Detection of Inferior Vena Cava in Ultrasound Scans through a Deep Learning Model

*electronics*

MDPI

*Article*

# Detection of Inferior Vena Cava in Ultrasound Scans through a Deep Learning Model

**Piero Policastro [1], Giovanni Chiarion [1], Francesco Ponzio [2], Leonardo Ermini [3], Stefania Civera [4], Stefano Albani [4,5], Giuseppe Musumeci [4], Silvestro Roatta [3] and Luca Mesin [1,*]**

[1] Mathematical Biology and Physiology, Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy
[2] Department of Regional and Urban Studies and Planning, Politecnico di Torino, 10129 Turin, Italy
[3] Department of Neuroscience, University of Torino, c.so Raffaello 30, 10125 Turin, Italy
[4] Division of Cardiology, Ospedale Ordine Mauriziano di Torino, 10128 Turin, Italy
[5] Division of Cardiology, Umberto Parini Regional Hospital, 11100 Aosta, Italy
* Correspondence: luca.mesin@polito.it; Tel.: +39-011-0904-085

**Abstract:** Ultrasound (US) scans of inferior vena cava (IVC) are widely adopted by healthcare providers to assess patients' volume status. Unfortunately, this technique is extremely operator dependent. Recently, new techniques have been introduced to extract stable and objective information from US images by automatic IVC edge tracking. However, these methods require prior interaction with the operator, which leads to a waste of time and still makes the technique partially subjective. In this paper, two deep learning methods, YOLO (You only look once) v4 and YOLO v4 tiny networks, commonly used for fast object detection, are applied to identify the location of the IVC and to recognise the either long or short axis view of the US scan. The output of these algorithms can be used to remove operator dependency, to reduce the time required to start an IVC analysis, and to automatically recover the vein if it is lost for a few frames during acquisition. The two networks were trained with frames extracted from 18 subjects, labeled by 4 operators. In addition, they were also trained on a linear combination of two frames that extracted information on both tissue anatomy and movement. We observed similar accuracy of the two models in preliminary tests on the entire dataset, so that YOLO v4 tiny (showing much lower computational cost) was selected for additional cross-validation in which training and test frames were taken from different subjects. The classification accuracy was approximately 88% when using original frames, but it reached 95% when pairs of frames were processed to also include information on tissue movements, indicating the importance of accounting for tissue motion to improve the accuracy of our IVC detector.

**Keywords:** ultrasound; inferior vena cava; deep learning; YOLO; YOLO tiny



**Citation:** Policastro, P.; Chiarion, G.; Ponzio, F.; Ermini, L.; Civera, S; Albani, S.; Musumeci, G.; Roatta, S.; Mesin, L. Detection of Inferior Vena Cava in Ultrasound Scans through a Deep Learning Model. *Electronics* **2023**, *12*, 1725. https://doi.org/10.3390/electronics12071725

Academic Editor: Hyunjin Park

Received: 15 February 2023
Revised: 29 March 2023
Accepted: 3 April 2023
Published: 5 April 2023



## 1. Introduction

One of the most difficult duties for clinicians in many clinical departments (including Emergency, Internal Medicine, Cardiology and Intensive Care Unit, ICU) is the assessment of the intravascular volume status [1]. Medical professionals actually use fluid infusion to boost cardiac output and tissue perfusion. The improper amount of fluids, however, might result in pulmonary edema, peripheral edema, or body cavity effusion [2]. Typically, a central venous catheter (CVC) is used to invasively measure the intravascular volume condition [3]. However, due to the potential difficulties brought on by an invasive procedure, this approach is not appropriate for all patient circumstances [3,4].

As ultrasound (US) is noninvasive, it causes no harm to the patient's health, and it is inexpensive to acquire. Doctors frequently employ it for diagnostic and prognostic procedures [5]. For example, US scans have been used to determine the condition of the intravascular volume [6]. In US videos, specialists examine the diameter and collapsibility of inferior vena cava (IVC), which correlate with volume status [7] and right atrial pressure [8,9]. However, this method is not standardized [10]. In order to support the clinicians to obtain reliable findings, some

automated methods have been introduced [11,12]. However, in our previous works [12–14], some preliminary steps are required to start the algorithms: the user must choose the type of view to be examined (long or short axis view, i.e., with the probe either longitudinal or transverse to the IVC) and indicate where the vein is located. Then, the algorithms will start processing the US videos and the vein's movement will be followed. The detection of IVC required by our algorithms introduces some subjectivity, which can be influenced by a number of conditions. The first is the doctor's experience, as the position and shape of the vein might vary greatly from patient to patient. In addition, physicians handle the probe with one hand during the acquisition: thus, the indication of the IVC by a mouse managed by the other hand could be uncomfortable, reducing the accuracy of the selection of the parameters needed to begin the automatic analysis.

