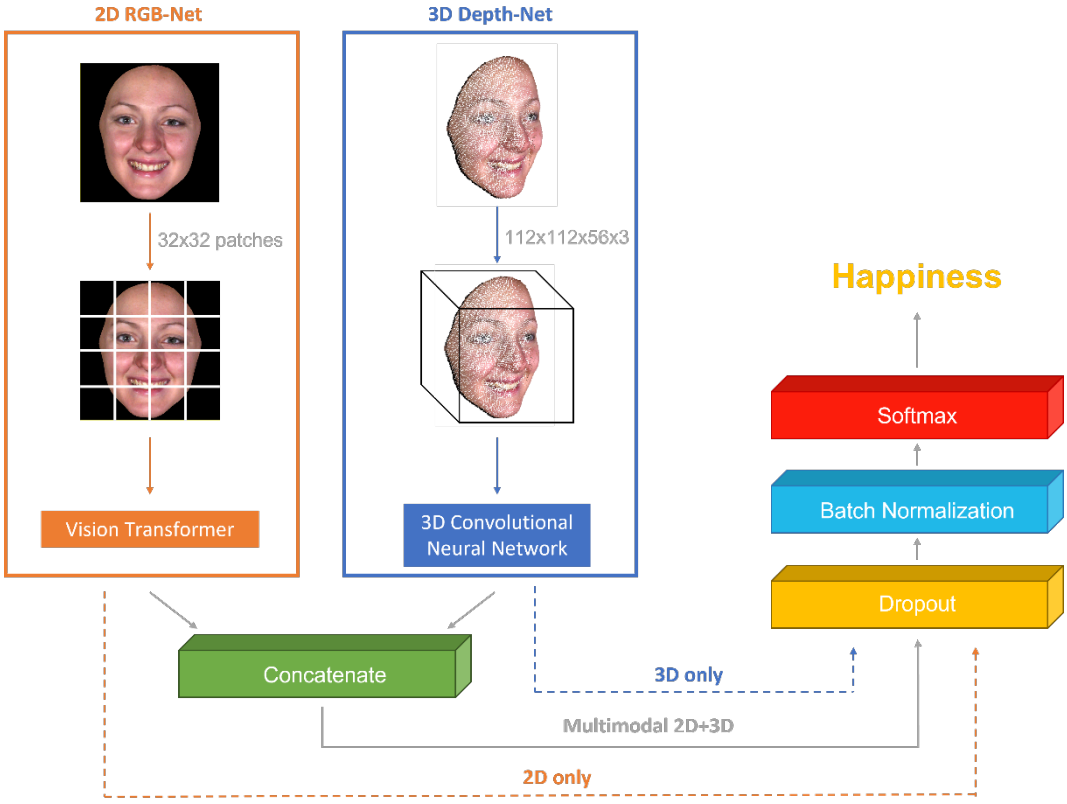
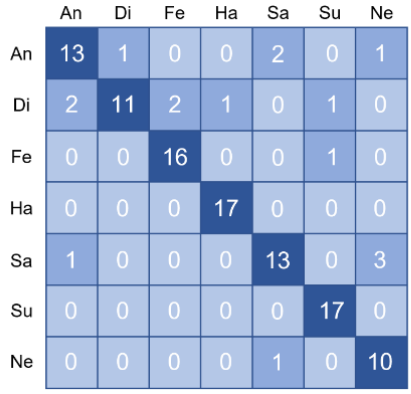


Figure 30 Scheme of the overall architecture, composed of a 2D RGB-Net and a 3D Depth-Net. The two networks can work independently (2D only and 3D only), or be concatenated by an additional layer and work together (Multimodal 2D+3D)

Results

In order to demonstrate the real advantage of a multimodal approach compared to using only two-dimensional or three-dimensional data, a comparison is proposed. Several configurations have been tested to determine the most performing architecture, which turned out to be ViT-B/32 concatenated with VGG-3D, with an accuracy of 0.858 considering the test subset (113 pairs of 2D face images and 3D face models for the 6 basic emotions and the neutral expression of 11 subjects). Before selecting these architectures, different tests were carried out. First, it was assessed whether results could improve using a "from scratch" instead of a pre-trained ViT. Secondly, in order to justify the use of ViT although it is well-known

that usually outperforms CNNs if trained with a large number of images, it was replaced with a classic ResNet50 obtaining lower accuracies. The 3D branch was then tested and the VGG-3D was replaced with a more complex ResNet3D architecture achieving an accuracy of 0.655 combined with ViT-B/32. Finally, the best performance was compared with monomodal branches. VGG-3D alone obtained an accuracy of 0.734 while the ViT-B/32 configuration had a very low accuracy of 0.354. The confusion matrix, which reports the performance of the classification model for each emotion, is shown in Figure 31 for the best 2D + 3D configuration only.



	Precision	Recall	f1-score	Support
Anger	0.81	0.76	0.79	17
Disgust	0.92	0.65	0.76	17
Fear	0.89	0.94	0.91	17
Happiness	0.94	1.00	0.97	17
Sadness	0.81	0.76	0.79	17
Surprise	0.81	1.00	0.94	17
Neutral	0.71	0.91	0.80	11

Figure 31 Confusion matrix and corresponding precision, recall, and f1-score values for each class, where precision is the ratio of correctly predicted positive observations to the total predicted positive observations, recall is the ratio of correctly predicted positive observations to the all observations in the actual class, and f1-score is the weighted average of precision and recall.

Conclusion

The aim of this study was to demonstrate the advantage of using a cutting-edge 2D+3D multimodal approach, capable of improving the recognition rate compared to the use of a single type of data. The implemented architecture is the concatenation of the top layer of a 3D CNN with ViT, a very recent deep learning model that interprets an image as a sequence of patches. Regarding the 2D branch, the choice fell on ViT as it has achieved cutting-edge results in almost all areas of image recognition, overcoming convolutional neural networks. On the other hand, regarding the deep network for 3D data classification, a convolutional network inspired by VGG-16 for its simplicity was built. The proposed architecture has been trained and tested on both posed (BU-3DFE and Bosphorus) and spontaneous (CALD3R) facial expression databases, using RGB images as 2D data and point clouds as 3D data.

The results show and confirm the importance of understanding the available data, and exploiting all the information contained therein. Indeed, expression lines, particularly accentuated for example in anger, curvatures of the face, or concavity/convexity changes contain more information than the chromatic variation of a color image, making it possible for 3D data to naturally overcome some demanding problems belonging to 2D data such as lighting conditions or geometric transformations. A multimodal approach is able to combine the advantages of both technologies, and it has been proposed in a novelty way. For the first time, to the best of our knowledge, a ViT architecture has been combined with another deep learning network to perform FER and overcome the problem of the low number of images. Moreover, 3D data was given in input to the CNN as point clouds with associated texture, representing a further element of novelty. This approach made it possible to obtain a recognition accuracy of the six basic emotions equal to 0.875, despite having chosen a very simple architecture.

The next step will be to focus only on spontaneous and ecologically valid data, exploiting both 3D and 4D data introducing the variable time. The first goal is therefore to be able to increasingly arouse true emotions to use a facial expression recognition system not only in a laboratory context but mostly in the real world.

Chapter 3

Second Layer and Full Framework

3.1 Assisted Surgery: overview

Formulation of the problem

In recent years, the adoption of minimally invasive surgery (MIS) technology has grown exponentially, to reduce access to wound trauma and decrease the incidence of postoperative complications [184]. In particular, the high demand for greater surgical precision has led to the birth of robotic surgery. The introduction of surgical robots has given many advantages to surgeons, for example, improved stereoscopic visualization, removal of hand tremors, and greater precision and enhanced maneuvering of the surgical tools. In the context of urology, robotic surgery was introduced 15 years ago and is now used worldwide [185] and, in particular, in the context of radical prostatectomy. Despite the above-mentioned positive improvements, it still presents challenges that require to be solved. Among them, we addressed the problem of the limited field of view offered by the endoscope used for navigation. AR could be the solution to overcome this obstacle, as it combines images from the real world with others digitally produced, with the intent of increasing the information the viewer can obtain from their combination [186]. In particular, when AR is applied to MIS the goal is fusing 3D objects produced from pre-operatory patient data with real-time images taken by the endoscope camera [187]. The challenge then became how to correctly align the virtual objects with their real-world equivalent. In a previous work from our research group [188], we presented our progress in enhancing endoscope video during RARP, by superimposing the 3D virtual model of the patient's prostate on its 2D counterpart, using different real-time techniques. The proposed framework was divided into 5 main phases that characterize a prostatectomy procedure [189]. To perform the task, a virtual model was generated from high-resolution preoperative medical imaging techniques, such as MRI. This 3D reconstruction accurately reproduces the organ and the surrounding structures of the patient

undergoing the operation and was modeled by bioengineers using the HA3D™ technique. According to [189], the steps of this particular procedure are highly standardized, hence we have grouped them into 5 subsequent steps, based on similar visual characteristics and similar levels of benefit from the use of RARP. These 5 steps are: 1) Defatting and incision of the endopelvic fascia. 2) Management of the bladder neck. 3) Vase clamping and nerve-sparing. 4) Surgery by the prostatic apex. 5) Targeted biopsy.

In the 2nd step, the 3D overlay was rarely requested by the surgeon. In the 4th step, 3D reconstruction from MRI is considered not accurate enough in depicting the apex. Hence, we excluded these stages from those requiring AR implementation. Instead, during the 1st step, it is critical for the surgeon to correctly identify and locate the neck of the prostate and, in the 3rd step, preserve the nerves' functionality after the procedure. In these stages, the presence of the 3D model correctly representing organ boundaries was requested by the surgeons to increase their intra-operative perception. Aligning the 3D model of the prostate with its physical counterpart, without clear visual references, proved to be extremely inaccurate. For this reason, we focused on the 5th phase. During this stage, the insertion of a catheter into the pelvic cavity provides an artificial feature easy to identify and used to guide the virtual-over-real overlay. Superimposing a 3D model along with the cancer position is fundamental to improving the localization of the tissue sample for post-operative biopsies. Our previous approach to detecting the catheter leveraged solely classic CV techniques, which made it very fast and suitable to operate in real-time; nevertheless, this method experienced great variability depending on conditions, such as illumination changes or camera movements, resulting in non-optimal performances. We now changed our approach, evaluating the ability of different neural network architectures to perform semantic segmentation and localize the catheter in the video stream, maintaining an acceptable elaboration speed.

Introduction

In recent decades, technology has helped several medical procedures to massively improve [8]; in particular, great progress has been achieved with Deep Learning (DL) [3] paradigms. Recently, DL has acquired a fundamental role in the medical environment, and many different remarkable applications have been implemented, from orthopedics [5,110] to oncology [32] and many others [190]. The same enhancement was brought in by Augmented Reality (AR), in particular in surgery. Neurosurgery was the first medical specialty in which AR was introduced to assist the surgeon [191], followed by orthopedics [192], dental surgery [193], and many

more. Surgery, in fact, is one of the main subfields in which DL and AR are making a fundamental contribution. The potential of DL in intraoperative surgery is formidable: by learning from data on past procedures, it would be possible to monitor the progress of a surgical procedure [194,195], estimate the pose of the medical staff [196,197], evaluate skill assessment and provide new ways of interacting with multimodal data during and after a procedure [198], just to name a few examples. All these procedures necessitate an intuitive interface for surgeons and specialists, that could be given by AR applications. Lately, the efficacy of AR as a tool to improve the transfer of information has been the topic of many different research works. In particular, give the surgeon the ability to see what cannot be normally perceived, as the hidden organs inside the body of a patient is the main addition that AR brings to surgery [199]. Unfortunately, in intraoperative surgery, joint utilization of DL and AR is still underrated due to the difficult compromise between working with 3D data and the efficiency needed for real-time elaboration.

In this work, we developed three subsequent approaches to solve this task. At first, we focused on the catheter, and we performed a segmentation of the intraoperative surgery in order to obtain the information necessary to perform the overlay. In the second step, we added a second CNN to predict the rotation and we extended the work to organs, namely the prostate and the kidney. Finally, we started to experiment with event prediction, inserting a multi-modal network able to produce both the segmentation map and a warning in case of high blood accumulation. Part of the first two works was already published in [200,201], while the third is still under-review.

3.1.1 First Approach: Segmentation CNN

Introduction

In Section 1.1.2, we outlined a definition of an Intelligent Operating Room (IOR), a collaborative operating room based on highly intuitive, natural, and multimodal interaction. In this study, we moved the first step in this context. The preliminary method used a complex Computer Vision approach to localize different areas in the endoscopic video. Here, an encoder-decoder structure based on Convolutional Neural Network (CNN) is applied to obtain real-time semantic segmentation of the scene and improve in precision the subsequent 3D enhancement. We compared different combinations of segmentation architecture and base neural network to select the most performing one, based on training and testing with 4 different videos (*A, B, C,* and *D*) presenting the last phase of prostatectomy procedures. Finally, we

used video *E* to validate the performances of the 3D overlay method with a new dataset. The developed application is currently being used during in vivo surgery, for extensive testing, by the Urology unity of the San Luigi Hospital, in Orbassano (To) Italy, and the augmented video stream can be accessed directly into the Tile-Pro visualization system of the Da Vinci surgical console. The main contribution of this work is to present a solid evaluation of our methodology to visually overlay a 3D model over a 2D endoscopic stream. The focus of this research is to provide a robust clinical application solution to the stated problem, based on a numerical evaluation. Rather than techniques based solely on Computer Vision (CV) to determine the 3D organ model position and rotation, as in our previous approach [188], we leveraged the output of a state-of-the-art CNN segmentation architecture, obtaining better results.

Methods

Semantic Segmentation

The semantic segmentation of an image aims to associate each pixel with a class of a predefined set. Segmentation techniques are mostly based on CNN. The typical architecture adopted in these procedures uses two opposite branches of CNN called the encoder-decoder structure. The encoder has the typical structure of a CNN with convolutional and pooling layers that sample the input to generate a high dimensional characteristics vector. The decoder has the opposite structure: it takes as input a high dimensional characteristics vector and generates a semantic segmentation mask. We explored the best combination of architecture and neural networks and selected the most performing one. We tested 3 segmentation architectures among the vast amount of versions: SegNet [202], U-Net [17] and PSPNet [203]. They were chosen for a particular reason related to our objective, as SegNet is specifically designed for real-time tasks, U-Net was created especially for biomedical imaging, and for this reason, it works very well with a limited number of images and PSPNet is optimized to learn better global context representation of a scene. The next task was to choose an appropriate base network. The models we decided to test are ResNet [28], VGG [204], and MobileNet [205]. We trained and tested each possible combination of these segmentation architectures and the base networks, except for the combination of PSP-Net and MobileNet, which was not present in the library used.

Dataset

We collected 5 videos from different surgical procedures showing the insertion of the catheter, i.e., the above-mentioned fifth phase “targeted biopsy”, and we extracted, tagged, and resized a set of frames. The first video selected, named *A*, due to its variety of operations and camera views, and the fact that the surgical equipment appears and disappears in a continuous fashion, was considered the most exemplary. Adding up the total number of frames of all the videos of different duration, we ended up with approximately 15.570 images. From these, we extracted from 275 images to be used for training, plus another 50 images taken from the same video to test the segmentation. Then, 90 images from *B, C,* and *D* were selected to generalize the results: 50 were used to test the former model, 40 to re-train the model, including frames from different videos, after the selection of the best combination of segmentation architecture and encoder network. We decided to use a small percentage of the total number of images to demonstrate that this approach could rely on a limited number of training samples. Each image was manually tagged by two senior urologists using *labelme* that provides an interactive GUI to produce the ground-truth segmentation for 3 categories: *background, tool,* and *catheter*. Nevertheless, we gave importance to optimizing the value of the *catheter*, as we needed its location to map the 3D model to the 2D video. Finally, we extracted 100 frames from video *E* to test the overlay performances of the AR framework. Each frame has been selected for its peculiarity (e.g., extreme rotations, partially hidden catheter, uncommon catheter rotational values, etc…) and tagged by our specialists’ team with the optimal value of the anchor point (*p1, p2*) and the rotation along the *X* and *Z* axis for the 3D model. Figure 32 shows some samples for the 5 different videos. All the samples have been resized $416 \, x \, 608$ pixels.

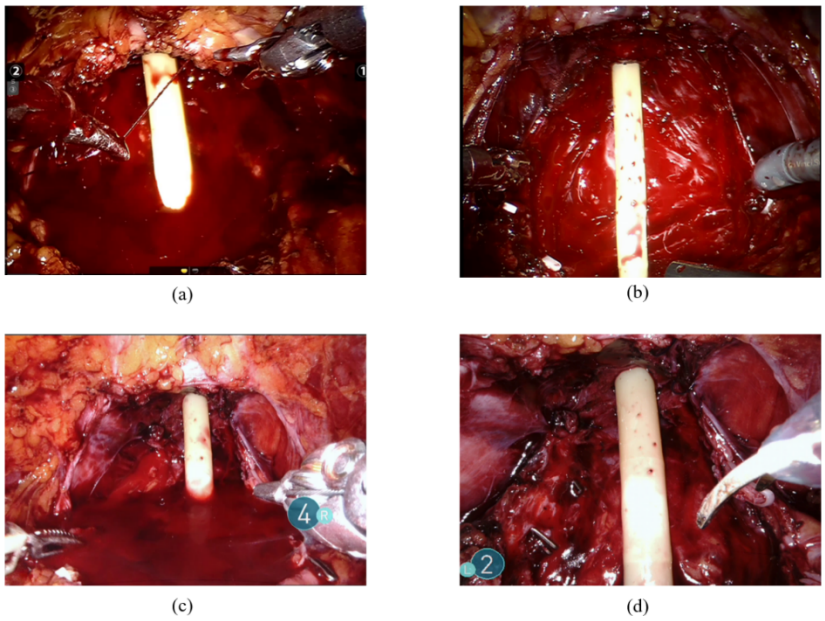


Figure 32 Samples taken from video A, B, C, D and E videos

Metrics

Different combinations of network and basic architecture were tested, and the best pair was selected using Intersection over Unit (IoU) metrics and the number of iterations per second (*it/s*). The first metric was used to measure the precision of the segmentation, and the second metric to calculate the number of frames that could be processed every second. The IoU is defined as:

$$IoU = \frac{AOverlap}{AUnion}$$

where *AOverlap* is the area of overlap between the expected segmentation and ground truth, and *AUnion* is the area of union between the expected segmentation and ground truth. This metric is normalized in the interval [0, 1], with 0 meaning that there is no overlap and 1 meaning a perfectly superimposed segmentation. Each pixel of the network's output is compared with the corresponding pixel in the ground truth segmentation image. We not only computed the IoU for the *catheter,* but also the Mean IoU, which is the average IoU between the 3 classes. This metric was used to demonstrate that the network was learning to also segment the *tool* and *background* classes. The second metric, *it/s*, is defined as:

$$it/s = \frac{n}{sec}$$

where n is the number of iterations, and *sec* is the time unit. In this case, one iteration consists of predicting the segmentation output given an input image. As the frame rate metric depends on different aspects, such as the complexity of the mesh, the rendering engine used, the hardware specifics, the particular implementation of the pipeline, etc., these factors can determine a strong fluctuation of the metric, for this reason, we preferred to opt for a metric independent of these parameters. We empirically noticed that our frame rate was acceptable if we kept the *it/s* greater than 10; this evaluation is not indicative but sufficient for us to obtain a real-time validation. Regarding the validation of the 3D overlay, we computed two different metrics. The 3D model is attached to the video stream, given the coordinates $(p1, p2)$ of the anchor point and the rotation along the X and Z axis. The difference between the predicted $(\widehat{p1}, \widehat{p2})$ and the actual anchor point $(p1, p2)$ was evaluated with the classical Euclidean distance between two 2D points:

$$eucDist = \sqrt{\left(p1 - \widehat{p1}\right)^2 + \left(p2 - \widehat{p2}\right)^2}.$$

The difference between the rotations was computed by converting the values of the rotations (considering the one along the Y-axis equal to 0, since rarely involved) in quaternions and then calculating the geodesic distance between two quaternions' coordinates $q1$ and $q2$. To get a distance between two-unit quaternions, you have to rotate both such that one of them becomes the identity element. To do this for our pair $q1$ and $q2$, we simply multiplied $q1$ by $q2$'s inverse from the left

$$Q = (inverse(q2) * q1)$$

and normalize the obtained quaternion Q through L2 normalization:

$$geoDist = L2(Q) = \sqrt{Q \cdot Q}$$

The metric is a positive amount corresponding to the length of the geodesic arc connecting $q1$ to $q2$.

Post-Processing

From the segmentation output, we used the algorithm proposed by Suzuki *et al.* [207] to obtain the segmentation contours and the Sklansky's algorithm [208] to obtain the convex hull given the main contour, to extract the information needed to correctly align the 3D model to the real-time video. This information is:

1. the rotation angles for X and Z axis. Y-axis is not considered since rarely involved;

2. the anchor point where to plot the upper extremity of the catheter.

In particular, as shown in more detail in Figure 33, the central point of the upper edge of the boundary box, shown as a black dot, is considered the anchor point of the 3D mesh, corresponding to the apex of the catheter. The dimension of the upper edge is compared to the diameter of the catheter in the real world to determine the scale of the virtual model. The vector from the anchor point to the centroid, shown as a white dot in Figure 33, is used to calculate the rotation of the mesh along its *Z*-axis (standard Cartesian coordinate system). The rotation along the *X*-axis is processed by comparing the upper and lower part, divided by the centroid, of the area of the catheter's shape detected. These two sections are highlighted in Figure 33 with two different colors. When the lower area is larger than the upper area, the mesh must be rotated towards the camera. When the lower area is smaller than the upper area, which happens less often, the 3D model should be turned so that its base is further away from the camera. The full pipeline of this method is shown in Figure 34.

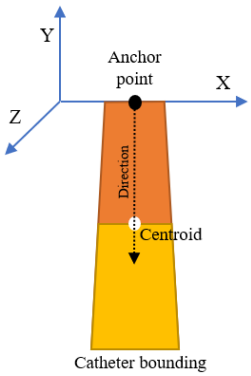


Figure 33 The bounding polygon of the catheter. The upper black dot marks the anchor point, while the white one marks the centroid. The direction arrow was used to calculate the Z rotation and the two sections of the area divided by the centroid to compute the X rotation

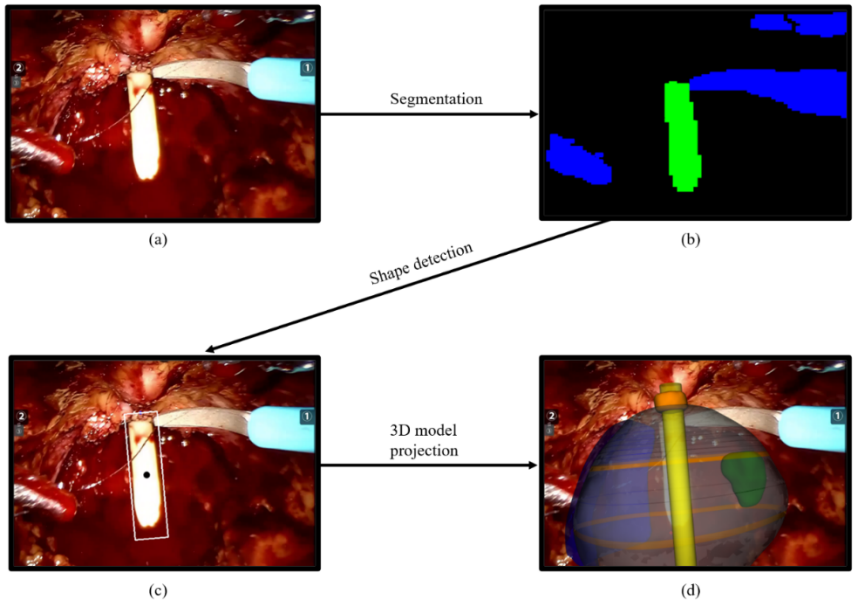


Figure 34 The full pipeline of our method. The original image (a) is segmented (b) and, after the shape detection (c), the 3D model is projected over the 2D image (d)

Training, Framework, and Evaluation

We first trained the model with 275 images of the *A* dataset, and tested it with 50 images taken from *A,* and subsequently with 50 images from *B, C,* and *D* each, to select the best combination of segmentation architecture and CNN for this specific case. Then, we trained the model selected with *A* video plus two among the *B, C,* and *D* videos and tested with the excluded one to obtain more generalized results. This was done with 40 previously unseen images for classes *B, C,* and *D*. Finally, we trained and tested the model with all the videos. We then took the model trained with *A, B, C,* and *D* and tested the overlay precision with 100 frames extracted from video *E*. After trying different configurations, we obtained the best results using a batch size of 4 and Adam [209] optimizer with a learning rate of 0.001 and beta values of 0.9 and 0.999, respectively. The function to calculate the loss was categorical cross-entropy. We run the model for just 20 epochs before it starts to converge. We used Keras [210], an open-source neural-network library written in Python, running on top of TensorFlow, and, in particular, the *keras-segmentation* [211] library, on Windows 10 Pro with NVIDIA Quadro P4000.