This study aims to introduce a deep learning algorithm to deal with the constraints imposed by user choices and turn our algorithms into completely automated software programs. We chose deep learning to detect IVC in US scans because it has produced impressive results in many image-recognition tasks [15–18]. Convolutional neural networks (CNNs) have been extensively used in a wide range of applications, including human pose estimation [19], medical image analysis [20,21], and other computer vision tasks [22]. Some applications on IVC investigation have also been proposed in the literature [23,24]. Recently, the time for classification required by this kind of model has reduced, making real-time application feasible in US video applications, where the frame rate is some tens per second. Specifically, the *You Only Look Once* (YOLO) deep learning algorithm outperformed the R-CNN [25] in terms of speed and provided good classification accuracy [17]. This model is mainly adopted for object detection: it was able to identify more than 9000 different objects [26]. Sporadically, it has been used to analyze medical US images [27].

In this paper, we extend the range of applications of YOLO architecture to the identification of IVC positions in US video clips. We produced different datasets of manually segmented US frames to train and test different detectors. The datasets contain information on anatomy and tissue motion.

- Knowledge of the anatomy of the blood vessels can be useful to distinguish between arteries and veins. Arteries are the blood vessels exiting from the heart and are subjected to high blood pressure. Veins are the blood vessels coming back to the heart; they are subjected to low internal pressure and rely on large compliance to ensure that the blood continues to flow even with a low pressure gradient. Both arteries and veins have three main layers. In order to cope with higher blood pressure, arteries have a thicker tunica media (i.e., the middle layer in the vessel wall) than veins, which are thinner and less elastic (i.e., more compliant). Thus, their shape can change more over time than arteries and is largely affected by the external pressure (in fact, the dynamics of the walls of blood vessels are driven by the transmural pressure, i.e., the internal minus the external pressure [28]).
- IVC exhibits large respirophasic movements, as it is attached to the diaphragm. It is a property which is quite specific of IVC, so it can be very useful to identify and discriminate it from other veins.

In conclusion, we created three datasets of manually delineated IVCs. In the first, we used a single frame, reflecting the anatomy of the tissues; in the second one, we considered the difference between two frames, focusing on information on tissue movements; finally, a linear combination of two frames was used for the third dataset, reflecting both anatomy and movement of the tissues. The classification performance of YOLO applied on the different datasets was assessed and discussed.

## 2. Materials and Methods

### *2.1. Manual Segmentation of US Frames*

We enrolled 18 patients. The Ethics Committee of Mauriziano Hospital (Turin) approved the study (approval number 388/2020; experimental protocol identifier v.1, 1 June 2020), and informed consent was obtained according to the policies of the institutional review board. Anonymization was applied to protect patient data.

X-plane mode US videos of the IVC were obtained for each patient (GE Vivid E95; GE Healthcare, Vingmed Ultrasound, Horten, Norway). X-plane videos provide the user with two simultaneous views of the same target in perpendicular US planes. The videos were exported by means of a frame grabber (Vinmooog HDMI Video Capture; Shenzhen Yiminmin Trading Co., Ltd., Shenzhen, China), sampling at 30 Hz, connected to the video output of the US system and allowing data to be transferred to a PC. The frames had a resolution of 480 × 480 pixels. Custom software (implemented in Matlab, the Mathworks, Natick, Massachusetts, USA) was written to allow the manual segmentation of the frames. The algorithm allows the operator to perform the following actions:

- Delineate the border of the vein with the mouse;
- Select the pitch between two segmented frames (we set a pitch of 10 frames);
- Analyze the video backwards if the vein is not visible at the beginning of the video, or set exactly the number of starting and ending frames when the vein is visible.

The operator had to set a single point to proceed to the next image if the IVC was not visible in a frame.

Two operators were involved in the segmentation of the US video frames using the above software. Two additional operators reviewed and cleaned the dataset, using a specific software (written in Matlab) to analyze the segmentation, find possible anomalies and correct them.

### 2.2. Datasets Creation

The original dataset D, consisting of 8334 segmented frames, was used to create three comparable datasets, referred to as D1, D2, and D3. These datasets were used to train and test our detection method. For each subject, consecutive pairs of images were considered, excluding frames that did not have a corresponding frame after a pitch of 10 frames. D1 was created by collecting all the frames. On the other hand, D2 and D3 were obtained by applying the following equation:

$$AugmentedFrame_{(i)} = |\alpha \cdot frame_{(i+1)} - frame_{(i)}| \quad (1)$$

where $frame_{(i+1)}$ is the target frame on which to detect the IVC.

The frames of D2 were obtained by setting $\alpha = 1$: in this way, information on the movement of tissues was obtained by calculating the difference between consecutive frames. The information obtained from this dataset is believed to be useful in identifying the IVC, as it exhibits significant respirophasic movements. D3 was generated using $\alpha = 2$: both anatomy and movement information were captured, as each image is the sum of the target frame and the movement $frame_{(i+1)} - frame_{(i)}$.

Figure 1 shows two frames from each augmented dataset.