Results

We first run all the possible combinations between the 3 segmentation architectures (U-Net, SegNet, PSPNet) and the 3 base networks (MobileNet, VGG, ResNet), except the ensemble of PSPNet and MobileNet due to the missing implementation in the Python library. The IoU values obtained are shown in Table 15, for networks trained just with the *A* video and tested with different frames taken from *A, B, C,* and *D* videos. The values of Mean IoU are shown for completion to demonstrate that the network was learning also to segment the *tool* and *background* classes. In Table 15 it is also shown the iterations per second for each combination, because we are working with a real-time application, and we needed the application to have an acceptable frame rate. After picking U-Net with MobileNet as the most performing ensemble for performance and speed, we re-trained the chosen network with *A* video plus two videos among *B, C, D*, and tested it with the remaining one. These results are shown in Table 16, where we also indicate the results of a network trained with all 4 videos.

Table 15 Results training with A and testing with A, B, C, and D. Catheter IoU (Cat IoU) and Mean IoU for the available combination of architecture and base network, training with 235 frames taken from A video and testing with 50 frames taken from A, B, C, D. It is also shown the number of iterations per second for each combination. One iteration consists in predicting the segmentation output given an input image.

Architecture / Base Network		U-Net			SegNet			PspNet		
	Test Video	Cat IoU	Mean IoU	*It/s*	Cat IoU	Mean IoU	*It/s*	Cat IoU	Mean IoU	*It/s*
MobileNet	A	0.926	0.927	11.04	0.913	0.911	10.96	NA	NA	NA
	B	0.888	0.819		0.763	0.776		NA	NA	
	C	0.467	0.705		0.259	0.611		NA	NA	
	D	0.868	0.753		0.634	0.690		NA	NA	
VGG	A	0.815	0.720	7.81	0.914	0.887	8.02	0.863	0.870	8.68
	B	0.752	0.733		0.846	0.830		0.709	0.736	
	C	0.147	0.537		0.499	0.710		0.324	0.587	
	D	0.580	0.680		0.836	0.735		0.622	0.614	
ResNet	A	0.937	0.933	6.23	0.829	0.847	8.95	0.901	0.900	6.46
	B	0.925	0.811		0.711	0.714		0.792	0.725	
	C	0.626	0.756		0.254	0.511		0.369	0.628	
	D	0.892	0.760		0.646	0.577		0.787	0.678	

Table 16 Training and testing of U-Net architecture with MobileNet as the base network to generalize the results. We trained the network with A video and two videos among B, C, D, and tested it with the remaining one. In the first row, the results for the baseline are reported again to make the comparison valid and improvements are shown in parenthesis. In the last row are shown results given from training and testing the network with all the videos.

Train \ Test	A		B		C		D	
	Cat **IoU**	Mean **IoU**	Cat **IoU**	Mean **IoU**	Cat **IoU**	Mean **IoU**	Cat **IoU**	Mean **IoU**
A (Baseline)	*0.926*	*0.927*	*0.888*	*0.819*	*0.467*	*0.705*	*0.868*	*0.753*
A+B+C							0.899 (↑0.031)	0.864 (↑0.111)
A+B+D					0.661 (↑0.194)	0.783 (↑0.078)		
A+C+D			0.841 (↓0.047)	0.840 (↑0.021)				
A+B+C+D	0.925 (↓0.001)	0.927 (=)	0.936 (↑0.048)	0.921 (↑0.102)	0.762 (↑0.295)	0.858 (↑0.153)	0.945 (↑0.077)	0.937 (↑0.184)

Table 17 Comparison between the proposed approach based on Deep Learning (DL) and the former approach based on Computer Vision (CV) techniques for segmentation. Standard deviation σ is also shown in the rightmost column.

	A cat IoU	B cat IoU	C cat IoU	D cat IoU	Mean
DL approach	0.925	0.936	0.762	0.945	0.894 (σ = 0.076)
CV approach [188]	0.450	0.590	0.079	0.240	0.339 (σ = 0.195)

Then we compared our best network with the one based on Computer Vision techniques used in the previous work. The results of the comparison are shown in Table 17 while a graphical example is shown in Figure 35**.** We also evaluated the

performances of the 3D overlay with the values of Euclidean Distance between the real and the predicted anchor point and the Geodesic Distance between the *X* and *Z* real and predicted rotations.

Table 18 Comparison between the proposed approach based on Deep Learning (DL) and the former approach based on Computer Vision (CV) techniques for the 3D overlay performances with standard deviation σ

	Euclidean Distance	Geodesic Distance	# Images
DL approach	4.160 (σ = 1.448)	0.169 (σ = 0.073)	100
CV approach [188]	12.569 (σ = 4.456)	0.266 (σ = 0.131)	100

Discussion

In this work, we present the results of our effort to improve the precision of catheter identification in the endoscope frames during the fifth phase of RARP. During this phase, the 3D overlay, based on the correct catheter positioning, is crucial to locate tissue sampling for post-operatory biopsies. For this reason, we trained an encoder-decoder structure for segmentation. Our dataset was composed of frames extracted from 5 different videos: 325 frames from *A*, 90 from *B, C,* and *D* each, and 100 for *E*. We first tested different combinations of segmentation architecture and base networks. We performed the training with frames taken from *A* video and the testing with frames from *A, B, C,* and *D* videos, and the best catheter's IoUs were obtained with the U-Net architecture with ResNet and MobileNet as base networks. ResNet reached an average catheter IoU of 0.845 while MobileNet 0.780. Nevertheless, MobileNet performed 11.04 *it/s* and ResNet 6.23 *it/s*. As already discussed, we noticed that the frame rate of the real-time application was acceptable if we kept the *it/s* greater than 10: with 11.04 *it/s*, on the hardware used for testing, we are able to reach a frame rate of approximately 8*fps*, which is a good value for a medical application, where the constraint on real-time is not so severe. Since we work with real-time data, we decided to choose MobileNet as a good compromise between precision and speed. We then improved the IoU by training the network with *A* video plus two among *B, C,* and *D* and testing with the remaining one. When testing with unseen data, the IoU computed with this approach had a significant increase, especially for the *C* video, where the catheter IoU passed from 0.467 to 0.661. In the last line of Table 16**,** we can see the best results. They were obtained when training the network with all 4 videos. These results outperformed the former

approach, based on Computer Vision segmentation techniques, where the Gaussian blurred frames were converted to HSV format and binary thresholded given an adaptive pixel range, in order to identify the pixels belonging to the catheter. The average IoU obtained with this former method was 0.339 (σ = 0.195) versus an IoU of 0.894 (σ = 0.076) of the new proposed approach, as shown in Table 17. After extracting the segmentation mask, we computed the rotation angles and the pivot point to position the 3D model. In Figure 36**,** representative samples of the network's output, together with the subsequent 3D overlay, are shown. We then evaluated our technique using 100 frames extracted from video *E* and testing the precision of the predicted anchor point and rotation angles, obtaining a Euclidean Distance of 4.160 (σ = 1.448) and a Geodesic Distance of 0.169 (σ = 0.073), compared to the previous 12.569 (σ = 4.456) and 0.266 (σ = 0.131), as shown in Table 18. In Figure 35 a visual comparison of our approach with the previous approach is shown: given a wrong segmentation map, the approach based on CV 1) predicted an anchor point $(\widehat{p1}, \widehat{p2})$ with a wrong value of $\widehat{p2}$ compared to the actual point $(p1, p2)$ and 2) returned values of rotations *X* and *Z* slightly wrong, while in our approach, it is possible to notice that the value of *X* correctly rotates the catheter toward the camera, and the value of *Z* matches the left-right rotation of the catheter. However, there are still challenges to overcome. As shown in Figure 36**,** the segmentation network is still vulnerable to abrupt light changes (a) and blood occlusion (b). These issues consequently affect the 3D overlay performances: in Figure 36 it is possible to notice, also underlined with a dashed white circle, that in (a) the anchor point of the catheter is wrongly detected as the area related to a high presence of light; in (b) the values of rotations are conditioned by the lower part of the catheter, affected by blood occlusion. In the next approaches, we will address this problem by increasing the variability of data samples in the training set and acquiring more footage from RARP procedures showing different light and surgical conditions. Concerning the 3D overlay performances, the main problem is related to the anterior-posterior rotation along the *Z*-axis, which is often mispredicted. For this reason, in the next approach we present a second CNN to identify the correct rotation keeping an overall acceptable frame rate. Finally, this work is limited to situations where the catheter is present. The next section will show also how to deal with organs without relying on physical objects.

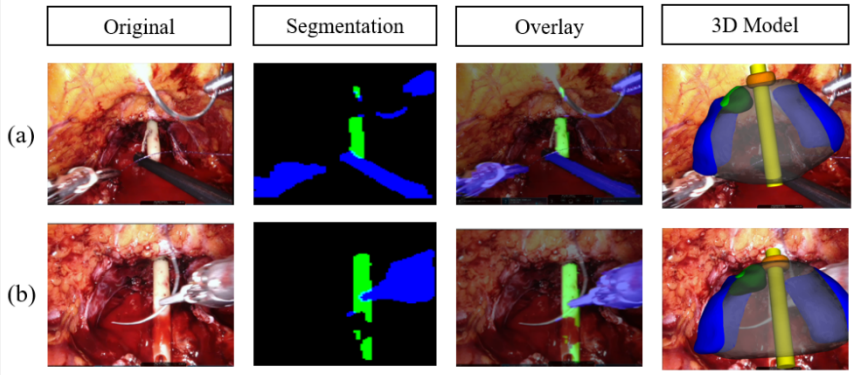


Figure 35 Comparison between the pipeline of the CV based approach from our previous work (178) and the method proposed on this work based on DL. It is possible to notice that the approach based on CV predicted a wrong anchor point and slightly worse rotations X and Z

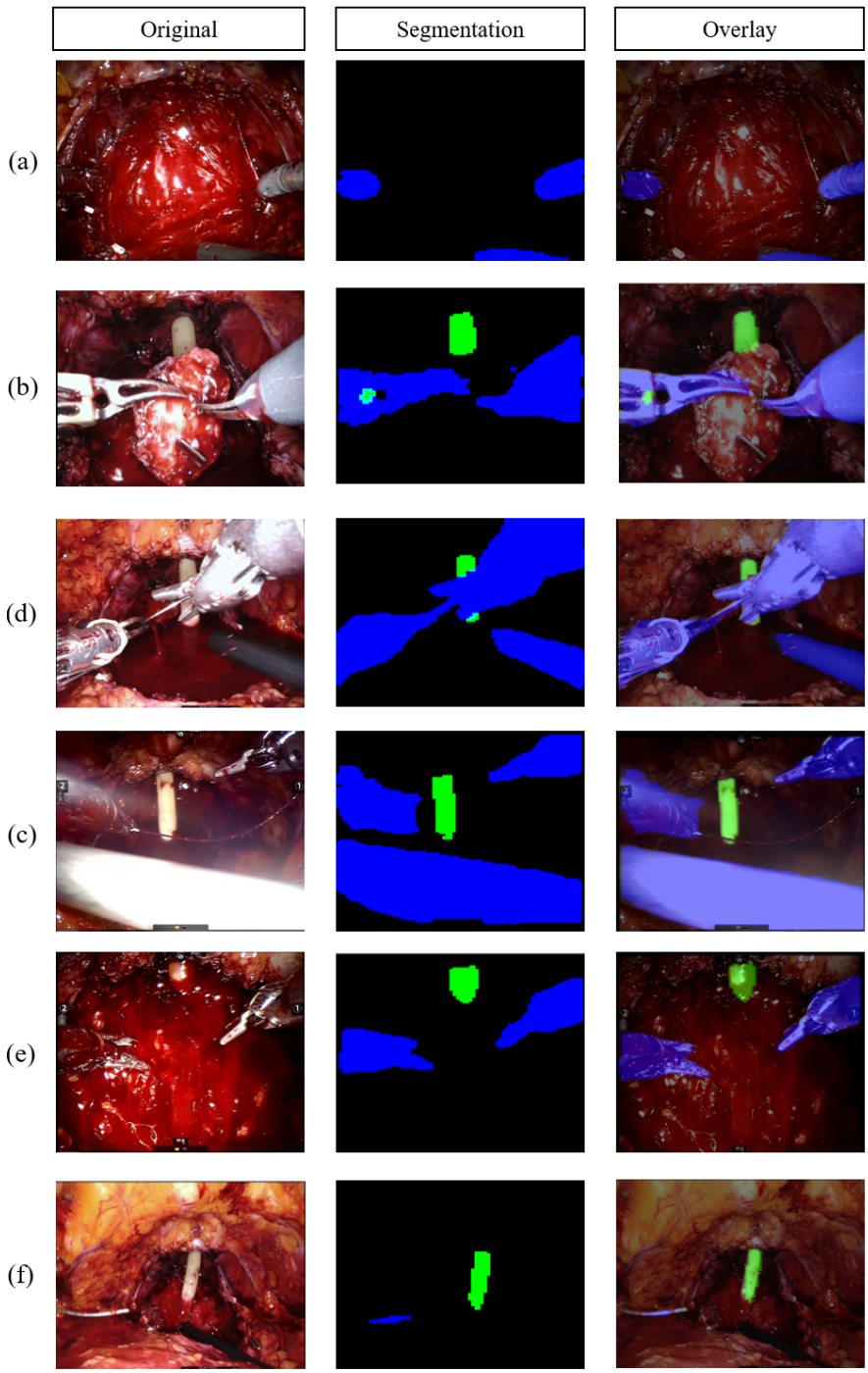


Figure 36 Some segmentation masks returned by our network and the consequent error of the 3D overlay. It is possible to notice how the network performs good even in borderline situations: when the catheter is partially occluded (a), when a tool is featured in the foreground (b) and when the camera is far from the scene (c)

3.1.2 Second Approach: RotationCNN

Introduction

In order to superimpose the preoperative model on the intraoperative image, it is necessary to know the six degrees of freedom, i.e., three parameters for position and three for rotation along the three X, Y, and Z axes. These six parameters allow to univocally determine the position and orientation of an object in three-dimensional space and are referred to as its 6D pose. As higher demand for new scenarios has risen, this topic has been highly investigated in recent years. Different kinds of information is used to address this task, from texture, color, and geometry to depth information, since lately a great development of hardware has occurred, nevertheless, depth-based approaches have not been considered for this study. After a comprehensive literature review of the current methodologies for 6D pose estimation, the top performers were selected. Considering these approaches, a novel framework was proposed to support the surgeon in the visualization and localization of surgical targets and critical structures of interest, such as endophytic or posteriorly located tumors, that would otherwise be arduously identified. To this purpose, a virtual 3D model of the patient's organ, obtained from the preoperative imaging [212], was intraoperatively and automatically superimposed on the endoscopic view and displayed on the screen of the surgery robotic system in use, to guide the surgeon during the incision. Given the variety of solutions presented in the literature using different data and devices, this methodology is an easy-to-integrate alternative, whatever robotic surgical system is in use, as it does not add hardware complexity.

The core of the framework is to recover the 6D position of the organ in real-time, relying solely on endoscopic video RGB frames. In other words, to address this task, we could rely only on the elements present in the endoscopic video, i.e., the rigid instruments, which are inserted and remain static during the operation, and the organs, which are deformable and not easily distinguishable from the surrounding tissues.

Three phases could be defined: 1) localization, 2) registration, and 3) tracking.

To localize the different elements in the scene, the first step consisted of semantic segmentation (SegmentationCNN), consisting of assigning class labels to the pixels of the image, obtaining color-coded segmentation masks of the classified objects, while, in the second phase, a CNN addressed the rotation estimation (RotationCNN) to perform the registration. Depending on the element used as a reference, the

RotationCNN showed a lack of robustness. When rigid instruments were figured out, it proved to be a stand-alone solution, and it was possible to indirectly retrieve the 6D organ position using some specific surgical instruments as references. When it was tested on organs, it was necessary to add to the methodology an alternative third approach, based on Optical Flow (OF), to mitigate the lack of robustness of the RotationCNN. To this aim, since consecutive frames have a high correlation, instead of inferring the 6D pose for each frame through the RotationCNN, the Optical Flow, starting from an initial frame, deduces the 6D pose by performing a pixel-wise tracking of the organ. In this way, the temporal information of the video flow is exploited. The SegmentationCNN was tested by evaluating the Intersection-over-Union (IoU) metric, while the RotationCNN and the OF, were tested by comparing the predicted values with the actual ones. The presented framework was conceived as a generalized procedure capable of handling different surgical operations. To demonstrate the generalization capability in terms of both the surgical procedure and the element used for the registration, two surgical operations, namely the Robot-Assisted Radical Prostatectomy (RARP) and the Robot-Assisted Partial Nephrectomy (RAPN), were considered, and three case studies were taken into consideration: the first involving a static artificial element, such as the catheter, while the second and the third relying solely on organs, i.e., the prostate and the kidney.

Related Works

6D position estimation is an essential task in many Computer Vision (CV) applications. It concerns, as well as the surgical field[213,214], robotics [215], autonomous driving [216], and Virtual/Augmented Reality (VR/AR) applications [217] and is extensively used in the entertainment industry [216]. The problem itself is simple and consists of determining the 3D rotation and translation of an object whose shape is known in relation to the camera, using details observable from the reference 2D image. However, achieving a solution to this problem is not trivial [215]. Firstly, due to auto-occlusions or symmetries, the objects cannot be clearly and unequivocally identifiable. Moreover, the image conditions are not always optimal in terms of lighting and occlusions between the objects represented in the picture [218,219]. In these situations, it is often necessary to add an earlier stage of object detection or localization to distinguish the area of the image which contains the object, before estimating its position.

Although this problem has been studied for many years, the advent of DL gave a renewed push to find a solution, as well as it fostered research in other fields of

application, such as the medical[200,220,221] or face recognition[222,223] domains. Old pose estimation methods were based on geometrical approaches, for example Feature-based methods[224], which tried to establish correspondences between 3D models and 2D images of objects by using manually annotated local features. With texture-less or geometrically complex objects, it was not easy to select local features. In these cases, even though the matching phase usually takes much time, it might fail and provides a not always accurate result [225].

In opposition to these methods, researchers introduced Template-based methods [226,227], which represented the 2D object from different points of view and compared these representations with the original image to establish the position and orientation. These approaches were very susceptible to variations in lighting and occlusions even if they could manage texture-less objects and required many comparisons to reach a certain accuracy level, increasing the execution time[217].

With the diffusion of DL, researchers improved traditional methods by introducing Learning-based methods, making them more efficient and performing. The basic idea of these systems involves CNNs to learn a mapping function between images with three-dimensional position annotations, and object 6D position. Some of these systems [228,229] employ a CNN to predict the 3D bounding box, and then a PnP algorithm, which calculates the 6D position from matches between 2D features on the test image and 3D points on the CAD model [230]. Other types of Learning-based methods, instead, need only a CNN to resolve a classification [231,232] or a regression problem [233,234]. For this reason, Learning-based methods are referred as Bounding box prediction and PnP algorithm-based, Classification-based, and Regression-based methods, respectively. These methods can reach very high levels of precision, but they need many data to train the network accurately and to be able to work well in real cases. Alternatively, CNNs can be used to execute the most critical steps of traditional methods to join the advantages of the various strategies into the final solution [225].

After identifying the golden standards of 6D pose estimation in various fields, works related to the laparoscopic surgery augmentation-specific purpose were investigated. This second phase of the review aimed to analyze the surgical goal to understand the variety of contexts of use and to find out if any effective method was already proposed.

The most popular 6D pose estimation algorithms, mainly focus on objects with distinctive shapes or textures. These characteristics make the process less complex than the one here addressed. As a matter of fact, anatomical structures do not have easily recognizable patterns, yet their texture and shape depend on many factors, from the anatomy of the patient to even his/her position or pathology. For this

reason, every approach used for anatomical purposes should be produced as intended to be generalizable for different patients and, at the same time, sufficiently accurate for the specific case. Most of the medical applications are referred to abdomen laparoscopic procedures; since the real-time video of the surgical scene is captured and displayed on an external monitor, it can be easily accessed to process the intraoperative information.

Applications were proposed for kidneys, prostate, uterus, and liver [235]. Depending on the organ some approaches were more suitable than others [236,237]. The aim of these applications is to perform a first registration of the virtual 3D model on the endoscopic view and the tracking phase, in which the model adapts its position and orientation in accordance with the real organ.

These tasks are challenging mainly because of the lack of stationery features to fall back on, as there are discrepancies between the preoperatively obtained data and the intraoperative real-time video flow. Performing the pneumoperitoneum, namely the insufflation of air or another gas in the peritoneal cavity, the abdominal cavity undergoes some deformations that make the target in the scene different from the one from the preoperative images. In addition to this, further deformations due to tool interactions, as well as smoke and blood presence, must be dealt with. For this reason, keeping track of the target position and recovering it, if needed, is fundamental.

The first and easiest method of anchoring is manual registration [238,239]. Efforts have been made to minimize the human contribution, by introducing semi-automatic systems, pause and editing approaches [238,240], deformation handling [236,239] and fully automatic algorithms [214,241].

For what concerns the tracking phase, the artificial model has to follow the organ movements in real-time. This task can be performed following four main approaches [235], which rely on techniques of increasing complexity. The first method, known as interactive, mainly exploits human assistance, such as the pause and editing approach, analogously to manual registration [242]. In some cases, artificial or anatomical cues can be used as fiducial markers, to perform a point-based match between the endoscopic organ and the virtual model, since they are visible both preoperatively and intraoperatively [214,239]. On the contrary, surface-based methods focus on the intraoperative perspective rather than on preoperative data, because the surface is intraoperatively reconstructed directly on laparoscopic images and registered only at a later stage [243,244]. Finally, the volume-based methodologies are the most complex ones, as they require an intraoperative imaging system in addition to the endoscope, to better locate the hidden structures [245]. On the other hand, double registration is needed, between preoperative and

intraoperative volumes and between the preoperative volume and the endoscopic image.

Moreover, the previous matching techniques are mainly based on the rigidity hypothesis, meaning that organ deformations are not considered, hence the registration accuracy may be affected. To mitigate this problem, deformation models have been introduced to intraoperatively modify the preoperative model, aiming to fulfill the compromise of adaptability and accuracy [236,239].

Up to now, the most investigated methods in the surgical field require, in addition to the two-dimensional RGB image, other types of data or other processing steps in the intraoperative stage. Although it is helpful to have supplemental data, the introduction of new sensors and possibly additional cameras is a cumbersome procedure that would lengthen the operating time of the procedure, as well as increase its cost, and introduces complications such as the synchrony of these elements. This work intends to propose a more immediate and time-saving novel methodology, able to perform a real-time registration relying on RGB intraoperative monocular images only.

Materials and Methods

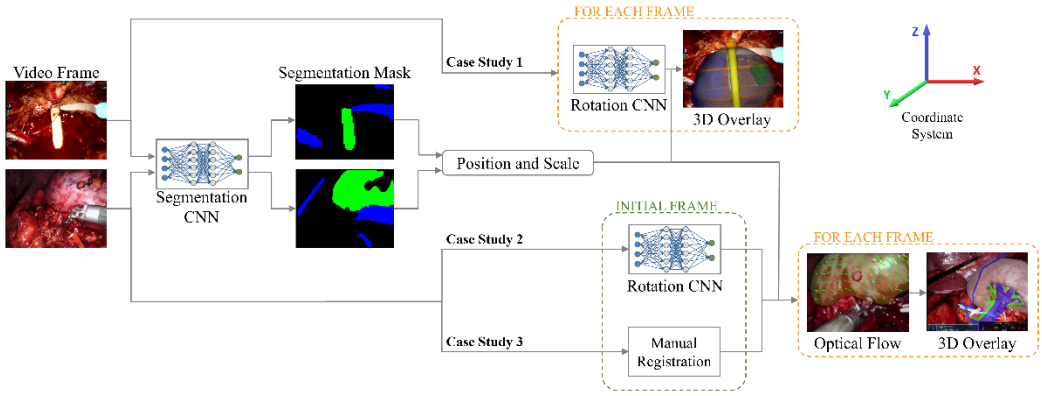


Figure 37 Schematic illustration of the system workflow. The video frame is fed to a SegmentationCNN and a RotationCNN, depending on the Case Study. Target's position and scale are obtained through the SegmentationCNN for every Case Study. Target's orientation is retrieved in different ways depending on the Case Study: RotationCNN output the predicted rotation for each frame (Case Study 1); RotationCNN for registration in an initial frame and OF tracking for next frames (Case Study 2); Manual Registration in an initial frame and OF tracking for next frames (Case Study 3).

The methodology of the proposed framework, represented in Figure 37, consists of two subsequent phases and an alternative third phase. Firstly, the RGB image, captured by the endoscope, is fed to a first CNN to perform the semantic segmentation of the actors. A segmentation mask is returned, which distinguishes the objects in the scene, in this way, the scale and position values of the element

used as reference (colored in green) are obtained. Contextually, the RGB image is fed to another CNN, which returns the rotation values. Depending on whether the main actor is a rigid instrument or an organ, the RotationCNN could show different performances. For this reason, in some cases, it would be more accurate to deem this kind of registration as an initialization step, performed on images in which the organ is fully exposed and well-framed by the camera.

Thus, three different use cases were considered:

1) When rigid instruments are present, such as the catheter (Case Study 1), the RotationCNN can be applied recursively, merging registration and tracking tasks in the same method. The reason why the rotation can be more easily retrieved, is that, among the three axes, only one is effectively unknown. Indeed, the rotation on the Z-Axis is irrelevant because of the catheter's symmetry, while the rotation along the Y-Axis can be successfully retrieved from the semantic map [200], as proven by results in Table 21. Thus, only the rotation along the X-Axis should be predicted.

2-3) When the organ is the only information to fall back on, different problems arise, such as organs' deformability, difficulty in distinguishing organ's texture from the surrounding tissues, and the fact that their rotation is bound to a fixed anatomical constraint. In this case, the RotationCNN should predict three rotations, as the one along the Z-Axis is now relevant and the one along the Y-Axis is not retrievable with precision from the semantic map. For this reason, when the organ has a particularly recognizable shape in ideal frames, such as the prostate (Case Study 2), the RotationCNN can be used for registration, while in other cases, when the organ presents a more complex structure, such as the kidney (Case Study 3), the initialization should be manually performed. After the semi-automated registration, the organ is tracked using a CV-based approach, which exploits an OF motion analysis algorithm. This technique performs the pixel-wise tracking of the organ to infer the frame-to-frame rotation and maintain the real-time superimposition of the model correctly oriented.

The information from the two CNNs is combined for locating and orienting the 3D virtual model, to overlay it on the endoscopic view and displayed it on the monitor of the robotic surgical system. The CNNs, used for these different tasks, are detailed in the next sub-sections.

Datasets

The dataset creation strategy was different depending on the semantic segmentation or the rotation estimation CNNs. Both the approaches are detailed below.

SegmentationCNN

The images employed for the dataset creation were acquired from 9 endoscopic videos of different patients, recording the laparoscopic procedures in analysis. In particular, 5 videos referred to the prostate and, depending on the phase of the procedure, frames were extracted for Case Study 1 and 2, while 4 videos, referring to kidney, were used for Case Study 3. The average length of the procedures' videos was about 20 minutes, from which the most significant portions, from few seconds to 5 minutes each, were selected. Among the extracted frames, only those belonging to portions of the procedures in which the target object was well framed by the endoscope were actually used for dataset creation. The SegmentationCNN output, for each frame, a segmentation mask, containing three different classes, each marked with a unique ID.

RotationCNN

To infer the organ rotations along the three axes through a Learning-based approach, the CNN must be trained on images in which the organ rotation values are provided. This is a troublesome task to perform through manual rotational tagging on real images because it would be heavily operator-dependent, hence unreliable. To avoid this source of uncertainty, a synthetic dataset was created in *Blender*, starting from the preoperative 3D virtual model of the target. The actual dataset creation is performed by a Python code, that renders synthetic images after computing scene changes such as: tools position and rotation, texture rotation, light changes, and organ rotation. The main advantage of the synthetic dataset is that the rotation of the organ is known, as it is controlled by the script itself.

Location and Scale Inferring

Given an RGB image, the first step of the organ *localization* is the semantic segmentation task, which aims to produce a matrix having the same size of the input image containing the predicted class identifiers corresponding to all pixels. To choose the network model, different attempts were made by combining pre-trained base models and segmentation architectures. Among all the combinations between architecture and methods, UNet architecture with ResNet as base model has shown the best test accuracy, as usually happens for medical images [246].

By applying the segmentation model to an intraoperative video, the target semantic mask can be visualized. To extract the target location, the irregular mask is circumscribed by a fitting ellipse, the center of which is retrieved. Its coordinate is assigned to the center of mass of the target's 3D model to perform the anchoring and the real-time superimposition translation in the frame space. The same ellipse is used to infer the scale factor of the frame, meaning the distance from the camera, as its dimension is proportional to the one of the organs that is fitting [200].

Rotation Inferring

To perform the *registration* task, the rotation values must be first estimated. The RGB image, used as input of the SegmentationCNN, is fed to the RotationCNN, based on a modified version of ResNet50 model, with a different branch for each axis rotation. Every branch has the same structure, which contains: a *Dense Layer*, with 4096 neurons and a ReLU activation function; a *Batch Normalization Layer*; a *Dropout Layer*; a *Classification Dense Layer*, with several neurons equal in number to the current axis classes, and an *Activation Layer*, which applies the *Softmax* activation function to the output. The estimation of the rotation along the three axes was solved as a classification problem. The set of possible rotation values along an axis was discretized in N classes, according to the range of orientation angles the target can assume during the surgical operation. The output neuron, corresponding to the highest probability value according to the *Softmax* activation function, fires and produces the corresponding rotation value. An overall representation of the Rotation Neural Network for the three axes rotation prediction case is presented in Figure 38.

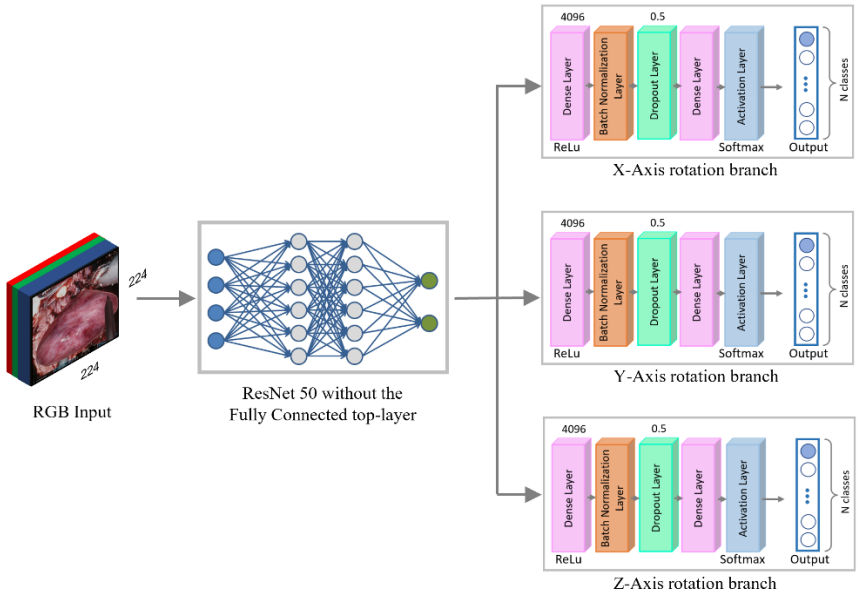


Figure 38 Rotation Neural Network overall architecture, which involves a modified version of the ResNet50 model, with a branch for each axis rotation.

Optical Flow

In case the target is the organ, the RotationCNN prediction is often not robust enough, hence must be manually corrected by the operator. The subsequent organ *tracking* is then addressed by the OF algorithm. The advantage of this method is the use of temporal information. As a matter of fact, the registration of the model, solved as a tracking task, is more easily retrievable, but not necessarily more accurate than if addressed by inferring the 6D pose within a single frame, due to the high correlation consecutive frames have. Through OF algorithm, motion analysis can be performed, by extracting the displacements, between consecutive frames, only regarding the organ pixels. OF returns two vectors, the magnitude and the angle direction of the displacement, used to display the displacement arrows on the image. The different distribution of the arrows direction on the organ contains information about the rotation trajectory of the organ. Since the framework is applied when the organ is *in situ*, it is constrained to the surrounding anatomical structures. Therefore, the organ cannot move freely in the abdominal cavity, so the flow vectors do not describe pure translation, but they also contain rotation information. From this assumption, it was hypothesized that the magnitude of the vectors is an indicator of the rotation angle, while their direction is an indicator of the rotation axis. Once peculiar characteristics within the specific case of the organ

movement were identified, according to the considered case studies, it was observed that, when the displacement follows a particular pattern, it is possible to assume the rotation direction. Since these assumptions cannot be generalized, a heuristic was written to identify and quantify organ rotation.

Training and Metrics

To evaluate the CNNs, two different metrics have been involved. The SegmentationCNN model was assessed again by the *Intersection-over-Union* (IoU) metric, already defined in the previous section. The RotationCNN was evaluated by comparing the predicted and the actual rotation values on the X, Y and Z axes. Both the CNNs ran on an NVIDIA Quadro P4000 GPU, involving the open-source Keras library for neural networks, which is written in Python and run on top of TensorFlow.

Case Studies

The proposed framework is designed to support the surgeon during critical phases of robot-assisted laparoscopic surgical procedures. In particular, the urological field was considered, as it deals with organs with symmetrical, rounded and not very distinctive shapes and textures, that make them hardly distinguishable from the surrounding anatomical parts. These drawbacks can be mitigated by AR solutions. The opportunity to visualize 3D reconstructions overlayed on their real counterpart introduces several benefits as stated in literature [235,247], as for example: exploiting AR during preoperative planning for tailoring incisions and cutting plans; reducing surgery time; augmenting the surgeon's spatial perception of the surgical field, compromised by the use of the robot console, avoiding unnecessary manipulations or accidental injuries to inner organs; reducing the surgeon's distractions by allowing him or her to access the augmented information contextually without having to look for it in a different monitor.
In this work, the datasets are parts of intraoperative endoscopic videos from different patients and were provided by the "Division of Urology, University of Turin, San Luigi Gonzaga Hospital, Orbassano (Turin), Italy". AR has been used to support the surgeon during the following case studies.

Robot-Assisted Radical Prostatectomy (RARP)

Radical Prostatectomy removes the prostate gland and tissues surrounding it in case of prostate cancer. The precision of this procedure is fundamental for patient well-being to avoid short and long-term complications. The RARP procedure can be

subdivided into five stages: 1) defatting and incision of the endopelvic fascia, 2) management of the bladder neck, 3) vase clamping and nerve-sparing, 4) surgery by the prostatic apex, and 5) targeted biopsy. According to surgeons, phases 3 and 5 could benefit the most from AR, for which appropriate registration techniques have already been implemented [248]. In both phases the overlay of the organ would be a significant help for the operating surgeon [249]: in the vase clamping and nerve-sparing phase, AR is helpful for accurate nerves and vases localization, crucial for preserving the patient's well-being; while AR guided tumor localization in targeted biopsy is essential, in case of extracapsular tumor, to verify that the extra parenchyma part, which may have been in contact with the surrounding tissue, has not compromised it leading to metastases development later on. In phase 5, a catheter is inserted, and it could be used as a target to overlay the 3D model (Case Study 1). On the contrary, in phase 3 the 6D information has to be retrieved from the prostate (Case Study 2).

Robot-Assisted Partial Nephrectomy (RAPN)

In Partial Nephrectomy, only the affected portion of the kidney is excised to preserve its functionality. Analogously to RARP, the RAPN procedure can be subdivided into five main stages: 1) bowel mobilization, 2) hilum dissection, 3) tumor identification and demarcation, 4) clamping of the hilum, and 5) tumor excision. According to surgeons, the procedure phase that benefits the most from AR is phase 3, which allows the surgeon to quickly identify endophytic and posteriorly located tumors [250], since during this procedure the kidney is not perfused with blood and consequently risks necrosis. Thus, the kidney was used as a target (Case Study 3).

Concerning the RARP and RAPN data at disposal, main datasets parameters, namely number of real or synthetic images for both the CNNs and rotation ranges, are resumed in Table 19. In all cases, datasets, for both segmentation and rotation, were split into 70%, 15%, 15% for train, validation and test, respectively.

The network used for segmentation was a UNet-ResNet architecture, trained for 50 epochs using a batch size of 4 and an Adam optimizer with a learning rate of 0.0001, while the one used for rotation was a modified version of ResNet50, trained for about 15 epochs, using a batch size of 32 and an Adam optimizer with a learning rate of 0.00001.

Table 19 Values referring to the images used for datasets creation, both synthetic and real, for segmentation and rotation CNNs. For the latter, considered rotation ranges are also shown. For the catheter, the RotationCNN was trained to predict only the X rotation, as Y was derived directly from the segmentation map and Z was considered irrelevant.

Surgical Operation	Case study	Target	Segmentation	Rotation	
			# real images	# synthetic images	Rotation ranges in degrees (-X,+X) / (-Y,+Y) / (-Z, +Z)
RARP	1	Catheter	375	40 000	(-40, 10) / - / -
RARP	2	Prostate	388	35 000	(-15, 20) / (-25, 25) / (-5, 15)
RAPN	3	Kidney	208	40 000	(-10, 10) / (-10, 10) / (-10, 10)

Results

SegmentationCNN

The SegmentationCNN showed optimal results for all case studies, represented in Table 20. Before selecting the final architecture, several of the most used models for semantic segmentation were compared. The final choice was made relying on the IoU metric, computed both on the individual classes (*background*, *tool*, *target*) and as an average between them. The IoU score for the target class was used as a discriminating factor, since organ identification is the main goal of the semantic segmentation task. Among all the model combinations, UNet-ResNet was finally deemed as the best, as it showed the highest IoU score in terms of both target class (0.9450 for the catheter, 0.7296 for the prostate and 0.8602 for the kidney) and mean value (0.8940 for the catheter, 0.8067 for the prostate and 0.9069 for the kidney). In Figure 39 representative samples of the network's output masks, together with their overlay on the real images, are shown.

Table 20 IoU scores referring to the mean value of the actor classes (background, tool and target) and the target class only: catheter (C), prostate (P) and kidney (K).

	IoU Score	
	Target Class	Mean
C	0.9450	0.8940
P	0.7296	0.8067
K	0.8602	0.9069

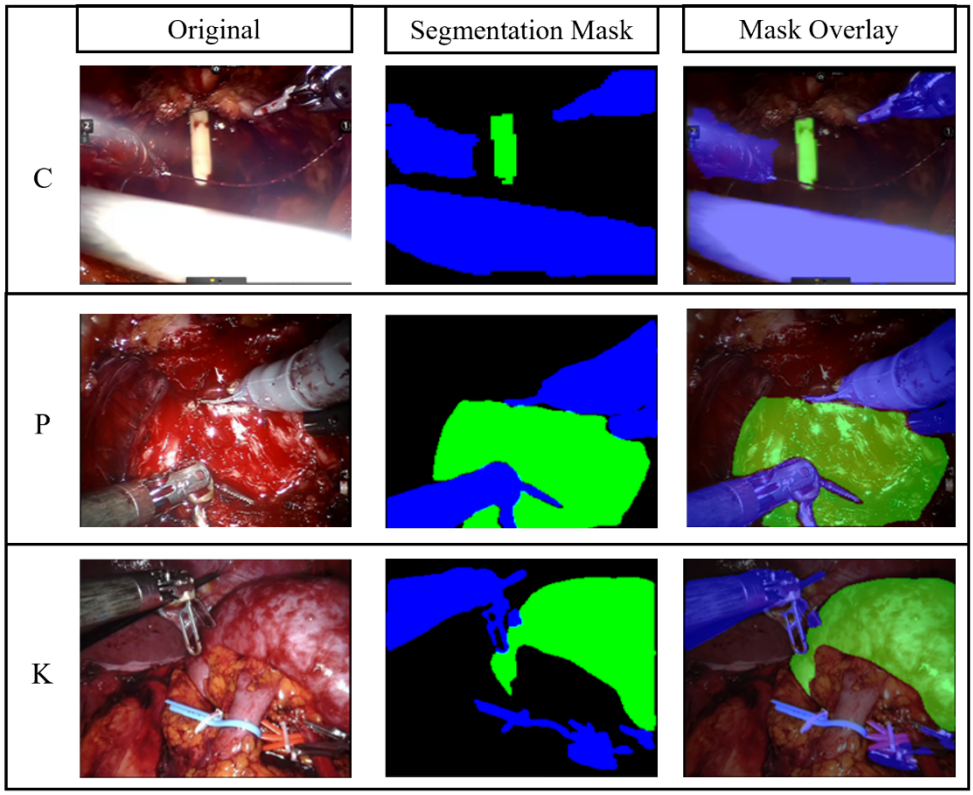


Figure 39 Sample images referring to original endoscopic images, segmentation masks and mask overlay (columns) for each case study (rows): catheter (C), prostate (P) and kidney (K).

RotationCNN

The results related to the rotations' prediction are resumed in Table 21. The network was first tested with 3000 synthetic images for catheter (C), 1750 for prostate (P) and 3000 for kidney (K) generated with *Blender*. The accuracies for *X*, *Y* and *Z*, were computed as the number of correct predictions over the total number of samples, using different acceptable ranges. In the testing phase, it was decided to consider 5 degrees of tolerance and therefore a prediction was deemed wrong in case the difference between the estimate and the actual value was greater than this threshold. This was considered a good compromise as a difference of 5 degrees does not significantly affect the visual result of the model superimposition. For the same reason, a 10 degrees tolerance range was considered acceptable by the surgeons. Each evaluating parameter, namely, accuracies for [-5, +5] and [-10, +10] ranges, were computed for each one of the rotation axes (X, Y, Z) and for C, P and K classes, respectively. This same network was then tested with 30 real images tagged manually for C, 36 for P and 30 for K. The accuracies in this case were predictably acceptable only for the catheter. To improve the results for the prostate

and the kidney, the network was subsequently fine-tuned with 100 additional real images and tested with the same images as before.

Table 21 Values of mean accuracies when testing with a synthetic dataset, a real dataset and the same real dataset after fine-tuning the network. Each value is computed for the considered target classes: catheter (C), prostate (P) and kidney (K). The accuracies for X, Y and Z, were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: prediction with an error in the range [-5, +5] and in the range [-10, +10]. *Y-Axis rotation values for C were retrieved from the semantic segmentation CNN.

		Accuracies [-5, +5]			Accuracies [-10, +10]			# of images
		X-Axis	Y-Axis	Z-Axis	X-Axis	Y-Axis	Z-Axis	
Synthetic Dataset	C	0.9987	-	-	0.9987	-	-	3000
	P	0.9971	1	0.9983	0.9994	1	0.9994	1750
	K	0.9990	1	0.9990	0.9997	1	0.9987	3000
Real Dataset	C	0.5333	0.9457*	-	0.7333	0.9886*	-	30
	P	0.5000	0.0000	0.2778	0.5556	0.0833	0.3611	36
	K	0.0000	0.0000	0.0000	0.0000	0.0000	0.2000	30
Real Dataset (After Fine Tuning)	P	0.3333	0.5000	0.6667	0.7778	0.7222	0.9444	36
	K	0.0000	0.0000	0.5000	0.0000	0.0000	0.7333	30

Optical Flow

The motion analysis algorithm was tested on an endoscopic video stream of a nephrectomy procedure fragment lasting 10 seconds. The results, comparing estimated and real rotation trends, are shown in Figure 40. The evaluation of the method was performed by comparing the tagged (blue values) and the inferred (red values) rotations for each axis separately. The rotation values were extracted for 30 sample frames, namely with a 10-frame-step, adding up the rotation variations of all the in-between frames. Analogously to the *RotationCNN* results, tolerances of 5 and 10 degrees were considered to deem an estimation as correct.

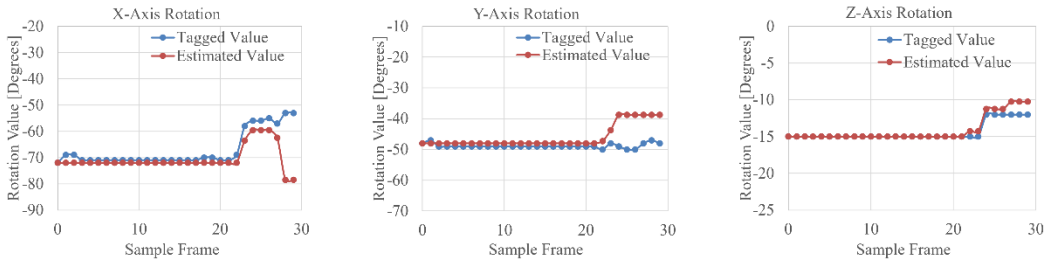


Figure 40 Trends representation of organ orientation on X, Y and Z axes comparing tagged (blue) and OF estimated (red) rotation values for each sample frame.

Discussion

In recent years Artificial Intelligence, Computer Vision and Augmented Reality technologies have spread to many fields, including surgery. In this article, the use of these technologies in supporting applications to the surgeon during laparoscopic Robot Assisted procedures was addressed. These technologies aim to augment the surgeon's perception of the surgical scene, which is usually more limited than in open procedures, as the field of view is narrowed. To mitigate this drawback, AR aims to superimpose the preoperative model on the intraoperative image, hence it is necessary to know the 6D pose of the organ.

After extensive literature analysis, both in the currently used methodologies for 6D pose estimation of an object and in the specific context of its application to AR in laparoscopic surgery, the discussed methodology was delineated. The proposed framework is a novel approach for automatic and real-time registration of the 3D model on the endoscopic view as it relies solely on RGB data, captured by the endoscope and displayed on the screen of the robotic surgical system, with no need for further external sensors or devices. Three case studies were identified and, depending on them, different ways of handling the problem were pursued. Case Study 1 relies solely on a specific rigid instrument, the catheter. Case studies 2 and 3 dealt with soft tissue having different complexity levels.

The first phase involved a SegmentationCNN, UNet-ResNet, for all the case studies, to distinguish the actors and obtain the target position and scale. In the second phase, a RotationCNN predicted the rotation values along the three axes, *X*, *Y* and *Z*. In the third additional phase, an OF algorithm handled the target tracking.

In Case Study 1, the approach handles both the location and rotation estimation tasks recursively for every single frame while, in Case Study 2, the rotation estimation is performed only for a starting ideal frame, then, the next frames rotations are inferred through OF. These two approaches are fully automated. On the contrary, in Case Study 3, given the complexity of predicting the rotation, the registration should be manually performed before applying the OF technique.

The SegmentationCNN showed optimal accuracies, with a mean IoU score greater than 80% for all the case studies.

For the RotationCNN, accuracies for [-5, +5] and [-10, +10] ranges, were computed for each one of the rotation axes (X, Y, Z). Test results for synthetic images reached values close to 1, showing that the network actually learned. To evaluate the generalization capability of the network, real images were tested too, showing, acceptable results just in the first case study involving the catheter, aided by the fact

that the Y accuracy was derived from the segmentation map. On the contrary, the results for the prostate and the kidney were not considered acceptable. Thus, fine-tuning was performed with 100 additional real images for each organ. After fine-tuning, the network underwent major improvements with the prostate, resulting in error values minor than 10 degrees for almost every image. These tests were performed on frames where the organ was well distinguishable. On the contrary, with the kidney, the network did not improve its performance even after fine-tuning. The only solution in this case was to perform a manual registration. Afterwards, an alternative method was included to tackle the rotation inferring task, the OF. This last approach was evaluated on an endoscopic video stream, showing that the general trend of the organ's orientation changes was correctly tracked, hence rotation events were recognized.

Nevertheless, there are still several limits to overcome. In particular, one of the limitations of the current application is the restricted amount of videos for the CNN training, which limits the heterogeneity of the dataset, in particular the organs have different shapes and textures depending on the patient's anatomy and health condition. In addition, the case studies addressed all fall within the urological field, hence, to effectively test its generalizability, it should be broadened also to other organs such as the uterus and liver. Another limitation for the generalization of the framework refers to the heuristic that infers rotation from OF, since this was deduced from specific movement patterns of the organ. The RotationCNN performances have still room for improvement, since it still struggles in detecting the value of the rotation precisely enough to perform the tracking when dealing with soft tissues.