### 2.3. Cross-Validation

Different YOLO models (v4 [29] and v4 tiny [30], described below) were first explored on the entire dataset, fixing the training set as the first 70% of the frames selected from each subject and the test as the last 30%. The best model (in terms of performance and computational cost) was then selected and more deeply tested by cross-validation, separating the subjects used for training and testing. Specifically, to validate the results, we used k-fold ($k = 3$) where the subject's frames were divided into $k$ groups. However, a simple equipartion of the subjects could not be performed since each recorded video consisted of a variable number of frames. From a computational complexity theory perspective, this is an $\mathcal{NP}$-hard problem since it involves a $k$-partition [31]. Thus, an optimization algorithm was designed to maintain a similar number of frames that equally divided the subjects between the folds. The pseudocode of the optimization method is shown in Algorithm 1, with $n\_iter$ set to $10^6$. The maximum difference in frames between the three folds was 32. We performed training and testing in $k$ iterations, leaving one fold for testing and training the model on the other two folds in each step. The performance parameters obtained from each iteration were averaged.

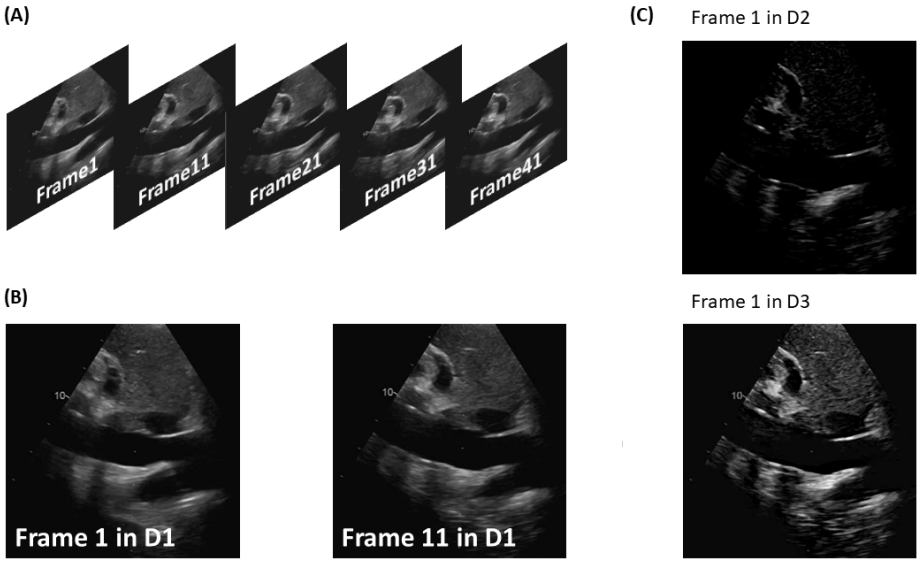


**Figure 1.** Process of creating the augmented datasets. (**A**) Frames extracted from a US video and segmented by the operator. (**B**) Example of a pair of consecutive frames. (**C**) Examples of images included in the two datasets obtained applying Equation (1): dataset 2 provides information on IVC movement; dataset 3 includes images including both information on the anatomy and on the movement of tissues.

**Algorithm 1:** An $n$-dimensional array (where $n$ is the number of subjects) containing integer numbers (indicating the number of frames from each subject) is subdivided into $k$ groups ($k-1$ used for training and one for testing) with $n$ multiple of $k$, by minimizing the differences of the sum of the numbers in each group (i.e., their available frames)

**Input:** $arr$, array of $n$ integer numbers
**Input:** $k \in \mathbb{N} \mid n \equiv 0 \pmod{k}$
**Input:** $n\_iter \in \mathbb{N} \mid n\_iter > 0$
**Result:** $best\_groups$, cell containing the indices of $arr$ as the best subdivision found

$actual\_saved\_frames \leftarrow 0$;
$groups \leftarrow cell(1, k)$;
**for** $iter \leftarrow 1$ *to* $n\_iter$ **do**
  $arr \leftarrow$ RandomShuffle($arr$);
  **foreach** $elem \in arr$ **do**
    $num\_frames \leftarrow$ sum of the element values in each group;
    $min\_i \leftarrow$ argmin($num\_frames$) ;
    $groups(1, min\_i) \leftarrow [groups(1, min\_i), elem]$;
  **end**
  **if** *each cell of groups has the same length* **then**
    $num\_frames \leftarrow$ sum of the element values in each group;
    $saved\_frames \leftarrow k \times$ min($num\_frames$) ;
    **if** $saved\_frames > actual\_saved\_frames$ **then**
      $actual\_saved\_frames \leftarrow saved\_frames$;
      $best\_groups \leftarrow groups$;
    **end**
  **end**
**end**
**return** $best\_groups$;

### *2.4. Deep Learning Model*

We applied a deep learning model called YOLO, which stands for "You Only Look Once". It is a recently introduced object detection algorithm, popular in computer vision and for real-time applications [32]. Given the already remarkable first results, the subsequent tuning over the years has allowed the development of new versions of this deep learning model with exceptional localization efficiency and speed. Some more specific details are given below on YOLO and on YOLO v4, which is the version that we used in this work, both in its original form and in its tiny counterpart.