Considering each case study, when satisfactory results were obtained during the test phase on the endoscopic videos, in vivo tests were performed. In particular, the first Case Study, meaning when the catheter is present, was successfully tested in vivo, and it worked in real-time. For what concerns the other case studies, dealing with complex scenarios, are currently being tested, gaining positive feedback from surgeons and new insights for further improvements.

In future works, adjustments could be made in several directions. For the concerns of segmentation, instead of traditional CNNs, the novel attention technique could be exploited to focus on those pixels that are informative of the motion information, fading out the others, resulting in a faded mask rather than in a binary one. For the concerns of rotation estimation, besides testing the RotationCNN in the operating room, the heuristic inferring of the OF method should be amplified and its precision improved. Acquiring a greater number of endoscopic procedure videos could also

benefit the training of the neural networks. To better exploit intraoperative information, more data should be collected, for example an organ could have different shapes and dimensions depending on the patient and, the more samples the better the ability of the neural network to improve generalization. Furthermore, the current version of this work only used one video stream from a monocular view to reducing processing complexity, yet, in future research, a binocular vision could be used to exploit the depth information of stereoscopy. Finally, for the concerns of the real-time tracking task, variations should be made to control whether the organ is fully framed or not, in order to correctly detect the center of the organ under different conditions.

In the next section, we will discuss an improvement to this work by introducing a preliminary event classifier that detect blood accumulation, but that could be extended to any type of adverse event.

3.1.3 Blood Accumulation

Introduction

Laparoscopy, widely known as minimally invasive surgery, is a surgical procedure that allows a surgeon to see within the abdomen and pelvis without creating significant cuts [251], by having trocars with attached instruments controlled from consoles by the main operating surgeon and the assistant operator [252]. With the advancement of technology, robotic-assisted laparoscopy has grown in popularity during the last few decades, and it is now extensively and thoroughly used in surgical procedures, replacing most open surgeries. Its advantages include a smaller incision, better aesthetic outcomes, less discomfort, faster recovery, and consequently shorter postoperative hospital stay, a decreased risk of infection, and no oncological drawbacks in cancer patients [252–254]. Furthermore, the laparoscope's enlarged vision allows surgeons to observe anatomical structures in detail and accurately dissect, suture, and repair them [253].
On the other hand, there are several drawbacks to consider: the surgical approach is different, the workspace during laparoscopic surgery is smaller than for open surgery, lowering the field of view, the difficulty in maintaining hemostasis rises, the extra personnel is costly, it delegates too much work to humans who are prone to making mistakes, the primary operator and the assistant operator have communication blunders due to not staring at the same console [252,253,255].

In laparoscopy, the digital camera allows surgeons to obtain assistance from image-guided surgical technologies. Deep learning approaches have been employed in minimally invasive surgery for software and hardware-based solutions since enhanced computing power and recent breakthroughs in this field enable a standard computer to comprehend the content of an image or video stream in real-time. Some applications are real-time localization through object detection or semantic segmentation and tracking of anatomical structures or surgical instruments, simulation of surgeries for medical study purposes, and hemorrhage detection during surgery [254,256].

Dealing with intraoperative bleeding is one of the most difficult aspects of laparoscopic surgery [257] accounting for 23% of all adverse events [258]. Since the operation field is always restricted by the laparoscopic monitor and blood fills the bleeding site rapidly, it is difficult to quickly stop bleeding during laparoscopic surgeries [255] and without prompt treatment, bleeding can have serious implications or even be fatal [259]. Any sort of surgical manipulation, including suction, grabbing, retraction, cutting, and dissection might result in rapid bleeding. Bleeding can also happen afterward, for example, if a vessel is not completely sealed, it might burst spontaneously without direct touch. Bleeding can occur from a visible or macroscopic blood vessel (artery or vein) and the microvasculature and capillary network within tissues [260]. Arterial bleeding is more rapid and difficult to stop than venous hemorrhage. Venous hemorrhage is characterized by the widespread distribution of the blood pool, whereas arterial hemorrhage is always characterized by discrete tiny blood columns [255]. Therefore, immediate, and precise diagnosis of bleeding spots is important to surgical success, and reactive procedures will be necessary to address these stressful events.

Some efforts have been undertaken in recent years to speed up bleeding identification in endoscopic procedures. Some methods detected bleeding by utilizing the RGB space parameters blue/red (B/R) and green/red (G/R) [261] or categorizing pixels into "blood" or "non-blood" using color features and support vector machine (SVM) for classification [262–264]. Other studies focused on convolutional neural networks to automatically extract features and detect adverse events [265], segment bleeding sources and display them to the surgeon [257], classify the images into bleeding and non-bleeding [266], or for real-time bleeding point location, recognition, and tracking [255,260]. However, most of this research has limited use in real-time videos, and other studies are not end-to-end, requiring time-consuming additional steps such as stabilizers to minimize jitter and smooth camera path, expensive multi-stage temporal convolutional neural networks, or

Optical Flow input data so that features may be gathered over numerous frames while leveraging temporal information.

The challenge of detecting bleeding in laparoscopic recordings is substantially more difficult, requiring distinctions between the presence of blood residual (Figure 41 (a)) and when blood is actively flowing (Figure 41 (b)), which is the event of interest. Detecting adverse occurrences during laparoscopic surgeries could be defined as an issue of action detection and localization. Action recognition and object segmentation interactions can be mutually advantageous and increase total video understanding. For example, precise positional identification of the key actor participating in action may boost the robustness of action recognition, and vice versa [267].

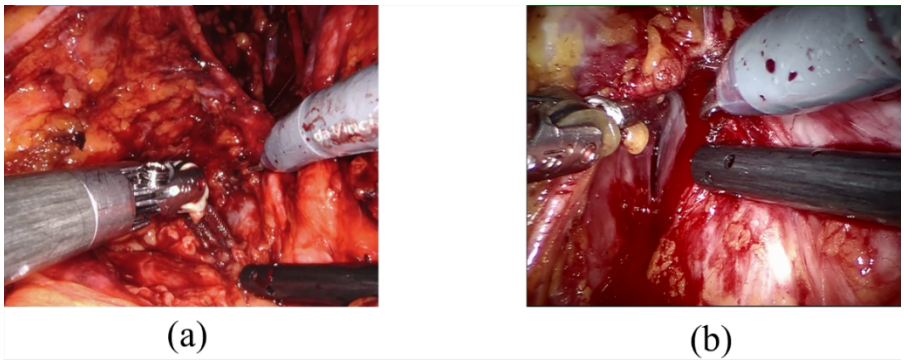


Figure 41 Blood residue on tissues (a) vs. Active bleeding (b)

Multi-task learning (MTL) is one strategy that may efficiently solve this challenge in a unified model. MTL more closely matches the learning process of humans than single-task learning since integrating information across domains is a core principle of human intelligence [268]. MTL is an area of machine learning that adopts a training paradigm in which a shared model learns many tasks at the same time [269]. This strategy increases data efficiency, decreases overfitting by exploiting shared features from multiple tasks, and could improve learning speed by leveraging contextual information, hence alleviating deep learning's renowned drawbacks: high data availability and computation power [268]. On the other hand, learning concepts for numerous tasks introduce problems that are not present in single-task learning. Choosing which tasks to study together is a difficult topic since various tasks may have competing requirements. In this situation, improving a model's performance on one job could damage performance on another with distinct requirements. Furthermore, determining how much of the model's parameters will be shared between tasks, is not an easy issue [268].

To overcome these problems, the authors of [18] suggested a 3D end-to-end deep multi-task model for joint outcome prediction and tumor segmentation from PET/CT images in the healthcare industry. In other research fields, 3D CNNs were employed to extract spatiotemporal characteristics and perform action recognition and segmentation in videos [267,270,271]. In general, these techniques shared the backbone structure for frame feature extraction while decoupling the classification branches since each had its own set of parameters. For improved performance, these systems investigated several methodologies such as the tube proposal network [270], the pyramid pooling module [271], and additional flow input [267].
This study proposes an end-to-end encoder-decoder multi-tasking neural network for joint blood accumulation detection and tool segmentation in laparoscopic surgery. For this purpose, we employed a shared backbone based on the encoder of the U-Net architecture [272], the gold standard for semantic segmentation in medical images. Two separate branches were instead implemented to classify the blood accumulation event and output the segmentation map. Our main contribution is an efficient multi-tasking approach that achieved acceptable results during the test on real-time surgical videos, although trained with only RGB images and no other additional information, commonly unavailable in real applications.

Methods

he current investigation introduced a multi-task convolutional neural network, namely an architecture able to simultaneously learn multiple tasks. Particularly, in the examined case study, the model could jointly perform semantic segmentation and event detection in real-time during laparoscopic surgery. To this aim, a new architecture was implemented, fed with a properly manually labeled dataset. A *shared trunk* architecture was utilized for this purpose, which included a global feature extractor composed of convolutional layers shared by all tasks, followed by a different output branch for each output, performing the same computation for each input of the same task [268]. The weights for multiple tasks were pooled, such that each weight is trained to minimize several loss functions simultaneously. The neural network architecture, the dataset, and the training process are described in the following sections.

Neural Network Architecture

Figure 42 illustrates the network architecture. The backbone is a slightly modified version of the U-Net architecture [272]. The contracting path (left side) represents a global feature extraction, and it is shared by all tasks. It comprises a four-times repetition of the same sequence: two 3x3 convolutions (padded convolutions) to double the number of feature channels, each followed by a rectified linear unit (ReLU) and a down-sampling step, made of a 2x2 max pooling operation with stride 2, to halve the x-y image size. The last max pooling operation is followed by two 3x3 convolutions, which generate the bottleneck of the network. Two separate output branches derive from this bottleneck, each addressing a distinct task. The first branch (right-top side) is adapted from the expansive path of U-Net. Here, every step is symmetrical to the related contracting part. It includes an up-sampling of the feature map, a 2x2 convolution ("up-convolution") to halve the number of

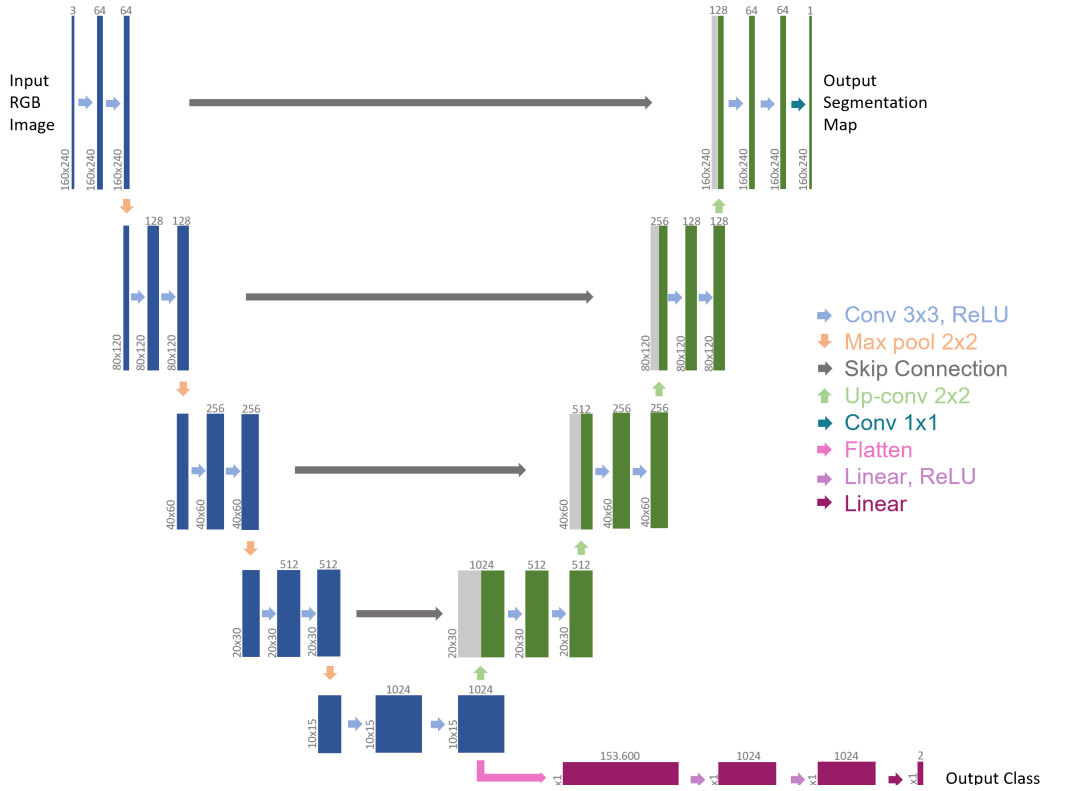


Figure 42 Multi-tasking neural network architecture (example for 160x240 input image). Each box represents a multi-channel feature map. The number of channels is provided on top of the box, while the x-y size is shown at the lower left edge. Blue boxes belong to the backbone, green boxes refer to the first branch for semantic segmentation, grey boxes denote copied feature maps (skip connections), and purple boxes represent the second branch for event detection. The arrows refer to the different operations.

feature channels, a concatenation with the symmetrical feature map of the contracting path (skip connection), and two 3x3 convolutions which double the x-y image size, each followed by a ReLU. The final layer of this branch applies a 1x1 convolution to map each 64-component feature vector to the desired number of classes in the output segmentation map. Unlike U-Net, padded convolutions were employed so that the output segmentation map and the input RGB image had the same size.
The second branch (right-bottom side) receives in input a flattened version of the bottleneck and addresses event detection as a classification problem. This branch is based on a sequence of fully connected layers: two Linear layers with 1024 features, each followed by a ReLU and a Dropout Layer, and a final Linear layer that map each 1024-component feature vector to the desired number of classes.

Dataset

The images were acquired from 26 endoscopic videos recording Robotic Assisted Radical Prostatectomy (RARP), a laparoscopic procedure conducted to remove the prostate gland and tissues surrounding it in case of prostate cancer. These data were provided by the "Division of Urology, University of Turin, San Luigi Gonzaga Hospital, Orbassano (Turin), Italy". The length of the procedures' videos ranged from a few seconds to 15 minutes. Among the extracted frames, only those with good quality were included in the final dataset, which comprised 318 images. The dataset was then divided into train, validation, and test sets which contained 200, 32, and 86 images, respectively. Among them, four videos were kept apart for testing the network in real-time.
The training and validation images were tagged according to the specific task, namely, semantic segmentation and event detection. For semantic segmentation, the tool *labelme* was utilized to manually obtain the masks. Two labels were considered. "*tool*", and "*background*". For event detection, an excel file was appropriately created, which included each image ID and the corresponding label. Two possible classes were considered: 0 for "*no blood accumulation*" and 1 for "*blood accumulation*".
Data augmentation was added during the network training, to improve the numerosity and transformation invariance of the medical image dataset. Particularly, train samples were rotated by a random factor in the range (-35, 35), and randomly flipped, including vertically, horizontally, or both flips.

Training and metrics

The multi-tasking architecture was trained for 30 epochs using a batch size of 32, and an Adam optimizer with a learning rate of 0.0001. A multi-task loss was chosen for parameter optimization:

$$loss = seg_{loss} + cls_{loss},$$

where seg_{loss} is a Binary Cross Entropy Loss function followed by a *Sigmoid* activation function, and cls_{loss} is a Cross Entropy Loss function followed by a *Softmax* activation function. The model ran on an NVIDIA Quadro P4000 GPU, adopting the open-source *PyTorch* machine learning framework, written in Python, and based on the Torch library.

The semantic segmentation branch accuracy was assessed by the *Dice Coefficient (F1 Score)* metric, a diffuse metric for semantic segmentation, defined as:

$$Dice\ Coefficient = \frac{2*Overlap\ Area}{Total\ pixels\ combined},$$

where the overlap area represents the intersection between the pixels belonging to the predicted segmentation masks and those belonging to the ground truth one, and the total pixels parameter represents the total number of pixels in both images. The Dice Coefficient ranges from 0 to 1, where the edge values mean completely wrong and perfectly correct predictions, respectively.

The event detection branch accuracy was calculated as follows:

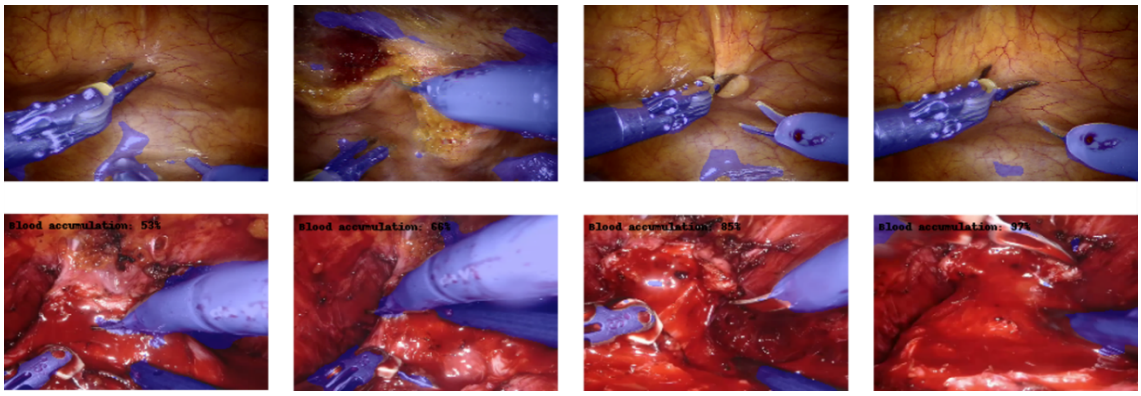
$$Classification\ Accuracy = \frac{\#correct_predictions}{\#samples}.$$

This value was monitored both to examine the accuracy of the entire branch, and the accuracy of each class separately.

Results and Discussion

The implemented multi-task neural network was tested both on images and videos. For images, it achieved without any pre- or postprocessing step a Dice Score equal to 81,89% for the semantic segmentation task, and an accuracy equal to 90,63% for the event detection task. Furthermore, the accuracy for each class was derived, obtaining an accuracy of 86,67% and 94,12% for classes "*no blood accumulation*" and "*blood accumulation*", respectively.

Figure 43 . Sample imaged referring to the multi-tasking neural network outputs, namely the semantic segmentation masks for tools (colored in blue) and the percentage of accumulated blood if greater than 50%.



Afterward, the network was also tested on videos, to evaluate its ability to work in real time. Starting from test videos with a resolution of 1280x720 and a frame rate of 30 frames-per-second, the model output a processed video stream with 15 frames-per-second, which is an acceptable value for near real-time applications. Figure 43 shows some examples of output frames, containing the output segmentation map overlapped on the real counterpart, and the predicted percentage of the "blood accumulation" class, if it is greater than 50%. As can be seen from the figure, both branches provided satisfactory results. The neural network succeeded, although with some marginal imperfections, in correctly recognizing the surgical tools and overlaying the mask. In addition, the event related to blood accumulation was also properly detected, as visible by the fact that the percentage goes above 50 percent when bleeding is effectively visible. In addition, it is noteworthy that during the test on the videos, the network adequately recognized the reduction of accumulated blood in the operating scene as the percentage decreased when the laparoscopic aspirator was removing the accumulated blood. It can be inferred that tasks to be solved simultaneously were chosen properly since the features extracted from the shared backbone for tool segmentation also proved worthwhile for event detection since the two tasks are related. In fact, in most cases, active bleeding occurs at the time when one of the tools closes or interacts with anatomical structures. Contrarily, there is a reduction in accumulated bleeding when the aspirator starts to extract it.

However, it was noted that this mutual benefit was lost when the number of epochs increased. In this situation, there was an improvement in accuracy relative to semantic segmentation, but the event detection task degenerated.

Certainly, the results achieved are only a preliminary version and future studies are needed to improve the system in several aspects. Future work will proceed in four different directions:

1. Neural Network Architecture. The architecture of the network should be extended to consider temporal information extracted from sequential images. This improvement will likely enhance the branch related to event detection in terms of accuracy and reliability. In this research context, for example, it may be easier to distinguish passive bleeding, namely, blood residue on tissues, and active bleeding. Moreover, the semantic segmentation task should distinguish tools with different labels to improve the prediction, while the detection task should be extended by adding the localization of the origin of bleeding, which may provide remarkable clinical advantages when the human eye cannot instantly catch it [256].
2. Dataset. An improved dataset in terms of numerosity and variance could be beneficial to increase the accuracy of prediction. Furthermore, the images that do not contain any structure of interest (external view of the operating room, images inside the trocar) in a limited-size dataset might improve the knowledge of the network about the studied environment [256].
3. Testing. The model should be evaluated in the operating room to determine its practical limitations in the setting of a real-time application and then enhanced accordingly by adding new features.
4. Research field. Extending the algorithm into different domains would be of interest, to perform further analysis both in terms of actor segmentation and tracking and from the point of view of event detection and classification.

3.2 Assisted 3D Printing

In the previous framework, we discussed the full framework applied to surgery in three different steps. Here we present the extension of the approach to the industrial environment to demonstrate the scalability of this approach. Part of this work has already been published in [273].

Introduction

In recent decades, technology has helped several procedures to improve massively; in particular, significant progress has been achieved with Deep Learning (DL) [3] paradigms. Recently, DL has acquired a fundamental role in various environments, and many different remarkable applications have been implemented, from the

medical domain [110,111,274] to natural language processing [275] and even gaming [276]. The same enhancement was brought in by Augmented Reality (AR) in a wide range of fields: tourism [277], education [278], surgery [188], manufacturing [279], and others.

The application of DL and AR has recently shown several advantages in the maintenance area as well. Maintenance refers to all actions that aim to restore any functionality of a product in its life cycle. The actions that can be performed to restore product functionality can be technical, administrative, and managerial [280].

Studies on DL and AR used independently in maintenance showed promising results to improve human performance in performing technical maintenance tasks, improving the administration of maintenance operations, and supporting maintenance managerial decision-making. Modern production machines are becoming increasingly complex, often integrating advanced components that require a high level of expertise in the skills necessary for their maintenance. In order to allow technicians to work on equipment they are not familiar with, it has been assessed in the literature [14] that DL and AR systems allow faster problem diagnosis, which in turn may lead to faster repair tasks. The augmentation is especially useful for quickly identifying the involved components as well as to present the professional with additional information about diagrams or protocols to be followed. Moreover, more repairing tasks can be performed by differently skilled technicians with the same expected level of quality, which in turn lessens the need for a single expert individual, that has not to travel from location to location to perform maintenance.

Unfortunately, in this field, joint utilization of DL and AR is still underrated due to the problematic compromise between working with 3D data and the efficiency needed for real-time elaboration.

In this study, we moved the first step in this context, as we aim to demonstrate how to apply DL to retrieve the position and the rotation data of the machine in need of maintenance from live video frames only. With those data, it will be possible to implement an AR system to give guidance to non-expert users during a specific maintenance operation.

As a case study to test our framework, we chose to use FDM 3D printers. This choice was due to two main reasons: firstly, they are ubiquitous and relatively cheap printers, primarily used by non-expert users that our system could facilitate; secondly, as in our research group we work with additive manufacturing, this software could help new members when facing such problems.

The presented framework leverages two different Convolutional Neural Networks (CNN) to determine the position and rotation data of the machine to be maintained. The two nets work on frames from a live video stream, like the one that may be obtained using any standard smartphone. The first neural network, a YOLOv3 [143] architecture, is applied to obtain the object localization inside the image. Once the object has been found, the rest of the image is cropped, and the resulting ROI (Region Of Interest) is then passed to a second CNN, called RotationCNN, trained on a synthetic dataset generated with Blender [281] to predict the rotation of the printer on the three cartesian axes.

We tested YOLOv3 with AP (Average Precision) value, a metric that defines the quality of an object detector, and the RotationCNN with values of Geodesic Distance, a positive amount corresponding to the length of the geodesic arc connecting the two rotations expressed in quaternions.

With the information of position ad scale retrieved from the YOLOv3's output and the ones of rotations predicted by the RotationCNN, an AR application can be implemented to project the 3D model directly onto its real-world counterpart. In section 5, we propose an example of such an application.

As a case study, we chose a specific FDM 3D Printer, the Prusa i3 MKS model. We implemented a particular maintenance activity, such as the replacement of filament, which is divided into several steps explained in detail further on. We selected this procedure as it is one of the most frequently handled. Nevertheless, this method works with any procedure and any FDM 3D Printer (or generic object as well) of which the 3D model is available. We then asked five non-expert users to perform the procedure with the help of the CAD system and five more without the CAD system but instead using the official Prusa i3 MKS's instruction manual to demonstrate the validity of our system.