#### 2.4.1. YOLO

Object detection algorithms aim at identifying the exact location of specific elements in images or videos. Recently, robotics, self-driving cars and automation technologies require accuracy as well as speed. Deformable component models (DPM) assess each filter applied to the picture at a different scale using a classifier. This method showed good accuracy performance [33]; it requires specific software and hardware optimization to analyze videos in real-time [34].

The anchor box [35] strategy was used to reduce the computational cost necessary to achieve classification, speeding up the detection. Using a CNN, YOLO divides the input image into a grid of cells and predicts a border box and the probability that it contains the target to be detected. One convolutional network is performed on the image by YOLO, and model confidence is used to threshold the resulting detections. YOLO combines detection and classification into the following loss function [17]

$$\begin{aligned} Loss = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2. \end{aligned} \tag{2}$$

where $(x, y)$ are the coordinates, $(w, h)$ are the width and height of the boundary boxes, $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$ are two parameters set to reduce the instability during training (caused by the high probability that a boundary box does not contain the target object), $1_i^{obj}$ represents the presence of the object in the cell $i$ (where the image is divided into a grid of $S \times S$ cells), $1_{ij}^{obj}$ denotes that the $j$th boundary box in the $i$th cell is responsible for that prediction ($B$ is the number of bounding boxes within each cell), $C_i$ denotes the confidence score of the box $j$ in the cell $i$, and $p_i$ refers to conditional class probabilities for class $c$ in cell $i$. This method enables the deep learning system to achieve outstanding accuracy and amazing classification speed [17].

#### 2.4.2. YOLO v4

The architecture of YOLO v4 [29] can be divided into three parts: head, neck, and backbone. Inside these three structural elements, a collection of new methods, called *the bag of freebies* and *the bag of specials*, were introduced. The bag of freebies is a set of methods that increase the training cost (e.g., data augmentation) but improves the learning of data by the model. The bag of specials contains different plugins and post-processing modules that only increase the inference cost by a small amount but can drastically improve the accuracy of the object detector, e.g., the Mish activation function. We trained YOLO v4 to

identify the view of the US scan (among long and short axis) and to detect the IVC in the US frames in our datasets.

#### 2.4.3. YOLO v4 Tiny

The YOLO v4 model is compacted into the YOLO v4 tiny [30]. Compared to the full model, in the tiny version there are fewer anchor boxes for prediction and only two YOLO layers as opposed to three. Furthermore, 29 convolutional layers are used instead of 137. The architecture is simplified and the number of parameters to be trained is decreased, which improves recognition time performance at the expense of accuracy. Usually, embedded and mobile devices employ this approach [36].

### *2.5. Performance Evaluation Methods*

To evaluate the performance of our object detection method, the primary metric used was the mean average precision ($mAP$). This was achieved by comparing the predicted area produced by the algorithm, represented by a rectangular region, to the ground truth using the intersection over union (IoU) metric:

$$IoU = \frac{\text{Area of intersection}}{\text{Area of union}}. \tag{3}$$

By setting a threshold of 50% for the IoU, we differentiated between True Positive (TP) and False Positive (FP) predictions based on whether the IoU was above or below the threshold, respectively. When the IVC was not detected, a False Negative (FN) was identified. Precision and recall were then calculated as:

$$p = \frac{TP}{TP + FP} \tag{4}$$

$$r = \frac{TP}{TP + FN} \tag{5}$$

The average precision ($AP$) was computed for each class as the area under the curve defined by the precision–recall relationship:

$$AP = \int_0^1 p(r)dr \tag{6}$$

The $mAP$ was then obtained as the average of the individual $AP$ values.
In addition, the F1 score was calculated to assess the detection performance:

$$F1_{score} = 2 \cdot \frac{r \cdot p}{r + p} \tag{7}$$

The performance in terms of processing time was measured as frames per second (FPS), calculated as:

$$FPS = \frac{\text{Number of frames}}{\text{Detection time}} \tag{8}$$

## 3. Results

### *3.1. Examples of Application to US Frames*

The YOLO networks trained on each of our three datasets were applied to classify the images into two categories, longitudinal or transverse view, and to detect the location of the IVC. An example of IVC detection is shown in Figure 2. The boundaries, green for the longitudinal view and violet for the transversal view, demarcate the regions where the vein is identified. Usually, the position of the vein is around the centre of the frame, but this is not always the case, as shown in the examples reported in Figure 3. YOLO networks examine each box in which the frame is split without having information on its location: thus, there is no bias in detecting the IVC in a specific location.