Methods

The general method of our framework is detailed in Figure 44 Full pipeline of the proposed approach. The frame is passed to an object detection algorithm which returns the cropped area related to the detected 3D printer and the position and scale information. Its output is passed to a CNN, which predicts the rotation values, according to the coordinates system shown in the figure. This information is then combined to correctly overlay the 3D model to the 2D stream with specific instructions and highlighted components. However, for clarity, in this example is shown the whole 3D model instead of a specific highlighted part The original RGB image is firstly passed to an object detection algorithm, in our case YOLOv3, which

returns the ROI (Region Of Interest) related to the object detected. From this bounding box, we can obtain the values of scale and position of the 3D model. The cropped area is then passed to the RotationCNN, which returns the rotations values along the X, Y, and Z axes. This information is combined to overlay the 3D model on the 2D stream. All these steps will be discussed broadly in the following sub-sections.

Object detection

The first step of our approach aimed to locate the 3D Printer object in the video stream. The reason is two folds:

1. allow the RotationCNN to concentrate solely on the printer without getting confused by the noisy background;
2. retrieve the positions and scale values from the bounding box vertexes' coordinates. In particular, the center of the bounding box is used as an anchor point to attach the 3D model, and the scaling is computed comparing with a specific ratio of the width of the bounding box and the width of the 3D model.

As the application aimed to be real-time, we chose to use YOLO (You Only Look Once) algorithm [282], in particular, YOLOv3 [143], which, compared to the latest object detection framework such as Faster-RCNN [21], is less precise in some aspects, for example, it struggles with small objects within the image, but an order of magnitude faster. Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales, with high scoring regions of the image considered as detections. YOLO uses a totally different approach with just a single neural network that divides the image into regions and predicts bounding boxes (weighted by the predicted probabilities) and probabilities for each region.

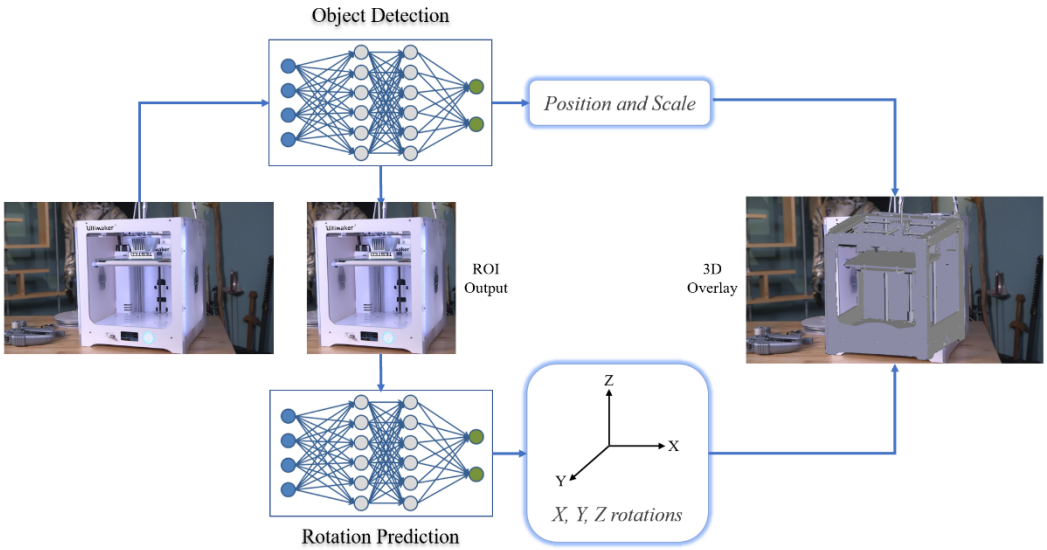


Figure 44 Full pipeline of the proposed approach. The frame is passed to an object detection algorithm which returns the cropped area related to the detected 3D printer and the position and scale information. Its output is passed to a CNN, which predicts the rotation values, according to the coordinates system shown in the figure. This information is then combined to correctly overlay the 3D model to the 2D stream with specific instructions and highlighted components. However, for clarity, in this example is shown the whole 3D model instead of a specific highlighted part

Rotation CNN

For the RotationCNN, which takes as input the area around the bounding box returned by YOLOv3, we used a ResNet50 [28] with three different branches for *X*, *Y,* and *Z* axes rotation. We choose ResNet50 after comparing it with two state-of-the-art model, InceptionV3 [14] and ViT-B16 [142], in terms of accuracy, geodesic distance and the number of iterations per seconds. Each branch has the same structure, which contains: a Dense layer with 4096 neurons and a ReLU (Rectified Linear Unit) activation function, a Batch Normalization layer [283], a Dropout layer [284] with a random parameter of 0.5, and a Dense output layer, with several neurons equal in number to the specific axis' range of rotations and a Softmax activation function. We solved the X, Y and Z axes rotation value estimation as a classification problem. We subdivided the set of possible rotation values along an axis according to the possible configurations of rotations that the 3D printer can possibly assume during this specific procedure. We considered 40 classes for *X*-axis rotation (from −10° to 30°), 60 for *Y*-axis rotation (from −30° to 30°), and 120 for *Z*-axis rotation (from -60° to 60°). Therefore, the neuron with the highest probability according to the Softmax activation function will fire and produce the corresponding rotation value as output. The architecture of the model

is shown in Figure 45 Architecture of the RotationCNN. The network takes as input the RGB images, which are passed through the first layers of ResNet50. The last layer was substituted with three branches composed of a Dense layer, a Batch Normalization layer, a Dropout layer, and a final Dense layer with *n* neurons, where *n* is the range of the rotation along the specific axis

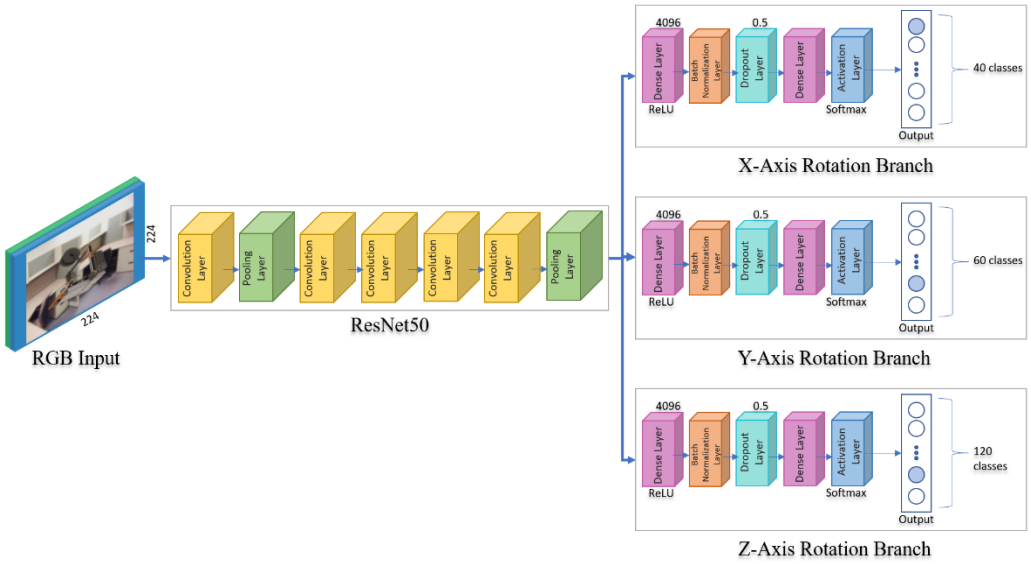


Figure 45 Architecture of the RotationCNN. The network takes as input the RGB images, which are passed through the first layers of ResNet50. The last layer was substituted with three branches composed of a Dense layer, a Batch Normalization layer, a Dropout layer, and a final Dense layer with *n* neurons, where *n* is the range of the rotation along the specific axis

Datasets

Three datasets have been used in this work. For training YOLOv3, we collected and tagged with bounding boxes 545 images showing the FDM 3D printers in different positions and rotations, of which 20% were kept away for testing. Each image was then resized to 461x461 pixels, and augmentation was applied with random horizontal flip, random zoom crop (0 to 45%), and random rotation (-15° to 15°), resulting in a total of 1653 images. To train the RotationCNN, we artificially build a synthetic dataset through Blender to make the rotation estimation easier by using as inputs 17 3D models of different FDM printers, and 36.500 real backgrounds taken from SUN Database [285]. According to meaningful rotation values of the FDM 3D printers, we considered the following ranges in degrees for the three rotation axes: [−10°,30°] for *X*-axis, [−30°, 30°] for *Y*-axis, and [−15°, 15°] for *Z*-axis. We generated a render for each combination of *X* and *Z* rotation values and 1/3 of *Y* rotation values and randomly changed lighting conditions and the scene's

background. With this process, we obtained a synthetic dataset of 94.220 images for training and 5000 for testing. Finally, we also created a dataset of 235 real-images of FDM 3D printers tagged with values of rotations, to fine-tune the network with 150 images and test the overall performances with the remaining 85 images.

Training, Metrics, and Framework

YOLOv3 was trained for 500 epochs with the first 49 layers frozen and a batch size of 32 and 100 additional epochs with all layers un-frozen and a batch size of 16. The metric used to evaluate the object detection is the Average Precision (AP) criterium defined in the PASCAL VOC 2012 [286] competition. First, the neural net detection results were sorted by decreasing confidence and were assigned to ground-truth objects. We had a match when the IoU (Intersection over Union) was more significant than a certain threshold. In the PASCAL VOC criterium, the IoU threshold was set to 0.5. In this work, as we were looking for a higher precision to overlap the 3D model, we also presented the values of AP using 0.6 and 0.7 thresholds. Using this approach, we calculated the precision/recall curve. Then, we computed a version of the measured precision/recall curve with precision monotonically decreasing and calculated the AP as the area under this curve by numerical integration.

The RotationCNN was trained with the synthetic dataset for 20 epochs with a batch size of 32 and Adam optimizer with a learning rate of 0.0001 and fine-tuned with real images for 100 epochs with the same batch size but Adam optimizer with a learning rate of 0.000001 and with all the layers except the three final branches frozen. The difference between the predicted and the actual rotations was computed by converting the values of the rotations in quaternions and then calculating the geodesic distance between two quaternion coordinates *q1* and *q2*. To get a distance between two unit quaternions, you have to rotate both of them such that one of them becomes the identity element. To do this for our pair *q1* and *q2*, we simply multiplied *q1* by *q2*'s inverse from the left

$$Q = (inverse(q2) * q1)$$

and normalize the obtained quaternion Q through L2 normalization:

$$geoDist = L2(Q) = \sqrt{Q \cdot Q}$$

The metric is a positive amount corresponding to the length of the geodesic arc connecting *q1* to *q2*. We choose the geodesic distance as is the most common metric used in literature to evaluate the difference between two angles. Finally, to evaluate

the performance of the different networks, we used the number of iterations per seconds, defined as:

$$it/s = \frac{n}{sec}$$

where n is the number of iterations, and *sec* is the time unit. In this case, one iteration consists of predicting the rotation's angles given an input image. As the frame rate metric depends on different aspects, such as the complexity of the mesh, the rendering engine used, the hardware specifics, the particular implementation of the pipeline, etc., these factors can determine a strong fluctuation of the metric, for this reason we preferred to opt for a metric independent of these parameters. We empirically noticed that our frame rate was acceptable if we kept the *it/s* greater than 5; this evaluation is not indicative but sufficient for us to obtain a real-time validation. We used Keras [210], an open-source neural-network library written in Python, running on top of TensorFlow and, on Windows 10 Pro with NVIDIA Quadro P4200.

Results

Object Detection

After the training phase, YOLO obtained a test loss of 11.353. We then computed the AP with IoU thresholds of 0.5, 0.6, and 0.7, shown in light blue as the area under the curve of the precision/recall curve in Figure 46 Precision/recall curve with IoU thresholds of 0.5 (a), 0.6 (b) and 0.7(c). The AP is shown in light blue as the area under the curve (a), (b), and (c), respectively. These values of AP were 100% with 0.5 as IoU threshold, 95.68% with 0.6 as IoU threshold, and 83.27% with 0.7 as IoU threshold.

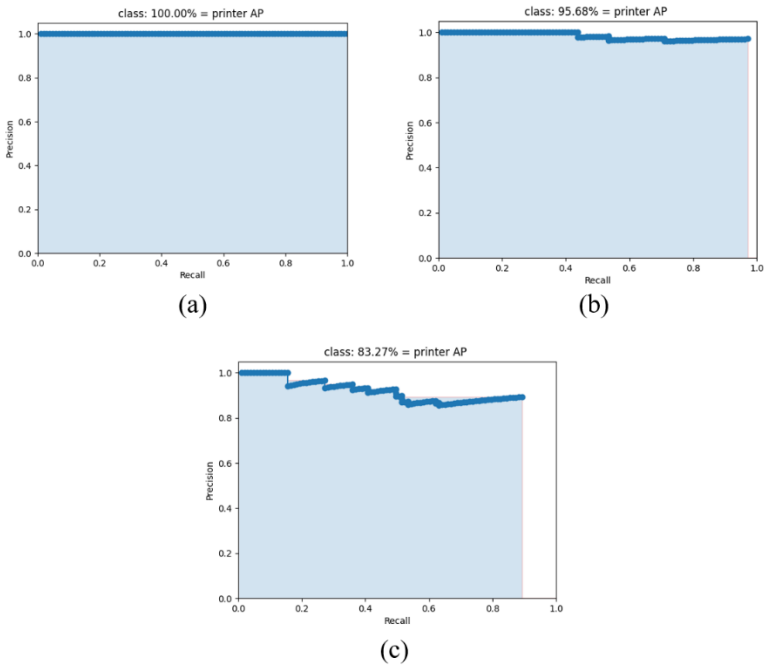


Figure 46 Precision/recall curve with IoU thresholds of 0.5 (a), 0.6 (b) and 0.7(c). The AP is shown in light blue as the area under the curve

RotationCNN

The results of the comparison between different networks are showed in Table 22.

Table 22 Values of mean accuracies together geodesic distance. All values are shown with related standard deviation. The accuracies were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: exact prediction, with an error in the range [-5, +5] and in the range [-10, +10], while the geodesic distance is computed as the positive amount corresponding to the length of the geodesic arc connecting the two rotation's angles. It/s are iterations per second, where one iteration is considered as a forward pass in the full pipeline

Architecture	Accuracies	Accuracies [-5, +5]	Accuracies [-10, +10]	Geodesic Distance	*it/s*
ResNet50	0.188 (σ=0.044)	0.443 (σ=0.077)	0.619 (σ=0.130)	0.250 (σ=0.210)	7
InceptionV3	0.158 (σ=0.043)	0.343 (σ=0.098)	0.531 (σ=0.160)	0.299 (σ=0.206)	6
ViT-B16	0.180 (σ=0.049)	0.402 (σ=0.080)	0.601 (σ=0.123)	0.255 (σ=0.203)	2

In addition, in Table 23 are showed the extensive results for the most performing network, ResNet50, chosen as the best compromise between accuracy, geodesic distance and number of iterations per seconds.

Table 23 Values of accuracies for each axes together with mean accuracy and geodesic distance. All values are shown with related standard deviation for tests performed with a synthetic dataset, a real dataset, and the same real dataset after fine-tuning the network. The accuracies for X, Y, and Z, were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: exact prediction, with an error in the range [-5, +5] and in the range [-10, +10], while the geodesic distance is computed as the positive amount corresponding to the length of the geodesic arc connecting the two rotation's angles

	Accuracies			**Accuracies [-5, +5]**			**Accuracies [-10, +10]**			**Geodesic Distance**	**# of images**
Axes	**X**	**Y**	**Z**	**X**	**Y**	**Z**	**X**	**Y**	**Z**		
Synthetic Dataset	0.644	0.961	0.950	0.998	0.999	0.998	1	1	1	0.0038 (σ=0.005)	5000
	0.852 (σ=0.147)			0.999 (σ=0.001)			1 (σ=0)				
Real Dataset	0.000	0.012	0.007	0.060	0.151	0.125	0.164	0.290	0.239	0.454 (σ=0.217)	85
	0.007 (σ=0.005)			0.113 (σ=0.038)			0.231 (σ=0.052)				
Real Dataset (After Fine Tuning)	0.130	0.237	0.196	0.334	0.502	0.491	0.435	0.713	0.707	0.250 (σ=0.210)	85
	0.188 (σ=0.044)			0.443 (σ=0.077)			0.619 (σ=0.130)				

We first tested the network with 5000 synthetic images generated with Blender. The accuracies for *X*, *Y,* and *Z*, were computed as the number of correct predictions over the total number of samples, using different acceptable ranges: exact prediction, with an error in the range [-5, +5] and with an error in the range [-10, +10**],** were 0.852 (σ=0.147), 0.999 (σ=0.001) and 1 (σ=0) respectively. We also computed the geodesic distance, which resulted in an average of 0.0038 (σ=0.005).

We then test this same network with 85 real images tagged manually. The accuracies were 0.007 (σ=0.005), 0.113 (σ=0.038) and 0.231 (σ=0.052) and the geodesic distance 0.454 (σ=0.217).

We finally fine-tuned the network with 150 images and tested with the same 85 images as before, resulting in an accuracy of 0.188 (σ=0.044), 0.443 (σ=0.077), and 0.619 (σ=0.130) respectively and a geodesic distance of 0.250 (σ=0.210).

Case Study

To validate our method, we chose a specific FDM 3D Printer, the Prusa i3 MKS model, and implemented a specific maintenance action, such as the replacement of filament, as it is one of the most frequent to be handled. Nevertheless, this system works for any procedure and any FDM 3D Printer (or generic object as well) of which the 3D model is available.

Procedure

After the extraction of the bounding box and the rotations value, the 3D model is attached at the center of the bounding box with the predicted rotations. The tool is then used to guide the user in the filament substitution. The phases, also underlined in Figure 47 In the specific operation of filament replacement four components of the Prusa i3 MKS are involved: the Screen and the Button for steps 1, 2, 3, 4, and 8, the Filament for steps 5 and 7, the Coil for step 6 and the Extruder for steps 9 and 10. These components are highlighted during the corresponding step, are:

1) Press the **button** and search in the menu screen “Unload the **filament**”
2) Press the **button** and specify the material to unload
3) Press the **button** and wait until the acoustic signal
4) Press the **button** to eject the **filament**
5) Pull the **filament** upwards
6) Replace the **coil**
7) Insert the new **filament** into the **extruder**’s **filament** hole
8) Search in the menu **screen** for “Load the **filament**”
9) Press the **button** and check if the extruded **filament** has the correct color
10) If *Yes*, clean the **extruder**, if *No*, repeat step 9

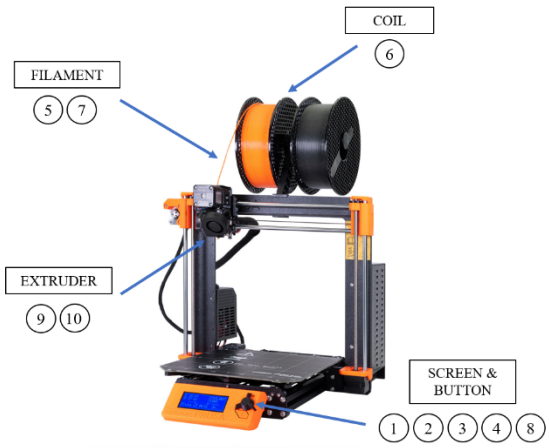


Figure 47 In the specific operation of filament replacement four components of the Prusa i3 MKS are involved: the Screen and the Button for steps 1, 2, 3, 4, and 8, the Filament for steps 5 and 7, the Coil for step 6 and the Extruder for steps 9 and 10. These components are highlighted during the corresponding step

Evaluation

Finally, we asked five non-expert users to perform the procedure with the help of the CAD System and five more without the CAD system but using the official Prusa i3 MKS's instructions manual. This method of validation is more accurate, because if we asked the same ten people to perform the operation with and without the CAD help the second time they would be facilitated by the fact that they have already done the operation once. Our system applies instead to users who have never performed the operation. The video stream with the overlay model and the textual information was provided on the PC screen, and the user has simply to press a generic button when he/she finished a step. In Figure 48 The ten steps of the filament substitution procedure. In each step, the specific component involved in the current precision was highlighted in orange, while the other components were kept in grey. The text instructions were the same defined above are shown the ten different augmented steps.

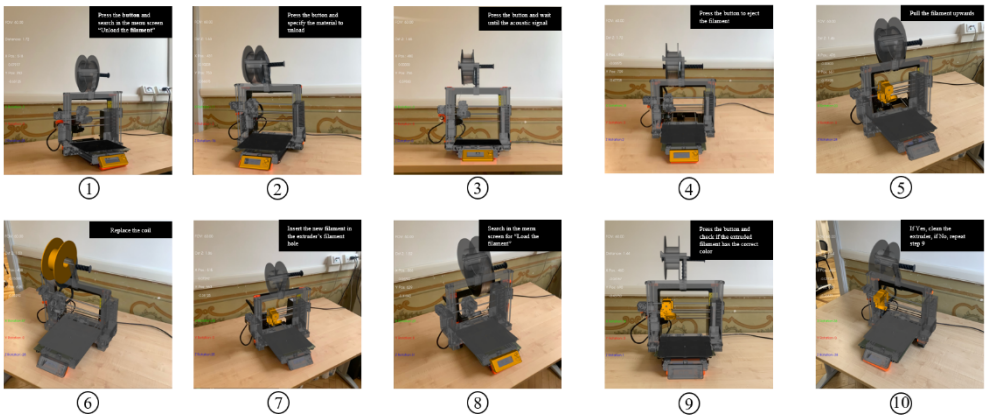


Figure 48 The ten steps of the filament substitution procedure. In each step, the specific component involved in the current precision was highlighted in orange, while the other components were kept in grey. The text instructions were the same defined above

The specific component involved in the current precision was highlighted in orange, while the other components were kept in grey. The text instructions were the same as defined above. To evaluate the quality of the intervention, we chose two parameters suggested by an expert in 3D printing in our research group: the time spent and the number of actions to perform the whole operation. An action is defined as a single step performed. With these two metrics we could evaluate both the speed and the precision (i.e., how many times a step had to be repeated). An expert user performs the whole procedure in the ten actions defined above with an average time of approximately 2 *minutes*. The five users performing the operations consulting the instructions manual obtained a mean speed of 9.61 (σ=1.53), and the mean number of actions was 12.60 (σ=2.15), while the five users helped by the CAD system obtained a mean speed of 5.30 (σ=1.30) and the mean number of action was 11.00 (σ=0.89). Results are resumed in Table 24.

Table 24 Values of speed (in minutes) and a number of actions performed by the ten non-expert users with the help of the user manual or with the help of our CAD system. In the two colored columns are showed the means and standard deviations for each distribution.

	User Manual						**CAD system**					
Users	**1**	**2**	**3**	**4**	**5**	**Mean**	**6**	**7**	**8**	**9**	**10**	**Mean**
Speed (*minutes*)	9.08	10.11	7.41	9.33	12.12	9.61 (σ=1.53)	7.21	4.32	5.23	3.56	6.20	5.30 (σ=1.30)
Number of actions	11	14	12	10	16	12.60 (σ=2.15)	12	10	11	10	12	11.00 (σ=0.89)

Discussion

In this work, we proposed a framework to assist non expert users in maintenance procedures of FDM 3D printers, through an AR overlay of the printer's 3D model based on DL algorithms. After a literature review, we chose to implement a two steps approach. The first phase involved an object detection algorithm, specifically YOLOv3, to locate the area related to the generic FDM 3D printer and obtain the value of the 3D model position and scale. This network was trained using 1653 augmented images together with their bounding boxes. In the second phase, we passed the output of YOLOv3 to a RotationCNN, which predicted the rotations values along the three axes, *X*, *Y,* and *Z*. This second network was trained with 94.220 synthetic images produced with Blender and fine-tuned with 150 real images tagged manually. The performances of these two networks are discussed in Table 23 and section 4. YOLOv3 detected 100% of the FDM 3D printer with an IoU threshold of 0.5, the official threshold used in the PASCAL VOC challenge, and obtained good results even if we increased the value of IoU. The most critical network was the RotationCNN, as the prediction of rotations is far more complex than the object detection task. We tested three different network, ResNet50, InceptionV3 and ViT-B16, after selecting ResNet50 as the most performing one. In fact, ResNet50 achieved similar accuracies and geodesic distances as ViT-B16 while performing three times as many operations per second. We used two metrics: the accuracy, considering different acceptable ranges: exact prediction, with an error in the range [-5, +5] and in the range [-10, +10**],** and the geodesic distance. Testing with 5000 synthetic images, we obtained values of accuracies close to 1 and a very low geodesic distance of 0.0038 (σ=0.005), showing that the network actually learned; the main question was if he was able to generalize the results with authentic images. Without fine-tuning, the performances of the network were very poor, with accuracies close to 0 and a geodesic distance of 0.454 (σ=0.217). After fine-tuning the network with 150 real images tagged manually, we obtained the following results: the three values of accuracy were 0.188 (σ=0.044), 0.443 (σ=0.077), and 0.619 (σ=0.130) respectively and the geodesic distance was 0.250 (σ=0.210). These results are good enough to implement our methodology.