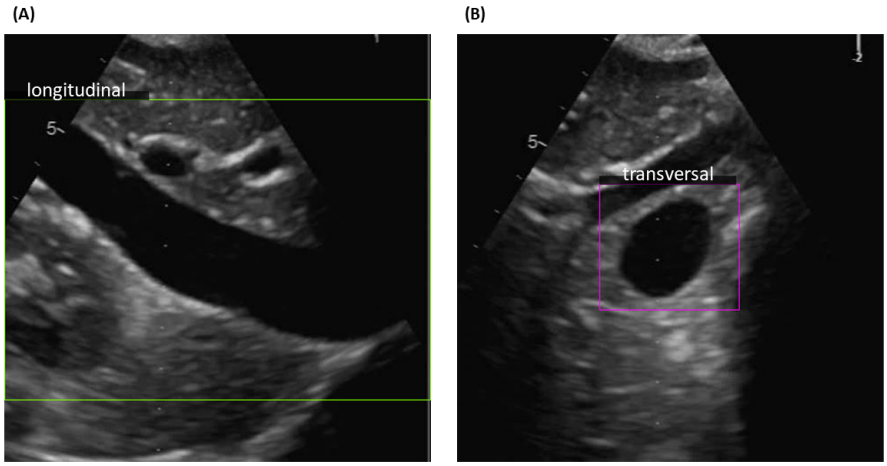


**Figure 2.** Examples of YOLO detection outputs. (**A**) Longitudinal and (**B**) Transversal view of IVC and their respective detected areas.

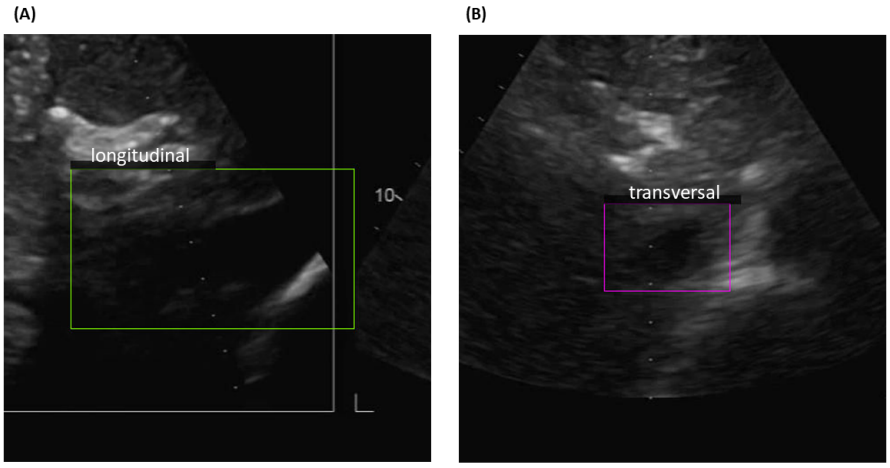


**Figure 3.** Examples of YOLO detection outputs in difficult cases, in which the IVC is not clearly visible. (**A**) Longitudinal and (**B**) Transversal view.

The three YOLO v4 (one for each dataset) and the three YOLO tiny models were trained respectively on the first and second hardware and software frameworks reported in Table 1. Then, the trained models were tested using both frameworks.

**Table 1.** Hardware and software specifications of the system used to train and test YOLO models.

| System | CPU | GPU | RAM | Dev. Language |
|---|---|---|---|---|
| PC | Intel (R) i3-9300F | Nvidia Q. P2200 | 32 GB | Python 3.9 |
| Colab | Intel(R) Xeon(R) | Nvidia Tesla K80 | 13 GB | Python 3.9.16 |

### *3.2. Comparison of Detection Performances of Different Architectures*

The mean accuracy of the YOLO v4 algorithm on the test set of D1 (showing IVC anatomy) reached 79.23%, with a mean IoU of 54.21%, which is 4% higher than the threshold that we have set to identify a TP. Comparing the mean accuracy values, reported in Table 2, the model trained with the second dataset (showing tissue movements) and the third dataset (including information on both anatomy and tissue movements) achieve a mean accuracy of 92.25% and 98.96%, respectively. The average IoU between the original and the two augmented datasets shows a discrete performance gain of about 10% and 24%, for D2 and D3, respectively.

**Table 2.** Performance evaluation parameters.

| | | IoU | Precision | Recall | F1 Score | mAP |
|---|---|---|---|---|---|---|
| YOLO v4 | D1 | 54.21% | 0.83 | 0.83 | 0.83 | 79.23% |
| | D2 | 64.81% | 0.90 | 0.90 | 0.90 | 92.25% |
| | D3 | 78.42% | 0.95 | 0.97 | 0.96 | 98.96% |
| YOLO v4 tiny | D1 | 70.22% | 0.89 | 0.91 | 0.90 | 92.54% |
| | D2 | 68.38% | 0.86 | 0.94 | 0.90 | 97.24% |
| | D3 | 79.53% | 0.94 | 0.97 | 0.95 | 98.77% |

The YOLO tiny version reached an impressive accuracy of 98.77% and an IoU of almost 80% on the test set of D3. Moreover, the detection performances were also fairly good on D1: in fact, surprisingly, they were superior with respect to those achieved by YOLO v4 model. Notice also that IoU obtained by the tiny model for D2 was inferior to the ones reached by the other two datasets.

Additional metrics are reported in Table 3.

The YOLO models had both to detect the boundary box in which the vein was contained and to identify the view provided by the US scan, i.e., to classify each frame as either longitudinal or transversal section. The performances for the identification of each class are shown in Table 3. There are no significant changes in the performance for the identification of the longitudinal view, comparing the model trained with dataset 1 and dataset 3. There is a slight decrease in average precision for the model trained with dataset 2 compared with the other two. In the case of the transversal section, there is an increasing trend of performances comparing the models trained with datasets 1, 2, and 3.

In longitudinal IVC detection, a performance improvement of roughly 9% between the D1 and the two augmented datasets is observed for the YOLO v4 tiny model. While there is an improvement in transversal detection performance between the three datasets, the gain is less than that of the longitudinal section.

**Table 3.** Identification of long and short axis view (i.e., longitudinal and transversal to the IVC).

| | | AP—YOLO v4 | AP—YOLO v4 Tiny |
|---|---|---|---|
| Longitudinal | D1 | 78.27% | 89.33% |
| | D2 | 86.79% | 97.11% |
| | D3 | 99.13% | 98.77% |
| Transversal | D1 | 80.19% | 95.75% |
| | D2 | 97.70% | 97.37% |
| | D3 | 98.79% | 98.78% |

### *3.3. Time Performance*

The inference times of the two YOLO architectures (YOLO v4 and tiny) trained on the three datasets were computed running them on both hardware listed in Table 1. Specifically, the inference time using YOLO v4 and the first hardware listed in Table 1 (PC) was 198 s. The frame rate of the analysis was 12.64 FPS. Instead, the YOLO v4 tiny, with the same hardware, took 50 s (50.2 FPS) to complete the analysis.

The performances when using Google Colab (the second hardware shown in Table 1) were better: the inference time of the entire test set was 51 s (with a frame rate of 49.2 FPS) when using YOLO v4; the test with the tiny version of YOLO lasted 13 s (the frame rate is 193.3 FPS). All test sets (i.e., D1, D2 and D3) had equivalent inference time.

### *3.4. K-Fold Cross-Validation*

After the preliminary tests on the entire dataset, we selected for further validation the YOLO tiny version, because of the great vein detection performance reported above and the impressive classification speed. A k-fold cross-validation (with $k = 3$) was considered,

splitting data in order to extract frames for training from different subject than those used for test. The cross-validation was performed on D1, D2 and D3. The classification performances are reported in Table 4. Additional information on vein identification for the single view are given in Table 5.

**Table 4.** K-fold performance evaluation parameters.

| | | IoU | Precision | Recall | F1 Score | mAP |
|---|---|---|---|---|---|---|
| YOLO v4 tiny | D1 | 65.74% | 0.86 | 0.86 | 0.86 | 88.67% |
| | D2 | 64.55% | 0.82 | 0.87 | 0.85 | 92.07% |
| | D3 | 68.97% | 0.86 | 0.92 | 0.89 | 95.04% |

**Table 5.** K-fold performance in the identification of long and short axis view (i.e., longitudinal and transversal to the IVC).

| | | AP—YOLO v4 Tiny |
|---|---|---|
| Longitudinal | D1 | 91.71% |
| | D2 | 90.06% |
| | D3 | 93.55% |
| Transversal | D1 | 85.64% |
| | D2 | 94.08% |
| | D3 | 96.52% |

As reported in the previous section, the same trend of the mAP was also visible when applying the cross-validation. In fact, the accuracy increased by about 7% between D1 and D3. The accuracy of each class showed the same trend as reported above.

## 4. Discussion

Reducing operator dependency is very important in US imaging. Our group introduced different methods to process US videos of IVC [13,14] supporting standardization and improving repeatability of measurements [37]. Some preliminary applications have been obtained [7–9,38,39]. However, our algorithms require a preliminary interaction with the user, so that our techniques can be considered semi-automated and results still depend on the operator's settings. Moreover, if the operator loses sight of the vein during an acquisition, the measurement should stop and the process should come back to the beginning.

In this paper, we focus on automatizing the detection of the IVC, in order to free the operator from the preliminary procedure and to allow the possibility of IVC recovering in case of problems. Moreover, the either long or short axis view is identified, thus allowing the operator to even change it during the acquisition (by rotating the probe of 90°), without needing to stop the inspection.

We trained different deep learning object detection models, based on YOLO architecture, to identify the IVC in a US video. This is aimed at replacing the first user interaction in our previous segmentation software. Thus, we assume that the operator is interested on IVC detection and is able to point the probe toward the correct tissues: the behavior of our software when applied to US scans of other tissues was not tested, as it is not of interest.