To test our system, we also presented a case study with a specific FDM 3D printer, the Prusa i3 MKS. We asked five non-expert users to perform the procedure with the help of the CAD System and five more without the CAD system but using the official Prusa i3 MKS's instructions manual and evaluate the performances with speed and number of operations to complete the whole procedure. The mean improvement given by using our tool was 4 *minutes* and 31 *seconds* in speed and

1.6 in number of operations. The concept of evaluating AR-assisted maintenance was inspired by the realization that measuring user competence also serves as an indirect indicator for measuring the performance of the positioning method in actual use. Naturally, employing the support tool would not increase the users' performance if the position is not correctly obtained, as shown in Table 24.

Even if we showed our system's performance, there are still several limits to overcome. First of all, the RotationCNN performances are acceptable, but it still struggles in detecting the precise value of the rotation. Secondly, we provided the users with a GUI directly on the PC screen: we think the performances could vastly improve with a mobile application. Thirdly, the operation itself was quite simple to perform. Thus the practical improvement of the use of our tool may seem low.

In future works, we plan to compare the performances of our methodology to other similar, implementing a more complex application for mobile smartphones. Focusing on different models of FDM 3D printers, the improvement will include more complex maintenance operations.

Chapter 4

Discussion and Conclusions

In this Ph.D. thesis, we discussed the outcomes of our DL and AR-based human-machine interaction research in the medical and industrial sectors. In particular, we proposed a framework that is divided into two layers and could be applied to different contexts. The full framework takes care of reading a 2D video stream, extracting information, and overlaying 3D data to support the operator.

First, in Chapter 1, we provided a literature assessment of approaches for computer-assisted surgery and troubleshooting. This overview provided us with the fundamentals needed to identify the benefits of current strategies and combine them into a brand-new, innovative approach. In particular, we understood that modularity is a crucial aspect of this type of strategy in order to be able to modify particular components to new technologies or issues. Therefore, we chose to divide the framework into two layers: the first layer is based on DL and handles image processing using various methods, including segmentation and detection. The second uses AR to add more information to the video stream.

Then, in Chapter 2, we covered two projects that were only concerned with the first layer, which gave us a general grasp of DL in computer vision.

In the first project, we developed a classifier of femur fracture X-Rays together with the CTO of Turin, the KTH University of Stockholm and UTC Sorbonne University. The work was split into three steps: we first developed a cascade of CNN that adheres to the AO classification's hierarchical structure. We then overcome the drawbacks of this approach by substituting the cascade with a Vision Transformer, trained with a larger dataset, and implementing a fully automated pipeline. Finally, we narrowed our attention to hierarchies in general and provided a loss function and regularization, suitable for this kind of issue. The final classifier was able to correctly classify 83% of the test images. Additionally, we conducted a testing phase with 11 specialists and found that evaluating the type of fractures, with or without the use of the CAD system, improved by an average of 29%. The

application of this instrument in real-world situations may increase the accuracy of the specialist and, consequently, the health of the patients.

In the second project, we worked with the Polytechnic University of Milan to implement a classifier of six fundamental human emotions: anger, disgust, fear, happiness, sadness, and surprise. The approach is based on a multimodal network that combines 2D and 3D data on human emotion to produce a unique prediction. With this approach, we achieved a recognition accuracy of 87%. One potential real-world application of this work that we are now implementing is in the context of museums. Here, by reading the emotions of visitors and processing them through this system, the organizer may be able to recommend an ad-hoc path for each user.

In Chapter 3, we moved to the second layer and presented two possible applications of our full framework.

The first, which pertains to the medical field, is broken down into three more works. In the first two, we discussed the issue of aiding the surgeon in difficult surgeries by superimposing a 3D model over a 2D video stream. We concentrated on the catheter, the prostate, and the kidney as our three main targets. With the catheter, we were able to achieve a 99% accuracy. On the other hand, because organs are composed of soft tissue and are therefore deformable, we only got an accuracy of about 60% and used an optical flow method to increase this value. In the third, we concentrated on the idea of anticipating surgical occurrences, such as blood accumulation. By training a multi-output network for segmentation and even prediction, we obtained an accuracy of 90% to predict the potential accumulation of blood while confirming the previous performances in segmentation. The entire method is currently being tested during in-vivo surgery at the San Luigi Hospital of Orbassano.

The second extended a similar strategy for computer-assisted troubleshooting, where we set in place a framework to help novice users throughout a 3D printer repair phase. The network was able to detect the printer with an average precision of 95% and overlay the assistance 3D model with a geodesic distance of 0.25. We also presented a case study of a particular maintenance procedure and examined the proficiency of non-expert users in carrying out the procedure. The results showed a mean improvement of 4 minutes, 31 seconds in speed, and 1.6 in the number of operations required for each stage. The tool could be applied to any tasks or industries where a 3D model of the object being maintained is present.

The possible advantages of this framework are countless, starting with the two just presented. Indeed, this framework could be applied to any situation where a 2D video is present, and you want to help the user in a specific operation by producing an augmented video.

Nevertheless, there are still different limits to overcome. Concerning the first layer, data availability is the primary obstacle. Indeed, especially in the medical domain, the number of images is still very limited. To solve this issue, we relied on 3D render images and fine-tuned the models on a small set of real images. This, unfortunately, led to a decrease in accuracy. One solution to this would be to augment the dataset using Generative Adversarial Networks to produce new and reliable samples. The problem with GANs is that they might not perform well with classes that have a small number of samples because they will be less able to generalize and produce new images as there will be fewer images available for training. For this reason, one option could be to produce new generic samples, train the architecture with the entire dataset, and ask the specialists to categorize the generated images.

In addition, the key drawback of the second layer is that all of the studies were performed on local PCs without the use of mobile applications, which would have greatly improved the usability of this strategy.

These two ideas would be incorporated into future works.

References

1. Ren F, Bao Y. A Review on Human-Computer Interaction and Intelligent Robots. Int J Info Tech Dec Mak. 2020 Jan;19(01):5–47.

2. Artificial Intelligence: A Modern Approach, 4th US ed. [Internet]. [cited 2022 Nov 8]. Available from: http://aima.cs.berkeley.edu/

3. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 May;521(7553):436–44.

4. Aggarwal CC. Neural Networks and Deep Learning: A Textbook [Internet]. Cham: Springer International Publishing; 2018 [cited 2021 Jul 17]. Available from: http://link.springer.com/10.1007/978-3-319-94463-0

5. Tanzi L, Vezzetti E, Moreno R, Moos S. X-Ray Bone Fracture Classification Using Deep Learning: A Baseline for Designing a Reliable Approach. Applied Sciences. 2020 Feb 22;10(4):1507.

6. Tanzi L, Piazzolla P, Vezzetti E. Intraoperative surgery room management: A deep learning perspective. The International Journal of Medical Robotics and Computer Assisted Surgery. 2020;16(5):e2136.

7. Amparore D, Checcucci E, Gribaudo M, Piazzolla P, Porpiglia F, Vezzetti E. Non-linear-Optimization Using SQP for 3D Deformable Prostate Model Pose Estimation in Minimally Invasive Surgery. In: Arai K, Kapoor S, editors. Advances in Computer Vision [Internet]. Cham: Springer International Publishing; 2020 [cited 2020 Feb 21]. p. 477–96. Available from: http://link.springer.com/10.1007/978-3-030-17795-9_35

8. Olivetti EC, Nicotera S, Marcolin F, Vezzetti E, Sotong JPA, Zavattero E, Ramieri G. 3D Soft-Tissue Prediction Methodologies for Orthognathic Surgery—A Literature Review. Applied Sciences. 2019 Oct 26;9(21):4550.

9. International Osteoporosis Foundation. Broken bones, broken lives: a roadmap to solve the fragility fracture crisis in Europe [Internet]. 2018 [cited 2019 Sep 20]. Available from: https://www.iofbonehealth.org/broken-bones-broken-lives

10. Krupinski EA, Berbaum KS, Caldwell RT, Schartz KM, Kim J. Long Radiology Workdays Reduce Detection and Accommodation Accuracy. Journal of the American College of Radiology. 2010 Sep;7(9):698–704.

11. Hallas P, Ellingsen T. Errors in fracture diagnoses in the emergency department – characteristics of patients and diurnal variation. BMC Emerg Med. 2006 Dec;6(1):4.

12. Dimililer K. IBFDS: Intelligent bone fracture detection system. Procedia Computer Science. 2017;120:260–7.

13. Kim DH, MacKinnon T. Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. Clinical Radiology. 2018 May;73(5):439–45.

14. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. Las Vegas, NV, USA: IEEE; 2016 [cited 2019 Nov 25]. p. 2818–26. Available from: http://ieeexplore.ieee.org/document/7780677/

15. Fei-Fei L, Deng J, Li K. ImageNet: Constructing a large-scale image database. Journal of Vision. 2010 Mar 22;9(8):1037–1037.

16. Lindsey R, Daluiski A, Chopra S, Lachapelle A, Mozer M, Sicular S, Hanel D, Gardner M, Gupta A, Hotchkiss R, Potter H. Deep neural network improves fracture detection by clinicians. Proc Natl Acad Sci USA. 2018 Nov 6;115(45):11591–6.

17. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. MICCAI 2015. Springer International Publishing; 2015. p. 234–41.

18. Rajpurkar P, Irvin J, Bagul A, Ding D, Duan T, Mehta H, Yang B, Zhu K, Laird D, Ball RL, Langlotz C, Shpanskaya K, Lungren MP, Ng AY. MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. arXiv:171206957 [physics] [Internet]. 2018 May 22 [cited 2019 Nov 25]; Available from: http://arxiv.org/abs/1712.06957

19. Olczak J, Fahlberg N, Maki A, Razavian AS, Jilert A, Stark A, Sköldenberg O, Gordon M. Artificial intelligence for analyzing orthopedic trauma radiographs: Deep learning algorithms—are they on par with humans for diagnosing fractures? Acta Orthopaedica. 2017 Nov 2;88(6):581–6.

20. Yahalomi E, Chernofsky M, Werman M. Detection of Distal Radius Fractures Trained by a Small Set of X-Ray Images and Faster R-CNN. In: Arai K, Bhatia R, Kapoor S, editors. Intelligent Computing [Internet]. Cham: Springer International Publishing; 2019 [cited 2019 Nov 25]. p. 971–81. Available from: http://link.springer.com/10.1007/978-3-030-22871-2_69

21. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans Pattern Anal Mach Intell. 2017 Jun 1;39(6):1137–49.

22. Thurston M, MacKinnon T, Kim DH. Fracture detection with artificial intelligence: improved accuracy with region of interest focusing. In 2018.

23. Bradski G. The OpenCV Library. Dr Dobb's Journal of Software Tools; 2000.

24. Cheng CT, Ho TY, Lee TY, Chang CC, Chou CC, Chen CC, Chung IF, Liao CH. Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs. Eur Radiol. 2019 Oct;29(10):5469–77.

25. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: 2017 IEEE International Conference on Computer Vision (ICCV) [Internet]. Venice: IEEE; 2017 [cited 2019 Nov 25]. p. 618–26. Available from: http://ieeexplore.ieee.org/document/8237336/

26. Huang G, Liu Z, Maaten L van der, Weinberger KQ. Densely Connected Convolutional Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. Honolulu, HI: IEEE; 2017 [cited 2019 Nov 25]. p. 2261–9. Available from: http://ieeexplore.ieee.org/document/8099726/

27. Chung SW, Han SS, Lee JW, Oh KS, Kim NR, Yoon JP, Kim JY, Moon SH, Kwon J, Lee HJ, Noh YM, Kim Y. Automated detection and classification of the proximal humerus fracture by using deep learning algorithm. Acta Orthopaedica. 2018 Jul 4;89(4):468–73.

28. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. IEEE; 2016. p. 770–8. Available from: http://ieeexplore.ieee.org/document/7780459/

29. Jiménez-Sánchez A, Kazi A, Albarqouni S, Kirchhoff C, Biberthaler P, Navab N, Mateus D, Kirchhoff S. Towards an Interactive and Interpretable CAD System to Support Proximal Femur Fracture Classification. arXiv:190201338 [cs] [Internet]. 2019 Feb 4 [cited 2019 Nov 25]; Available from: http://arxiv.org/abs/1902.01338

30. Jin WJ, Dai LY, Cui YM, Zhou Q, Jiang LS, Lu H. Reliability of classification systems for intertrochanteric fractures of the proximal femur in experienced orthopaedic surgeons. Injury. 2005 Jul;36(7):858–61.

31. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative Adversarial Networks. arXiv:14062661

[cs, stat] [Internet]. 2014 Jun 10 [cited 2019 Nov 25]; Available from: http://arxiv.org/abs/1406.2661

32. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal. 2015;13:8–17.

33. Wachs JP. Gaze, Posture and Gesture Recognition to Minimize Focus Shifts for Intelligent Operating Rooms in a Collaborative Support System. INT J COMPUT COMMUN. 2010 Mar 1;5(1):106.

34. Hashimoto DA, Rosman G, Rus D, Meireles OR. Artificial Intelligence in Surgery: Promises and Perils. Annals of Surgery. 2018 Jul;268(1):70–6.

35. Yeung S, Rinaldo F, Jopling J, Liu B, Mehra R, Downing NL, Guo M, Bianconi GM, Alahi A, Lee J, Campbell B, Deru K, Beninati W, Fei-Fei L, Milstein A. A computer vision system for deep learning-based detection of patient mobilization activities in the ICU. npj Digit Med. 2019 Dec;2(1):11.

36. Cristancho SM, Vanstone M, Lingard L, LeBel ME, Ott M. When surgeons face intraoperative challenges: a naturalistic model of surgical decision making. The American Journal of Surgery. 2013 Feb;205(2):156–62.

37. Moher D, Liberati A, Tetzlaff J, Altman DG. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. International Journal of Surgery. 2010;8(5):336–41.

38. Martin JA, Regehr G, Reznick R, Macrae H, Murnaghan J, Hutchison C, Brown M. Objective structured assessment of technical skill (OSATS) for surgical residents. Br J Surg. 1997 Feb;84(2):273–8.

39. Gofton WT, Dudek NL, Wood TJ, Balaa F, Hamstra SJ. The Ottawa Surgical Competency Operating Room Evaluation (O-SCORE): A Tool to Assess Surgical Competence. Academic Medicine. 2012 Oct;87(10):1401–7.

40. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. Accurate and interpretable evaluation of surgical skills from kinematic data using fully convolutional neural networks. Int J CARS. 2019 Sep;14(9):1611–7.

41. Wang Z, Majewicz Fey A. Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery. Int J CARS. 2018 Dec;13(12):1959–70.

42. Gao Y, Vedula SS, Reiley CE, Ahmidi N, Varadarajan B, Lin HC, Tao L, Zappella L, Béjar B, Yuh DD, Chen CCG, Vidal R, Khudanpur S, Hager GD.

JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) : A Surgical Activity Dataset for Human Motion Modeling. In 2014.

43. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. arXiv:151204150 [cs] [Internet]. 2015 Dec 13 [cited 2019 Dec 18]; Available from: http://arxiv.org/abs/1512.04150

44. Jin A, Yeung S, Jopling J, Krause J, Azagury D, Milstein A, Fei-Fei L. Tool Detection and Operative Skill Assessment in Surgical Videos Using Region-Based Convolutional Neural Networks. arXiv:180208774 [cs] [Internet]. 2018 Jul 21 [cited 2020 Jan 21]; Available from: http://arxiv.org/abs/1802.08774

45. DiPietro R, Lea C, Malpani A, Ahmidi N, Vedula SS, Lee GI, Lee MR, Hager GD. Recognizing Surgical Activities with Recurrent Neural Networks. arXiv:160606329 [cs] [Internet]. 2016 Jun 22 [cited 2020 Jan 16]; Available from: http://arxiv.org/abs/1606.06329

46. Jin Y, Dou Q, Chen H, Yu L, Qin J, Fu CW, Heng PA. SV-RCNet: Workflow Recognition From Surgical Videos Using Recurrent Convolutional Network. IEEE Trans Med Imaging. 2018;37(5):1114–26.

47. Kitaguchi D, Takeshita N, Matsuzaki H, Takano H, Owada Y, Enomoto T, Oda T, Miura H, Yamanashi T, Watanabe M, Sato D, Sugomori Y, Hara S, Ito M. Real-time automatic surgical phase recognition in laparoscopic sigmoidectomy using the convolutional neural network-based deep learning approach. Surg Endosc [Internet]. 2019 Dec 3 [cited 2020 Feb 4]; Available from: http://link.springer.com/10.1007/s00464-019-07281-0

48. Nakawala H, Bianchi R, Pescatori LE, De Cobelli O, Ferrigno G, De Momi E. “Deep-Onto” network for surgical workflow and context recognition. Int J CARS. 2019 Apr;14(4):685–96.

49. Namazi B. Automatic Detection of Surgical Phases in Laparoscopic Videos. In 2018.

50. Nishio S, Hossain B, Nii M, Hiranaka T, Kobashi S. Workflow Recognition from Knee Surgical Videos: Role of Deep Neural Networks. ISASE. 2019;ISASE2019(0):1–5.

51. Zia A, Hung A, Essa I, Jarc A. Surgical Activity Recognition in Robot-Assisted Radical Prostatectomy using Deep Learning. arXiv:180600466 [cs] [Internet]. 2018 Jun 1 [cited 2020 Jan 28]; Available from: http://arxiv.org/abs/1806.00466

52. Wang S, Raju A, Huang J. Deep learning based multi-label classification for surgical tool presence detection in laparoscopic videos. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017). 2017. p. 620–3.

53. Twinanda AP, Shehata S, Mutter D, Marescaux J, de Mathelin M, Padoy N. EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. arXiv:160203012 [cs] [Internet]. 2016 May 23 [cited 2019 Dec 5]; Available from: http://arxiv.org/abs/1602.03012

54. Choi B, Jo K, Choi S, Choi J. Surgical-tools detection based on Convolutional Neural Network in laparoscopic robot-assisted surgery. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) [Internet]. Seogwipo: IEEE; 2017 [cited 2020 Jan 21]. p. 1756–9. Available from: https://ieeexplore.ieee.org/document/8037183/

55. Hossain M, Nishio S, Hiranaka T, Kobashi S. Real-time Surgical Tools Recognition in Total Knee Arthroplasty Using Deep Neural Networks. arXiv:180602031 [cs] [Internet]. 2018 Jun 6 [cited 2019 Dec 3]; Available from: http://arxiv.org/abs/1806.02031

56. Jo K, Choi Y, Choi J, Chung JW. Robust Real-Time Detection of Laparoscopic Instruments in Robot Surgery Using Convolutional Neural Networks with Motion Vector Prediction. Applied Sciences. 2019 Jul 18;9(14):2865.

57. Sarikaya D, Corso JJ, Guru KA. Detection and Localization of Robotic Tools in Robot-Assisted Surgery Videos Using Deep Neural Networks for Region Proposal and Detection. IEEE Trans Med Imaging. 2017 Jul;36(7):1542–9.

58. Nakazawa A, Harada K, Mitsuishi M, Jannin P. Real-time surgical needle detection using region-based convolutional neural networks. Int J CARS. 2020 Jan;15(1):41–7.

59. Nwoye CI, Mutter D, Marescaux J, Padoy N. Weakly supervised convolutional LSTM approach for tool tracking in laparoscopic videos. Int J CARS. 2019 Jun;14(6):1059–67.

60. Zhao Z, Chen Z, Voros S, Cheng X. Real-time tracking of surgical instruments based on spatio-temporal context and deep learning. Computer Assisted Surgery. 2019 Oct 1;24(sup1):20–9.

61. Attia M, Hossny M, Nahavandi S, Asadi H. Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder. In: 2017 IEEE International

Conference on Systems, Man, and Cybernetics (SMC) [Internet]. Banff, AB: IEEE; 2017 [cited 2020 Feb 3]. p. 3373–8. Available from: http://ieeexplore.ieee.org/document/8123151/

62. García-Peraza-Herrera LC, Li W, Gruijthuijsen C, Devreker A, Attilakos G, Deprest J, Vander Poorten E, Stoyanov D, Vercauteren T, Ourselin S. Real-Time Segmentation of Non-rigid Surgical Tools Based on Deep Learning and Tracking. In: Peters T, Yang GZ, Navab N, Mori K, Luo X, Reichl T, McLeod J, editors. Computer-Assisted and Robotic Endoscopy [Internet]. Cham: Springer International Publishing; 2017 [cited 2020 Jan 23]. p. 84–95. Available from: http://link.springer.com/10.1007/978-3-319-54057-3_8

63. Laina I, Rieke N, Rupprecht C, Vizcaíno JP, Eslami A, Tombari F, Navab N. Concurrent Segmentation and Localization for Tracking of Surgical Instruments. arXiv:170310701 [cs] [Internet]. 2017 Aug 1 [cited 2020 Jan 27]; Available from: http://arxiv.org/abs/1703.10701

64. Pakhomov D, Premachandran V, Allan M, Azizian M, Navab N. Deep Residual Learning for Instrument Segmentation in Robotic Surgery. In: Suk HI, Liu M, Yan P, Lian C, editors. Machine Learning in Medical Imaging [Internet]. Cham: Springer International Publishing; 2019 [cited 2020 Jan 27]. p. 566–73. Available from: http://link.springer.com/10.1007/978-3-030-32692-0_65

65. Shvets A, Rakhlin A, Kalinin AA, Iglovikov V. Automatic Instrument Segmentation in Robot-Assisted Surgery Using Deep Learning. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). 2018 Dec;624–8.

66. Bodenstedt S, Wagner M, Katić D, Mietkowski P, Mayer B, Kenngott H, Müller-Stich B, Dillmann R, Speidel S. Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. arXiv:170203684 [cs] [Internet]. 2017 Feb 13 [cited 2020 Jan 16]; Available from: http://arxiv.org/abs/1702.03684

67. Chen Y, Sun QL, Zhong K. Semi-supervised spatio-temporal CNN for recognition of surgical workflow. J Image Video Proc. 2018 Dec;2018(1):76.

68. Yu F, Silva Croso G, Kim TS, Song Z, Parker F, Hager GD, Reiter A, Vedula SS, Ali H, Sikder S. Assessment of Automated Identification of Phases in Videos of Cataract Surgery Using Machine Learning and Deep Learning Techniques. JAMA Netw Open. 2019 Apr 5;2(4):e191860.

69. Zisimopoulos O, Flouty E, Luengo I, Giataganas P, Nehme J, Chow A, Stoyanov D. DeepPhase: Surgical Phase Recognition in CATARACTS

Videos. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G, editors. Medical Image Computing and Computer Assisted Intervention – MICCAI 2018 [Internet]. Cham: Springer International Publishing; 2018 [cited 2020 Jan 28]. p. 265–72. Available from: http://link.springer.com/10.1007/978-3-030-00937-3_31

70. Hajj HA, Lamard M, Conze PH, Cochener B, Quellec G. Monitoring tool usage in surgery videos using boosted convolutional and recurrent neural networks. arXiv:171001559 [cs] [Internet]. 2018 May 6 [cited 2020 Mar 5]; Available from: http://arxiv.org/abs/1710.01559

71. Sahu M, Mukhopadhyay A, Szengel A, Zachow S. Addressing multi-label imbalance problem of surgical tool detection using CNN. Int J CARS. 2017 Jun;12(6):1013–20.

72. Issenhuth T, Srivastav V, Gangi A, Padoy N. Face detection in the operating room: comparison of state-of-the-art methods and a self-supervised approach. Int J CARS. 2019 Jun;14(6):1049–58.

73. Hansen L, Siebert M, Diesel J, Heinrich MP. Fusing information from multiple 2D depth cameras for 3D human pose estimation in the operating room. Int J CARS. 2019 Nov;14(11):1871–9.

74. Belagiannis V, Wang X, Shitrit HBB, Hashimoto K, Stauder R, Aoki Y, Kranzfelder M, Schneider A, Fua P, Ilic S, Feussner H, Navab N. Parsing human skeletons in an operating room. Machine Vision and Applications. 2016 Oct;27(7):1035–46.

75. Nishio S, Hossain M, Hossain B, Nii M, Hiranaka T, Kobashi S. Real-Time Orthopedic Surgery Procedure Recognition Method with Video Images from Smart Glasses Using Convolutional Neural Network. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) [Internet]. Miyazaki, Japan: IEEE; 2018 [cited 2020 Jan 22]. p. 379–84. Available from: https://ieeexplore.ieee.org/document/8616069/

76. Zhou T, Wachs JP. Spiking Neural Networks for early prediction in human–robot collaboration. The International Journal of Robotics Research. 2019 Dec;38(14):1619–43.

77. Aksamentov I, Twinanda AP, Mutter D, Marescaux J, Padoy N. Deep Neural Networks Predict Remaining Surgery Duration from Cholecystectomy Videos. In: Descoteaux M, Maier-Hein L, Franz A, Jannin P, Collins DL, Duchesne S, editors. Medical Image Computing and Computer-Assisted Intervention − MICCAI 2017. Cham: Springer International Publishing; 2017. p. 586–93.

78. Leibetseder A, Primus MJ, Petscharnig S, Schoeffmann K. Image-Based Smoke Detection in Laparoscopic Videos. In: Cardoso MJ, Arbel T, Luo X, Wesarg S, Reichl T, González Ballester MÁ, McLeod J, Drechsler K, Peters T, Erdt M, Mori K, Linguraru MG, Uhl A, Oyarzun Laura C, Shekhar R, editors. Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures [Internet]. Cham: Springer International Publishing; 2017 [cited 2020 Jan 27]. p. 70–87. Available from: http://link.springer.com/10.1007/978-3-319-67543-5_7

79. Lesnes-Cuisiniez E, Flores JZ, Radoux JP. Gesture ToolBox: Touchless Human-Machine Interface Using Deep Learning. In: Kern-Isberner G, Fürnkranz J, Thimm M, editors. KI 2017: Advances in Artificial Intelligence [Internet]. Cham: Springer International Publishing; 2017 [cited 2020 Feb 4]. p. 323–9. Available from: http://link.springer.com/10.1007/978-3-319-67190-1_27

80. Nasr-Esfahani E, Karimi N, Soroushmehr SMR, Jafari MH, Khorsandi MA, Samavi S, Najarian K. Hand Gesture Recognition for Contactless Device Control in Operating Rooms. Int J CARS. 2017 Jun;12(S1):1–286.

81. Scavazzin M, Valentini F, Radoux JP, Pereme F, Zegarra J. Conception of a touchless human machine interaction system for operating rooms using deep learning. In: Schelkens P, Ebrahimi T, Cristóbal G, editors. Optics, Photonics, and Digital Technologies for Imaging Applications V [Internet]. Strasbourg, France: SPIE; 2018 [cited 2020 Jan 23]. p. 80. Available from: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10679/2319141/Conception-of-a-touchless-human-machine-interaction-system-for-operating/10.1117/12.2319141.full

82. Li P, Hou X, Wei L, Song G, Duan X. Efficient and Low-Cost Deep-Learning Based Gaze Estimator for Surgical Robot Control. In: 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR) [Internet]. Kandima, Maldives: IEEE; 2018 [cited 2020 Jan 23]. p. 58–63. Available from: https://ieeexplore.ieee.org/document/8621810/

83. Flouty E, Zisimopoulos O, Stoyanov D. FaceOff: Anonymizing Videos in the Operating Rooms. 2018.

84. Sganga J, Eng D, Graetzel C, Camarillo D. OffsetNet: Deep Learning for Localization in the Lung using Rendered Images. arXiv:180905645 [cs] [Internet]. 2018 Sep 15 [cited 2020 Feb 3]; Available from: http://arxiv.org/abs/1809.05645

85. Bouget D, Benenson R, Omran M, Riffaud L, Schiele B, Jannin P. Detecting Surgical Tools by Modelling Local Appearance and Global Shape. IEEE Trans Med Imaging. 2015 Dec;34(12):2603–17.

86. Aviles AI, Alsaleh SM, Sobrevilla P, Casals A. Force-feedback sensory substitution using supervised recurrent learning for robotic-assisted surgery. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) [Internet]. Milan: IEEE; 2015 [cited 2020 Jan 17]. p. 1–4. Available from: http://ieeexplore.ieee.org/document/7318246/

87. Twinanda AP, Winata P, Gangi A, Mathelin M, Padoy N. Multi-stream deep architecture for surgical phase recognition on multi-view RGBD videos. In: Proc M2CAI Workshop MICCAI. 2016. p. 1–8.

88. Anh NX, Nataraja RM, Chauhan S. Towards near real-time assessment of surgical skills: A comparison of feature extraction techniques. Computer Methods and Programs in Biomedicine. 2020 Apr;187:105234.

89. Levin M, McKechnie T, Khalid S, Grantcharov TP, Goldenberg M. Automated Methods of Technical Skill Assessment in Surgery: A Systematic Review. Journal of Surgical Education. 2019 Nov;76(6):1629–39.

90. Jin Y, Li H, Dou Q, Chen H, Qin J, Fu CW, Heng PA. Multi-Task Recurrent Convolutional Network with Correlation Loss for Surgical Video Analysis. arXiv:190706099 [cs, eess] [Internet]. 2019 Jul 13 [cited 2020 Jan 27]; Available from: http://arxiv.org/abs/1907.06099

91. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. arXiv:150602640 [cs] [Internet]. 2016 May 9 [cited 2020 Mar 11]; Available from: http://arxiv.org/abs/1506.02640

92. Izhikevich EM. Polychronization: Computation with Spikes. Neural Computation. 2006 Feb;18(2):245–82.

93. Liu D, Jiang T. Deep Reinforcement Learning for Surgical Gesture Segmentation and Classification. In: Frangi AF, Schnabel JA, Davatzikos C, Alberola-López C, Fichtinger G, editors. Medical Image Computing and Computer Assisted Intervention – MICCAI 2018 [Internet]. Cham: Springer International Publishing; 2018 [cited 2020 Mar 4]. p. 247–55. (Lecture Notes in Computer Science; vol. 11073). Available from: http://link.springer.com/10.1007/978-3-030-00937-3_29

94. Palmarini R, Erkoyuncu JA, Roy R, Torabmostaedi H. A systematic review of augmented reality applications in maintenance. Robotics and Computer-Integrated Manufacturing. 2018 Feb 1;49:215–28.

95. Wójcicki T. Supporting the diagnostics and the maintenance of technical devices with augmented reality. Diagnostyka. 2017 Dec 20;15(1):43–7.

96. Regenbrecht H, Baratoff G, Wilke W. Augmented reality projects in the automotive and aerospace industries. IEEE Computer Graphics and Applications. 2005 Nov;25(6):48–56.

97. Webel S, Bockholt U, Engelke T, Gavish N, Olbrich M, Preusche C. An augmented reality training platform for assembly and maintenance skills. Robotics and Autonomous Systems. 2013 Apr 1;61(4):398–403.

98. Azuma RT. A Survey of Augmented Reality. Presence: Teleoper Virtual Environ. 1997 Aug;6(4):355–85.

99. Lee SG, Ma YS, Thimm GL, Verstraeten J. Product lifecycle management in aviation maintenance, repair and overhaul. Computers in Industry. 2008 Mar 1;59(2):296–303.

100. Sanna A, Manuri F, Lamberti F, Paravati G, Pezzolla P. Using handheld devices to sup port augmented reality-based maintenance and assembly tasks. In: 2015 IEEE International Conference on Consumer Electronics (ICCE). 2015. p. 178–9.

101. Westerfield G, Mitrovic A, Billinghurst M. Intelligent Augmented Reality Training for Motherboard Assembly. Int J Artif Intell Educ. 2015 Mar 1;25(1):157–72.

102. Wang X, Ong SK, Nee AYC. Real-virtual components interaction for assembly simulation and planning. Robotics and Computer-Integrated Manufacturing. 2016 Oct 1;41:102–14.

103. Li X, Cai Y, Wang S, Lu T. Learning Category-level Implicit 3D Rotation Representations for 6D Pose Estimation from RGB Images. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). 2019. p. 2310–5.

104. Muñoz E, Konishi Y, Beltran C, Murino V, Bue AD. Fast 6D pose from a single RGB image using Cascaded Forests Templates. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2016. p. 4062–9.

105. Liu F, Fang P, Yao Z, Fan R, Pan Z, Sheng W, Yang H. Recovering 6D object pose from RGB indoor image based on two-stage detection network with multi-task loss. Neurocomputing. 2019 Apr 14;337:15–23.

106. Zuo G, Zhang C, Liu H, Gong D. Low-Quality Rendering-Driven 6D Object Pose Estimation from Single RGB Image. In: 2020 International Joint Conference on Neural Networks (IJCNN). 2020. p. 1–8.

107. Zhao W, Zhang S, Guan Z, Luo H, Tang L, Peng J, Fan J. 6D object pose estimation via viewpoint relation reasoning. Neurocomputing. 2020 May 14;389:9–17.

108. Josifovski J, Kerzel M, Pregizer C, Posniak L, Wermter S. Object Detection and Pose Estimation Based on Convolutional Neural Networks Trained with Synthetic Data. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018. p. 6269–76.

109. Su H, Qi CR, Li Y, Guibas LJ. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In: 2015 IEEE International Conference on Computer Vision (ICCV). 2015. p. 2686–94.

110. Tanzi L, Vezzetti E, Moreno R, Aprato A, Audisio A, Massè A. Hierarchical fracture classification of proximal femur X-Ray images using a multistage Deep Learning approach. Eur J Radiol. 2020 Dec;133:109373.

111. Tanzi L, Audisio A, Cirrincione G, Aprato A, Vezzetti E. Vision Transformer for femur fracture classification. Injury. 2022 Jul 1;53(7):2625–34.

112. Woolf AD, Pfleger B. Burden of major musculoskeletal conditions. Bull World Health Organ. 2003;81(9):646–56.

113. Reginster JY, Burlet N. Osteoporosis: a still increasing prevalence. Bone. 2006 Feb;38(2 Suppl 1):S4-9.

114. Parker M, Johansen A. Hip fracture. BMJ. 2006 Jul 1;333(7557):27–30.

115. Journal of Orthopaedic Trauma. Femur. 2018 Jan;32:S33–44.

116. Kirby MW, Spritzer C. Radiographic Detection of Hip and Pelvic Fractures in the Emergency Department. American Journal of Roentgenology. 2010 Apr 1;194(4):1054–60.

117. Giannoulis D, Calori GM, Giannoudis PV. Thirty-day mortality after hip fractures: has anything changed? Eur J Orthop Surg Traumatol. 2016 May;26(4):365–70.

118. Jin WJ, Dai LY, Cui YM, Zhou Q, Jiang LS, Lu H. Reliability of classification systems for intertrochanteric fractures of the proximal femur in experienced orthopaedic surgeons. Injury. 2005 Jul;36(7):858–61.

119. Blundell CM, Parker MJ, Pryor GA, Hopkinson-Woolley J, Bhonsle SS. Assessment of the AO classification of intracapsular fractures of the proximal femur. The Journal of Bone and Joint Surgery British volume. 1998 Jul;80-B(4):679–83.

120. van Embden D, Rhemrev SJ, Meylaerts SAG, Roukema GR. The comparison of two classifications for trochanteric femur fractures: The AO/ASIF classification and the Jensen classification. Injury. 2010 Apr;41(4):377–81.

121. Cohen JF, Korevaar DA, Altman DG, Bruns DE, Gatsonis CA, Hooft L, Irwig L, Levine D, Reitsma JB, de Vet HCW, Bossuyt PMM. STARD 2015 guidelines for reporting diagnostic accuracy studies: explanation and elaboration. BMJ Open. 2016 Nov;6(11):e012799.

122. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. Las Vegas, NV, USA: IEEE; 2016 [cited 2019 Nov 25]. p. 2818–26. Available from: http://ieeexplore.ieee.org/document/7780677/

123. Fei-Fei L, Deng J, Li K. ImageNet: Constructing a large-scale image database. Journal of Vision. 2010 Mar 22;9(8):1037–1037.

124. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv:14126980 [cs] [Internet]. 2017 Jan 29 [cited 2019 Nov 27]; Available from: http://arxiv.org/abs/1412.6980

125. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:14091556 [cs] [Internet]. 2015 Apr 10 [cited 2019 Nov 25]; Available from: http://arxiv.org/abs/1409.1556

126. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. Las Vegas, NV, USA: IEEE; 2016 [cited 2019 Nov 25]. p. 770–8. Available from: http://ieeexplore.ieee.org/document/7780459/

127. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: 2017 IEEE International Conference on Computer Vision (ICCV) [Internet]. Venice: IEEE; 2017 [cited 2019 Nov 25]. p. 618–26. Available from: http://ieeexplore.ieee.org/document/8237336/

128. Chollet F, others. Keras [Internet]. 2015. Available from: https://keras.io

129. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia Y, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Internet]. 2015. Available from: http://tensorflow.org/

130. McHugh ML. Interrater reliability: the kappa statistic. Biochem Med (Zagreb). 2012;22(3):276–82.

131. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning Deep Features for Discriminative Localization. arXiv:151204150 [cs] [Internet]. 2015 Dec 13 [cited 2019 Dec 18]; Available from: http://arxiv.org/abs/1512.04150

132. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017 May 24;60(6):84–90.

133. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 6000–10. (NIPS'17).

134. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) [Internet]. Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 4171–86. Available from: https://www.aclweb.org/anthology/N19-1423

135. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018;

136. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-End Object Detection with Transformers. In: Vedaldi A, Bischof H, Brox T, Frahm JM, editors. Computer Vision – ECCV 2020. Cham: Springer International Publishing; 2020. p. 213–29. (Lecture Notes in Computer Science).

137. Ye L, Rochan M, Liu Z, Wang Y. Cross-Modal Self-Attention Network for Referring Image Segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. Los Alamitos, CA, USA: IEEE Computer Society; 2019. p. 10494–503. Available from: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.01075

138. Girdhar R, Carreira JJ, Doersch C, Zisserman A. Video Action Transformer Network. In IEEE Computer Society; 2019 [cited 2021 Jun 25]. p. 244–53. Available from: https://www.computer.org/csdl/proceedings-article/cvpr/2019/329300a244/1gyrQLoZXws

139. Zhang H, Goodfellow I, Metaxas D, Odena A. Self-Attention Generative Adversarial Networks. arXiv:180508318 [cs, stat] [Internet]. 2019 Jun 14 [cited 2021 Jun 25]; Available from: http://arxiv.org/abs/1805.08318

140. Lee C, Jang J, Lee S, Kim YS, Jo HJ, Kim Y. Classification of femur fracture in pelvic X-ray images using meta-learned deep neural network. Sci Rep. 2020 Aug 13;10(1):13694.

141. Kazi A, Albarqouni S, Sanchez AJ, Kirchhoff S, Biberthaler P, Navab N, Mateus D. Automatic Classification of Proximal Femur Fractures Based on Attention Models. In: Wang Q, Shi Y, Suk HI, Suzuki K, editors. Machine Learning in Medical Imaging. Cham: Springer International Publishing; 2017. p. 70–8. (Lecture Notes in Computer Science).

142. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations [Internet]. 2021. Available from: https://openreview.net/forum?id=YicbFdNTTy

143. Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv:180402767 [cs] [Internet]. 2018 Apr 8 [cited 2021 Mar 23]; Available from: http://arxiv.org/abs/1804.02767

144. Cohen JF, Korevaar DA, Altman DG, Bruns DE, Gatsonis CA, Hooft L, Irwig L, Levine D, Reitsma JB, de Vet HCW, Bossuyt PMM. STARD 2015 guidelines for reporting diagnostic accuracy studies: explanation and elaboration. BMJ Open. 2016 Nov;6(11):e012799.

145. Hassani A, Walton S, Shah N, Abuduweili A, Li J, Shi H. Escaping the Big Data Paradigm with Compact Transformers. arXiv:210405704 [cs] [Internet]. 2021 Aug 13 [cited 2021 Oct 19]; Available from: http://arxiv.org/abs/2104.05704

146. Doi K. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. Computerized Medical Imaging and Graphics. 2007 Jun 1;31(4):198–211.

147. Balki I, Amirabadi A, Levman J, Martel AL, Emersic Z, Meden B, Garcia-Pedrero A, Ramirez SC, Kong D, Moody AR, Tyrrell PN. Sample-Size Determination Methodologies for Machine Learning in Medical Imaging Research: A Systematic Review. Can Assoc Radiol J. 2019 Nov;70(4):344–53.

148. Baum E, Haussler D. What Size Net Gives Valid Generalization? In: Advances in Neural Information Processing Systems [Internet]. Morgan-Kaufmann; 1988 [cited 2022 Nov 18]. Available from: https://papers.nips.cc/paper/1988/hash/1d7f7abc18fcb43975065399b0d1e48e-Abstract.html

149. Jollans L, Boyle R, Artiges E, Banaschewski T, Desrivières S, Grigis A, Martinot JL, Paus T, Smolka MN, Walter H, Schumann G, Garavan H, Whelan R. Quantifying performance of machine learning methods for neuroimaging data. NeuroImage. 2019;199:351–65.

150. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q. A Comprehensive Survey on Transfer Learning. arXiv preprint [Internet]. 2019;abs/1911.02685. Available from: http://arxiv.org/abs/1911.02685

151. Blockeel H, Bruynooghe M, Džeroski S, Ramon J, Struyf J. Hierarchical multi-classification. In: Workshop Notes of the KDD'02 Workshop on Multi-Relational Data Mining [Internet]. 2002. p. 21–35. Available from: https://lirias.kuleuven.be/retrieve/398633

152. Zhu X, Bain M. B-CNN: Branch Convolutional Neural Network for Hierarchical Classification. arXiv preprint [Internet]. 2017;abs/1709.09890. Available from: http://arxiv.org/abs/1709.09890

153. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. University of Toronto; 2009.

154. Maji S, Rahtu E, Kannala J, Blaschko MB, Vedaldi A. Fine-Grained Visual Classification of Aircraft. arXiv preprint [Internet]. 2013;abs/1306.5151. Available from: http://arxiv.org/abs/1306.5151

155. Gaber T, Tharwat A, Ibrahim A, Hassanien AE. Linear discriminant analysis : a detailed tutorial. AI Communications. 2017;30(2):169–90.

156. Zhang H, He G, Peng J, Kuang Z, Fan J. Deep Learning of Path-Based Tree Classifiers for Large-Scale Plant Species Identification. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). 2018. p. 25–30.

157. Srivastava N, Salakhutdinov R. Discriminative Transfer Learning with Tree-based Priors. In: Advances in Neural Information Processing Systems 26 (NIPS) [Internet]. 2013. p. 2094–102. Available from: https://proceedings.neurips.cc/paper/2013/hash/9ac403da7947a183884c18a67d3aa8de-Abstract.html

158. Salakhutdinov R, Torralba A, Tenenbaum JB. Learning to share visual appearance for multiclass object detection. In: 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. 2011. p. 1481–8. Available from: https://doi.org/10.1109/CVPR.2011.5995720

159. Babbar R, Partalas I, Gaussier E, Amini MR, Amblard C. Learning Taxonomy Adaptation in Large-scale Classification. Journal of Machine Learning Research. 2016;17(98):1–37.

160. Zhao B, Fei-Fei L, Xing EP. Large-Scale Category Structure Aware Image Categorization. In: Advances in Neural Information Processing Systems 24 (NIPS) [Internet]. 2011. p. 1251–9. Available from: https://proceedings.neurips.cc/paper/2011/hash/d5cfead94f5350c12c322b5b664544c1-Abstract.html

161. Goo W, Kim J, Kim G, Hwang SJ. Taxonomy-Regularized Semantic Deep Convolutional Neural Networks. In: 14th European Conference on Computer Vision (ECCV) [Internet]. Springer; 2016. p. 86–101. (Lecture Notes in Computer Science; vol. 9906). Available from: https://doi.org/10.1007/978-3-319-46475-6_6

162. Wu Z, Saito S. HiNet: Hierarchical Classification with Neural Network. arXiv preprint [Internet]. 2017;abs/1705.11105. Available from: http://arxiv.org/abs/1705.11105

163. Xie S, Yang T, Wang X, Lin Y. Hyper-class augmented and regularized deep learning for fine-grained image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Internet]. 2015. p. 2645–54. Available from: https://doi.org/10.1109/CVPR.2015.7298880

164. Neal RM, Shahbaba B. Improving classification when a class hierarchy is available using a hierarchy-based prior. Bayesian Analysis. 2007;2(1):221–37.

165. Wu TY, Morgado P, Wang P, Ho CH, Vasconcelos N. Solving Long-Tailed Recognition with Deep Realistic Taxonomic Classifier. In: 16th European Conference on Computer Vision (ECCV) [Internet]. Springer; 2020. p. 171–89. (Lecture Notes in Computer Science; vol. 12353). Available from: https://doi.org/10.1007/978-3-030-58598-3_11

166. Bu L, Zhao D, Liu Y, Guan Q. A hierarchical classification algorithm for evaluating energy consumption behaviors. In: 2014 International Joint Conference on Neural Networks (IJCNN). 2014. p. 1461–6.

167. Nievola JC, Paraiso EC, Freitas AA. A hierarchical neural network for predicting protein functions. In: 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE). 2015. p. 1–5.

168. Koo J, Klabjan D, Utke J. Combined Convolutional and Recurrent Neural Networks for Hierarchical Classification of Images. In: IEEE International Conference on Big Data (BigData) [Internet]. 2020. p. 1354–61. Available from: https://doi.org/10.1109/BigData50022.2020.9378237

169. Yan Z, Zhang H, Piramuthu R, Jagadeesh V, DeCoste D, Di W, Yu Y. HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015 [Internet]. 2015. p. 2740–8. Available from: https://doi.org/10.1109/ICCV.2015.314

170. Deng J, Ding N, Jia Y, Frome A, Murphy K, Bengio S, Li Y, Neven H, Adam H. Large-Scale Object Classification Using Label Relation Graphs. In: 13th European Conference on Computer Vision (ECCV) [Internet]. Springer; 2014. p. 48–64. (Lecture Notes in Computer Science; vol. 8689). Available from: https://doi.org/10.1007/978-3-319-10590-1_4

171. Wang R, cai D, Xiao K, Jia X, Han X, Meng D. Label Hierarchy Transition: Modeling Class Hierarchies to Enhance Deep Classifiers. arXiv preprint [Internet]. 2021;abs/2112.02353. Available from: https://arxiv.org/abs/2112.02353

172. Deng J, Berg AC, Li K, Fei-Fei L. What does classifying more than 10,000 image categories tell us? In: European Conference on Computer Vision (ECCV). Springer; 2010. p. 71–84.

173. Deselaers T, Ferrari V. Visual and semantic similarity in ImageNet. In: 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2011. p. 1777–84.

174. Bertinetto L, Mueller R, Tertikas K, Samangooei S, Lord NA. Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks [Internet]. arXiv; 2020 [cited 2022 Nov 18]. Available from: http://arxiv.org/abs/1912.09393

175. Singh MS, Pondenkandath V, Zhou B, Lukowicz P, Liwicki M. Transforming Sensor Data to the Image Domain for Deep Learning - an Application to Footstep Detection. arXiv preprint [Internet]. 2017;abs/1701.01077. Available from: http://arxiv.org/abs/1701.01077

176. Pondenkandath V, Alberti M, Eichenberger N, Ingold R, Liwicki M. Identifying Cross-Depicted Historical Motifs. arXiv preprint [Internet]. 2018;abs/1804.01728. Available from: http://arxiv.org/abs/1804.01728

177. Kermany DS, Goldbaum M, Cai W, Valentim CCS, Liang H, Baxter SL, McKeown A, Yang G, Wu X, Yan F, Dong J, Prasadha MK, Pei J, Ting MYL, Zhu J, Li C, Hewett S, Dong J, Ziyar I, Shi A, Zhang R, Zheng L, Hou R, Shi W, Fu X, Duan Y, Huu VAN, Wen C, Zhang ED, Zhang CL, Li O, Wang X, Singer MA, Sun X, Xu J, Tafreshi A, Lewis MA, Xia H, Zhang K. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell. 2018;172(5):1122-1131.e9.

178. Nonis F, Dagnes N, Marcolin F, Vezzetti E. 3D Approaches and challenges in facial expression recognition algorithms—a literature review. Applied Sciences. 2019;9(18):3904.