As reported in the Results section, we achieved fairly good IVC detection performance. This work paves the way for a new fully automated IVC detection and edge tracking tool, once integrated with algorithms that delineate the vein either in long or in short axis view. We considered both single frames (showing tissue anatomy) and other two augmented datasets, to highlight different characteristics of IVC. Specifically, in the first dataset, the features that the CNN layers can extract from the images are mostly correlated with the anatomy of the tissues and the geometry of the vein. The second dataset highlights the movements of the tissues, thus providing information on compliance and flexibility of

the IVC walls, related to the collapsibility of the vein, together with the respirophasic movement. The last dataset includes a combination of the previous vein characteristics: in fact the geometry of the vein is visible, but at the same time there is additional information about its collapsibility and movement.

### *4.1. Comparison of Detection Performance*

Analyzing the result, we can assume that the additional information extracted from augmented datasets gives both YOLO algorithms additional knowledge to identify the IVC. Indeed, IVC shows large respirophasic motions and wall deformations, so augmenting the information considering tissue movements is reasonably important. In particular, for the YOLO v4 model there is a great performance gain in the identification of IVC in both longitudinal and transverse views: in fact, the average precision increases by 20% comparing the D1 dataset and the augmented dataset D3 (including both information on IVC anatomy and movements). The mean IoU increases by around 24% when comparing outcomes when using either D1 or D3. The model trained with D2 reached intermediate performances with respect to those based on D1 and D3, both in terms of IoU and mAP.

The detection performance of YOLO v4 tiny follows the trend of the YOLO v4 model, with an increase of mAP and IoU over the datasets. Even though the two models had different architectural designs, their performances over D3 were very comparable. However, the performances of the tiny version were generally superior, suggesting the full YOLO v4 model was too complicated (and sometimes overfitted the training data). As YOLO tiny showed equivalent performances and much lower computational cost in the comparison with YOLO v4, we selected it for further validation. Specifically, we used a k-fold cross-validation test, in which the subjects considered for the training were not used for the test. This is an application more similar to the natural conditions of use of our algorithm. The results confirm the accuracy improvement between the dataset D1 and D3. Moreover, the performances are fairly good and quite similar to those obtained by the first investigation, in which frames from the same subjects were used for training and test.

In general, the best performances were obtained with dataset 3: this means that the combination of information on IVC anatomy/geometry and movements/border displacement over time allows the model to identify better the region where the vein is located.

### *4.2. Time Performance*

Our research team has developed segmentation algorithms [12–14] that are designed to analyze US videos in real-time with low computational cost, once they are optimized and compiled (which is currently underway). Hence, the IVC detection algorithm also needs to be speedy, in order to be integrated in a tool feasible for applications.

The frame rate of US imaging is determined by the duration required to scan the area of interest. This scanning time, in turn, is influenced by three factors: the depth, line density, and pulse rate of each scan line [40]. Typically, we use US videos that are captured at a frame rate of around 30 FPS. The frame rate analysis by our YOLO models is promising: it can process videos in real-time, with minimal delay. For example, the YOLO tiny could provide the output with a delay which is less than the sampling period with both the hardware architectures that we tested (indicated in Table 1). When incorporating a pipeline that uses two frames to capture IVC dynamics, an additional delay is introduced due to the augmented frames being sampled with a 10-pitch (which corresponds to one third of a second, with a frame rate of 30 Hz); after this delay, the estimation provided by our YOLO model can be real time for all subsequent frames. Notice that by using a circular buffer keeping the last 10 frames available, this delay is removed in applications such as IVC retrieval, which is useful when the operator loses sight of the vein due to a fast movement or an artefact. Thus, we are confident that our detection method could be valuable for future integration into a fully automated IVC tracking software. Still considering the good performances and low computational cost of YOLO v4 tiny, apps for embedded and mobile devices could also be feasible.

## 5. Conclusions

A real-time IVC detection algorithm for US videos is introduced using YOLO v4 network. The model performance was evaluated on a vast dataset of manually segmented US images, and two augmented datasets were created to provide additional information to our detection algorithm. Our findings confirm that incorporating information on IVC movements is crucial to enhance detection performance. The performances of YOLO v4 algorithm and of its tiny version were very similar in preliminary tests on the entire dataset, thus leaning toward preferring the latter, which has a much lower computational cost. The YOLO tiny was then further investigated, by a cross-validation which included different subjects for training and test, obtaining again good performances.

In the future, we plan to integrate IVC detection with our edge tracking algorithms to create fully automated methods. The automation of IVC detection, tracking, and segmentation can reduce the subjectivity of the interpretation of US scans. Additionally, these tools will assist novices, paramedical staff, and nursing in acquiring good US videos and making useful assessments of IVC dynamics.

**Author Contributions:** Conceptualization, P.P., G.C. and L.M.; methodology, P.P. and G.C.; software, P.P., G.C. and F.P.; validation, P.P. and F.P.; formal analysis, P.P. and G.C.; investigation, P.P., F.P. and G.C.; data curation, L.E., S.C., S.A., G.M. and S.R.; writing—original draft preparation, P.P. and L.M.; writing—review and editing, L.M. and G.C.; visualization, P.P.; supervision, L.M., G.M. and S.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research did not receive any specific grants from any funding agency in the public, commercial, or not-for-profit sectors.