179. Ahmed E, Saint A, Shabayek AER, Cherenkova K, Das R, Gusev G, Aouada D, Ottersten B. A survey on deep learning advances on different 3D data representations. arXiv preprint arXiv:180801462. 2018;

180. Zhou S, Xiao S. 3D face recognition: a survey. Human-centric Computing and Information Sciences. 2018;8(1):1–27.

181. Yin L, Wei X, Sun Y, Wang J, Rosato MJ. A 3D facial expression database for facial behavior research. In: 7th international conference on automatic face and gesture recognition (FGR06). IEEE; 2006. p. 211–6.

182. Savran A, Alyüz N, Dibeklioğlu H, Çeliktutan O, Gökberk B, Sankur B, Akarun L. Bosphorus database for 3D face analysis. In: European workshop on biometrics and identity management. Springer; 2008. p. 47–56.

183. Ulrich L, Marcolin F, Vezzetti E, Nonis F, Scurati GW, Dozio N, Ferrise F. CALD3R: a spontaneous 3D facial expression database. Under review.

184. Ashrafian H, Clancy O, Grover V, Darzi A. The evolution of robotic surgery: surgical and anaesthetic aspects. Br J Anaesth. 2017 Dec 1;119(suppl_1):i72–84.

185. Buchholz NNP, Bach C. The age of robotic surgery – Is laparoscopy dead? Arab J Urol. 2018 Jul 17;16(3):261.

186. Fischer J, Neff M, Freudenstein D, Bartz D. Medical augmented reality based on commercial image guided surgery. In: Proceedings of the Tenth Eurographics conference on Virtual Environments. Goslar, DEU: Eurographics Association; 2004. p. 83–6. (EGVE'04).

187. Nakamoto M, Ukimura O, Faber K, Gill IS. Current progress on augmented reality visualization in endoscopic surgery. Curr Opin Urol. 2012 Mar;22(2):121–6.

188. Gribaudo M, Piazzolla P, Porpiglia F, Vezzetti E, Violante MG. 3D augmentation of the surgical video stream: Toward a modular approach. Computer Methods and Programs in Biomedicine. 2020 Jul 1;191:105505.

189. Huynh LM, Ahlering TE. Robot-Assisted Radical Prostatectomy: A Step-by-Step Guide. Journal of Endourology. 2018 May 1;32(Suppl 1):S-28.

190. Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Cui C, Corrado G, Thrun S, Dean J. A guide to deep learning in healthcare. Nature Medicine. 2019 Jan;25(1):24–9.

191. Bernhardt S, Nicolau SA, Soler L, Doignon C. The status of augmented reality in laparoscopic surgery as of 2016. Med Image Anal. 2017;37:66–90.

192. Wengert C, Cattin PC, Duff JM, Baur C, Székely G. Markerless Endoscopic Registration and Referencing. In: Larsen R, Nielsen M, Sporring J, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006. Berlin, Heidelberg: Springer; 2006. p. 816–23.

193. Edwards PJ, King AP, Maurer CR, de Cunha DA, Hawkes DJ, Hill DL, Gaston RP, Fenlon MR, Jusczyzck A, Strong AJ, Chandler CL, Gleeson MJ. Design and evaluation of a system for microscope-assisted guided

interventions (MAGI). IEEE transactions on medical imaging. 2000 Nov;19(11):1082–93.

194. Twinanda AP, Shehata S, Mutter D, Marescaux J, Mathelin M de, Padoy N. EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. IEEE Transactions on Medical Imaging. 2017;36(1):86–97.

195. Jin Y, Dou Q, Chen H, Yu L, Qin J, Fu CW, Heng PA. SV-RCNet: Workflow Recognition From Surgical Videos Using Recurrent Convolutional Network. IEEE Trans Med Imaging. 2018 May;37(5):1114–26.

196. Hansen L, Siebert M, Diesel J, Heinrich MP. Fusing information from multiple 2D depth cameras for 3D human pose estimation in the operating room. Int J CARS. 2019 Nov;14(11):1871–9.

197. Belagiannis V, Wang X, Shitrit HBB, Hashimoto K, Stauder R, Aoki Y, Kranzfelder M, Schneider A, Fua P, Ilic S, Feussner H, Navab N. Parsing human skeletons in an operating room. Machine Vision and Applications. 2016 Oct;27(7):1035–46.

198. Zhou T, Wachs JP. Spiking Neural Networks for early prediction in human–robot collaboration. The International Journal of Robotics Research. 2019 Dec;38(14):1619–43.

199. Ha Ho-Gun Hong Jaesung. Augmented Reality in Medicine. hmr. 2016;36(4):242–7.

200. Tanzi L, Piazzolla P, Porpiglia F, Vezzetti E. Real-time deep learning semantic segmentation during intra-operative surgery for 3D augmented reality assistance. Int J CARS [Internet]. 2021 Sep 1 [cited 2021 Oct 9];16(9):1435–45. Available from: https://doi.org/10.1007/s11548-021-02432-y

201. Padovan E, Marullo G, Tanzi L, Piazzolla P, Moos S, Porpiglia F, Vezzetti E. A deep learning framework for real-time 3D model registration in robot-assisted laparoscopic surgery. The International Journal of Medical Robotics and Computer Assisted Surgery. 2022;18(3):e2387.

202. Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;39(12):2481–95.

203. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid Scene Parsing Network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. p. 6230–9.

204. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Bengio Y, LeCun Y, editors. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015. 2015.

205. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR [Internet]. 2017;abs/1704.04861. Available from: http://arxiv.org/abs/1704.04861

206. Wada K. labelme: Image Polygonal Annotation with Python [Internet]. 2016. Available from: https://github.com/wkentaro/labelme

207. Suzuki S, be K. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing. 1985 Apr 1;30(1):32–46.

208. Sklansky J. Finding the convex hull of a simple polygon. Pattern Recognition Letters. 1982 Dec 1;1(2):79–83.

209. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv:14126980 [cs] [Internet]. 2017 Jan 29 [cited 2019 Nov 27]; Available from: http://arxiv.org/abs/1412.6980

210. Chollet F, others. Keras [Internet]. 2015. Available from: https://keras.io

211. Gupta D. keras-segmentation [Internet]. 2019. Available from: https://github.com/divamgupta/image-segmentation-keras

212. Porpiglia F, Fiori C, Checcucci E, Amparore D, Bertolo R. Hyperaccuracy Three-dimensional Reconstruction Is Able to Maximize the Efficacy of Selective Clamping During Robot-assisted Partial Nephrectomy for Complex Renal Masses. European Urology [Internet]. 2018 Nov 1 [cited 2021 Nov 8];74(5):651–60. Available from: https://www.sciencedirect.com/science/article/pii/S0302283817310953

213. Collins T, Bartoli A, Bourdel N, Canis M. Robust, Real-Time, Dense and Deformable 3D Organ Tracking in Laparoscopic Videos. In: Ourselin S, Joskowicz L, Sabuncu MR, Unal G, Wells W, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. Cham: Springer International Publishing; 2016. p. 404–12. (Lecture Notes in Computer Science).

214. Kong SH, Haouchine N, Soares R, Klymchenko A, Andreiuk B, Marques B, Shabat G, Piechaud T, Diana M, Cotin S, Marescaux J. Robust augmented

reality registration method for localization of solid organs' tumors using CT-derived virtual biomechanical model and fluorescent fiducials. Surg Endosc [Internet]. 2017 Jul 1 [cited 2021 Nov 5];31(7):2863–71. Available from: https://doi.org/10.1007/s00464-016-5297-8

215. Muñoz E, Konishi Y, Beltran C, Murino V, Del Bue A. Fast 6D pose from a single RGB image using Cascaded Forests Templates. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2016. p. 4062–9.

216. Li X, Cai Y, Wang S, Lu T. Learning Category-level Implicit 3D Rotation Representations for 6D Pose Estimation from RGB Images. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). 2019. p. 2310–5.

217. Liu F, Fang P, Yao Z, Fan R, Pan Z, Sheng W, Yang H. Recovering 6D object pose from RGB indoor image based on two-stage detection network with multi-task loss. Neurocomputing [Internet]. 2019 Apr 14 [cited 2021 Oct 8];337:15–23. Available from: https://www.sciencedirect.com/science/article/pii/S0925231218315236

218. Zuo G, Zhang C, Liu H, Gong D. Low-Quality Rendering-Driven 6D Object Pose Estimation from Single RGB Image. In: 2020 International Joint Conference on Neural Networks (IJCNN). 2020. p. 1–8.

219. Capellen C, Schwarz M, Behnke S. ConvPoseCNN: Dense Convolutional 6D Object Pose Estimation. In: 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications [Internet]. Valletta, Malta; 2020. p. 162–72. Available from: https://www.scitepress.org/PublicationsDetail.aspx?ID=QyUmCPzkF8E=&t=1

220. Tanzi L, Piazzolla P, Vezzetti E. Intraoperative surgery room management: A deep learning perspective. Int J Med Robot. 2020 Oct;16(5):1–12.

221. Tanzi L, Vezzetti E, Moreno R, Aprato A, Audisio A, Massè A. Hierarchical fracture classification of proximal femur X-Ray images using a multistage Deep Learning approach. Eur J Radiol. 2020 Dec;133:109373.

222. Olivetti EC, Ferretti J, Cirrincione G, Nonis F, Tornincasa S, Marcolin F. Deep CNN for 3D Face Recognition. In: Rizzi C, Andrisano AO, Leali F, Gherardini F, Pini F, Vergnano A, editors. Design Tools and Methods in Industrial Engineering. Cham: Springer International Publishing; 2020. p. 665–74. (Lecture Notes in Mechanical Engineering).

223. Building an Ecologically Valid Facial Expression Database – Behind the Scenes [Internet]. springerprofessional.de. [cited 2021 Dec 10]. Available from: https://www.springerprofessional.de/en/building-an-ecologically-valid-facial-expression-database-behind/19325736

224. Peng S, Liu Y, Huang Q, Zhou X, Bao H. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019. p. 4556–65.

225. Zhao W, Zhang S, Guan Z, Luo H, Tang L, Peng J, Fan J. 6D object pose estimation via viewpoint relation reasoning. Neurocomputing [Internet]. 2020 May 14;389:9–17. Available from: https://www.sciencedirect.com/science/article/pii/S0925231220300333

226. Massa F, Russell BC, Aubry M. Deep Exemplar 2D-3D Detection by Adapting from Real to Rendered Views. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. p. 6024–33.

227. Tjaden H, Schwanecke U, Schömer E. Real-Time Monocular Pose Estimation of 3D Objects Using Temporally Consistent Local Color Histograms. In: 2017 IEEE International Conference on Computer Vision (ICCV). 2017. p. 124–32.

228. Hu Y, Hugonot J, Fua P, Salzmann M. Segmentation-Driven 6D Object Pose Estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019. p. 3380–9.

229. Rad M, Lepetit V. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. In: 2017 IEEE International Conference on Computer Vision (ICCV). 2017. p. 3848–56.

230. Konishi Y, Hanzawa Y, Kawade M, Hashimoto M. Fast 6D Pose Estimation from a Monocular Image Using Hierarchical Pose Trees. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016. Cham: Springer International Publishing; 2016. p. 398–413. (Lecture Notes in Computer Science).

231. Su H, Qi CR, Li Y, Guibas LJ. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In: 2015 IEEE International Conference on Computer Vision (ICCV). 2015. p. 2686–94.

232. Kehl W, Manhardt F, Tombari F, Ilic S, Navab N. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In: 2017 IEEE International Conference on Computer Vision (ICCV). 2017. p. 1530–8.

233. Xiang Y, Schmidt T, Narayanan V, Fox D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In: Robotics: Science and Systems XIV [Internet]. Robotics: Science and Systems Foundation; 2018 [cited 2021 Oct 13]. Available from: http://www.roboticsproceedings.org/rss14/p19.pdf

234. Tekin B, Sinha SN, Fua P. Real-Time Seamless Single Shot 6D Object Pose Prediction. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018. p. 292–301.

235. Bernhardt S, Nicolau SA, Soler L, Doignon C. The status of augmented reality in laparoscopic surgery as of 2016. Medical Image Analysis [Internet]. 2017 Apr 1 [cited 2021 Nov 4];37:66–90. Available from: https://www.sciencedirect.com/science/article/pii/S1361841517300178

236. Le Roy B, Ozgur E, Koo B, Buc E, Bartoli A. Augmented reality guidance in laparoscopic hepatectomy with deformable semi-automatic computed tomography alignment (with video). Journal of Visceral Surgery. 2019 Jun 1;156(3):261–2.

237. Puerto-Souza GA, Mariottini GL. A Fast and Accurate Feature-Matching Algorithm for Minimally-Invasive Endoscopic Images. IEEE Transactions on Medical Imaging. 2013 Jul;32(7):1201–14.

238. Kim JH, Bartoli A, Collins T, Hartley R. Tracking by Detection for Interactive Image Augmentation in Laparoscopy. In: Dawant BM, Christensen GE, Fitzpatrick JM, Rueckert D, editors. Biomedical Image Registration. Berlin, Heidelberg: Springer; 2012. p. 246–55. (Lecture Notes in Computer Science).

239. Özgür E, Koo B, Le Roy B, Buc E, Bartoli A. Preoperative liver registration for augmented monocular laparoscopy using backward–forward biomechanical simulation. Int J CARS. 2018 Oct 1;13(10):1629–40.

240. Collins T, Pizarro D, Bartoli A, Canis M, Bourdel N. Computer-Assisted Laparoscopic myomectomy by augmenting the uterus with pre-operative MRI data. In: 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). 2014. p. 243–8.

241. Puerto-Souza GA, Cadeddu JA, Mariottini GL. Toward Long-Term and Accurate Augmented-Reality for Monocular Endoscopic Videos. IEEE Transactions on Biomedical Engineering. 2014 Oct;61(10):2609–20.

242. Soler L, Nicolau S, Pessaux P, Mutter D, Marescaux J. Real-time 3D image reconstruction guidance in liver resection surgery. Hepatobiliary Surg Nutr. 2014 Apr;3(2):73–81.

243. Bourdel N, Collins T, Pizarro D, Debize C, Grémeau A sophie, Bartoli A, Canis M. Use of augmented reality in laparoscopic gynecology to visualize myomas. Fertility and Sterility. 2017 Mar 1;107(3):737–9.

244. Ozyoruk KB, Gokceler GI, Bobrow TL, Coskun G, Incetan K, Almalioglu Y, Mahmood F, Curto E, Perdigoto L, Oliveira M, Sahin H, Araujo H, Alexandrino H, Durr NJ, Gilbert HB, Turan M. EndoSLAM dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos. Medical Image Analysis. 2021 Jul 1;71:102058.

245. Kalia M, Mathur P, Tsang K, Black P, Navab N, Salcudean S. Evaluation of a marker-less, intra-operative, augmented reality guidance system for robot-assisted laparoscopic radical prostatectomy. Int J CARS. 2020 Jul 1;15(7):1225–33.

246. Gupta D. Image Segmentation Keras : Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras. [Internet]. 2021 [cited 2021 Nov 8]. Available from: https://github.com/divamgupta/image-segmentation-keras

247. Vávra P, Roman J, Zonča P, Ihnát P, Němec M, Kumar J, Habib N, El-Gendi A. Recent Development of Augmented Reality in Surgery: A Review. Journal of Healthcare Engineering [Internet]. 2017 Aug 21 [cited 2021 Nov 4];2017:e4574172. Available from: https://www.hindawi.com/journals/jhe/2017/4574172/

248. Gribaudo M, Piazzolla P, Porpiglia F, Vezzetti E, Violante MG. 3D augmentation of the surgical video stream: Toward a modular approach. Computer Methods and Programs in Biomedicine [Internet]. 2020 Jul 1 [cited 2021 Nov 4];191:105505. Available from: https://www.sciencedirect.com/science/article/pii/S0169260719322801

249. Porpiglia F, Checcucci E, Amparore D, Manfredi M, Massa F, Piazzolla P, Manfrin D, Piana A, Tota D, Bollito E, Fiori C. Three-dimensional Elastic Augmented-reality Robot-assisted Radical Prostatectomy Using Hyperaccuracy Three-dimensional Reconstruction Technology: A Step

Further in the Identification of Capsular Involvement. European Urology. 2019 Oct 1;76(4):505–14.

250. Porpiglia F, Checcucci E, Amparore D, Piramide F, Volpi G, Granato S, Verri P, Manfredi M, Bellin A, Piazzolla P, Autorino R, Morra I, Fiori C, Mottrie A. Three-dimensional Augmented Reality Robot-assisted Partial Nephrectomy in Case of Complex Tumours (PADUA ≥10): A New Intraoperative Tool Overcoming the Ultrasound Guidance. European Urology [Internet]. 2020 Aug 1 [cited 2021 Nov 10];78(2):229–38. Available from: https://www.sciencedirect.com/science/article/pii/S0302283819308978

251. Laparoscopy (keyhole surgery) [Internet]. nhs.uk. 2018 [cited 2022 Sep 30]. Available from: https://www.nhs.uk/conditions/laparoscopy/

252. Kaping'a F. Deep learning for action and event detection in endoscopic videos for robotic assisted laparoscopy. :6.

253. Tomimaru Y, Noguchi K, Morita S, Imamura H, Iwazawa T, Dono K. Is Intraoperative Blood Loss Underestimated in Patients Undergoing Laparoscopic Hepatectomy? World J Surg. 2018 Nov 1;42(11):3685–91.

254. Kurian E, Kizhakethottam JJ, Mathew J. Deep learning based Surgical Workflow Recognition from Laparoscopic Videos. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES). 2020. p. 928–31.

255. Hua S, Gao J, Wang Z, Yeerkenbieke P, Li J, Wang J, He G, Jiang J, Lu Y, Yu Q, Han X, Liao Q, Wu W. Automatic bleeding detection in laparoscopic surgery based on a faster region-based convolutional neural network. Annals of Translational Medicine. 2022 May;10(10):546–546.

256. Madad Zadeh S, Francois T, Calvet L, Chauvet P, Canis M, Bartoli A, Bourdel N. SurgAI: deep learning for computerized laparoscopic image understanding in gynaecology. Surg Endosc [Internet]. 2020 Dec 1 [cited 2022 Oct 3];34(12):5377–83. Available from: https://doi.org/10.1007/s00464-019-07330-8

257. Rabbani N, Seve C, Bourdel N, Bartoli A. Video-based Computer-aided Laparoscopic Bleeding Management: a Space-time Memory Neural Network with Positional Encoding and Adversarial Domain Adaptation. In 2022 [cited 2022 Oct 3]. Available from: https://openreview.net/forum?id=kmV0i37vuCy

258. Zegers M, de Bruijne MC, de Keizer B, Merten H, Groenewegen PP, van der Wal G, Wagner C. The incidence, root-causes, and outcomes of adverse events in surgical units: implication for potential prevention strategies. Patient Saf Surg. 2011 May 20;5(1):13.

259. Kaushik R. Bleeding complications in laparoscopic cholecystectomy: Incidence, mechanisms, prevention and management. Journal of Minimal Access Surgery [Internet]. 2010 Jan 7 [cited 2022 Oct 3];6(3):59. Available from: https://www.journalofmas.com/article.asp?issn=0972-9941;year=2010;volume=6;issue=3;spage=59;epage=65;aulast=Kaushik;type=0

260. Richter F, Shen S, Liu F, Huang J, Funk EK, Orosco RK, Yip MC. Autonomous Robotic Suction to Clear the Surgical Field for Hemostasis Using Image-Based Blood Flow Detection. IEEE Robotics and Automation Letters. 2021 Apr;6(2):1383–90.

261. Garcia-Martinez A, Vicente-Samper JM, Sabater-Navarro JM. Automatic detection of surgical haemorrhage using computer vision. Artificial Intelligence in Medicine [Internet]. 2017 May 1 [cited 2022 Oct 3];78:55–60. Available from: https://www.sciencedirect.com/science/article/pii/S0933365716305590

262. Fu Y, Mandal M, Guo G. Bleeding region detection in WCE images based on color features and neural network. In: 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS). 2011. p. 1–4.

263. Fu Y, Zhang W, Mandal M, Meng MQH. Computer-Aided Bleeding Detection in WCE Video. IEEE Journal of Biomedical and Health Informatics. 2014 Mar;18(2):636–42.

264. Okamoto T, Ohnishi T, Kawahira H, Dergachyava O, Jannin P, Haneishi H. Real-time identification of blood regions for hemostasis support in laparoscopic surgery. SIViP [Internet]. 2019 Mar 1 [cited 2022 Oct 3];13(2):405–12. Available from: https://doi.org/10.1007/s11760-018-1369-7

265. Wei H, Rudzicz F, Fleet D, Grantcharov T, Taati B. Intraoperative Adverse Event Detection in Laparoscopic Surgery: Stabilized Multi-Stage Temporal Convolutional Network with Focal-Uncertainty Loss. In: Proceedings of the 6th Machine Learning for Healthcare Conference [Internet]. PMLR; 2021 [cited 2022 Sep 21]. p. 283–307. Available from: https://proceedings.mlr.press/v149/wei21a.html

266. Jia X, Meng MQH. A deep convolutional neural network for bleeding detection in Wireless Capsule Endoscopy images. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2016. p. 639–42.

267. Ji J, Buch S, Soto A, Niebles JC. End-to-End Joint Semantic Segmentation of Actors and Actions in Video. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer Vision – ECCV 2018. Cham: Springer International Publishing; 2018. p. 734–49. (Lecture Notes in Computer Science).

268. Crawshaw M. Multi-Task Learning with Deep Neural Networks: A Survey [Internet]. arXiv; 2020 [cited 2022 Sep 27]. Available from: http://arxiv.org/abs/2009.09796

269. Zhang Y, Yang Q. A Survey on Multi-Task Learning. IEEE Transactions on Knowledge and Data Engineering. 2021;1–1.

270. Hou R, Chen C, Shah M. An End-to-end 3D Convolutional Neural Network for Action Detection and Segmentation in Videos [Internet]. arXiv; 2017 [cited 2022 Oct 3]. Available from: http://arxiv.org/abs/1712.01111

271. Hou R, Chen C, Sukthankar R, Shah M. An Efficient 3D CNN for Action/Object Segmentation in Video [Internet]. arXiv; 2019 [cited 2022 Oct 3]. Available from: http://arxiv.org/abs/1907.08895

272. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham: Springer International Publishing; 2015. p. 234–41. (Lecture Notes in Computer Science).

273. Tanzi L, Piazzolla P, Moos S, Vezzetti E. Exploiting deep learning and augmented reality in fused deposition modeling: a focus on registration. Int J Interact Des Manuf [Internet]. 2022 Nov 17 [cited 2022 Nov 21]; Available from: https://doi.org/10.1007/s12008-022-01107-5

274. Tanzi L, Piazzolla P, Porpiglia F, Vezzetti E. Real-time deep learning semantic segmentation during intra-operative surgery for 3D augmented reality assistance. Int J CARS [Internet]. 2021 Jun 24 [cited 2021 Jun 25]; Available from: https://doi.org/10.1007/s11548-021-02432-y

275. Young T, Hazarika D, Poria S, Cambria E. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. IEEE Computational Intelligence Magazine. 2018 Aug;13(3):55–75.

276. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D. Mastering the game of Go without human knowledge. Nature. 2017 Oct;550(7676):354–9.

277. Cranmer EE, tom Dieck MC, Fountoulaki P. Exploring the value of augmented reality for tourism. Tourism Management Perspectives. 2020 Jul 1;35:100672.

278. Hughes CE, Stapleton CB, Hughes DE, Smith EM. Mixed reality in education, entertainment, and training. IEEE Computer Graphics and Applications. 2005 Nov;25(6):24–30.

279. Nee AYC, Ong SK, Chryssolouris G, Mourtzis D. Augmented reality applications in design and manufacturing. CIRP Annals. 2012 Jan 1;61(2):657–79.

280. Komonen K. A cost model of industrial maintenance for profitability analysis and benchmarking. International Journal of Production Economics. 2002 Sep 1;79(1):15–31.

281. Community BO. Blender - a 3D modelling and rendering package [Internet]. Stichting Blender Foundation, Amsterdam: Blender Foundation; 2018. Available from: http://www.blender.org

282. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. p. 779–88.

283. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Bach F, Blei D, editors. Proceedings of the 32nd International Conference on Machine Learning [Internet]. Lille, France: PMLR; 2015. p. 448–56. (Proceedings of Machine Learning Research; vol. 37). Available from: http://proceedings.mlr.press/v37/ioffe15.html

284. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014;15(56):1929–58.

285. Xiao J, Hays J, Ehinger KA, Oliva A, Torralba A. SUN database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010. p. 3485–92.

286. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results [Internet]. Available from: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html