**Informed Consent Statement:** The Ethics Committee of Mauriziano Hospital 76 (Turin) approval number 388/2020 (experimental protocol identifier v.1, 1 June 2020) was obtained. Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We thank William Bassolino, Emiliano Gallo, Letizia Cantore, Carlotta Broglia for the manual segmentation and data cleaning of the data sets. We thank Angelica Pulino to start bibliographic study. The contribution of the team of VIPER s.r.l. was greatly appreciated.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CVC | central venous catheter |
| IVC | inferior vena cava |
| CNN | Convolutional neural networks |
| YOLO | You only look once |
| US | Ultrasound |
| DPM | Deformable part model |
| AP | Average precision |
| MAP | Mean average precision |
| IoU | Intersect over union |
| TP | True positive |
| TN | True negative |
| FN | False negative |
| FPS | Frame per second |

## References

1. Van der Mullen, J.; Wise, R.; Vermeulen, G.; Moonen, P.J.; Malbrain, M.L.N.G. Assessment of hypovolaemia in the critically ill. *Anaesthesiol. Intensive Ther.* **2018**, *50*, 141–149. [CrossRef] [PubMed]
2. Hansen, B. Fluid Overload. *Front. Vet. Sci.* **2021**, *8*, 668668. [CrossRef] [PubMed]

3. Kalantari, K.; Chang, J.N.; Ronco, C.; Rosner, M.H. Assessment of intravascular volume status and volume responsiveness in critically ill patients. *Kidney Int.* **2013**, *83*, 1017–1028. [CrossRef] [PubMed]
4. Shah, M.R.; Hasselblad, V.; Stevenson, L. Impact of the Pulmonary Artery Catheter in Critically Ill Patients: Meta-analysis of Randomized Clinical Trials. *JAMA* **2005**, *294*, 1664–1670. [CrossRef] [PubMed]
5. Eber, J.; Villaseñor, C. Ultrasound: Advantages, disadvantages, and controversies. *Nurse Pract. Forum* **1991**, *2*, 239–242.
6. Piotrkowski; J; Buda, N; Januszko-Giergielewicz, B; Kosiak, W. Use of bedside ultrasound to assess fluid status: A literature review. *Pol. Arch. Intern. Med.* **2019**, *129*, 692–699. [CrossRef]
7. Mesin, L.; Roatta, S.; Pasquero, P.; Porta, M. Automated Volume Status Assessment Using Inferior Vena Cava Pulsatility. *Electronics* **2020**, *9*, 1671. [CrossRef]
8. Capomolla, S.; Febo, O.; Caporotondi, A.; Guazzotti, G.; Gnemmi, M.; Rossi, A.; Pinna, G.; Maestri, R.; Cobelli, F. Non-invasive estimation of right atrial pressure by combined Doppler echocardiographic measurements of the inferior vena cava in patients with congestive heart failure. *Ital. Heart J. Off. J. Ital. Fed. Cardiol.* **2000**, *1*, 684–690.
9. Albani, S.; Pinamonti, B.; Giovinazzo, T.; de Scordilli, M.; Fabris, E.; Stolfo, D.; Perkan, A.; Gregorio, C.; Barbati, G.; Geri, P.; et al. Accuracy of right atrial pressure estimation using a multi-parameter approach derived from inferior vena cava semi-automated edge-tracking echocardiography: A pilot study in patients with cardiovascular disorders. *Int. J. Cardiovasc. Imaging.* **2020**, *36*, 1213–1225. [CrossRef]
10. Wallace, D.; Allison, M.; Stone, M. Inferior vena cava percentage collapse during respiration is affected by the sampling location: An ultrasound study in healthy volunteers. *Acad. Emerg. Med.* **2010**, *17*, 96–99. [CrossRef]
11. Krupa, A.; Fichtinger, G.; Hager, G. D. Full Motion Tracking in Ultrasound Using Image Speckle Information and Visual Servoing. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 2458–2464. [CrossRef]
12. Mesin, L.; Pasquero, P.; Albani, S.; Porta, M.; Roatta, S. Semi-automated tracking and continuous monitoring of inferior vena cava diameter in simulated and experimental ultrasound imaging. *Ultrasound Med. Biol.* **2015**, *41*, 845–857. [CrossRef]
13. Mesin, L.; Pasquero, P.; Roatta, S. Tracking and Monitoring Pulsatility of a Portion of Inferior Vena Cava from Ultrasound Imaging in Long Axis. *Ultrasound Med. Biol.* **2019**, *45*, 1338–1343. [CrossRef]
14. Mesin, L.; Pasquero, P.; Roatta, S. Multi-directional assessment of Respiratory and Cardiac Pulsatility of the Inferior Vena Cava from Ultrasound Imaging in Short Axis. *Ultrasound Med. Biol.* **2020**, *46*, 3475–3482. [CrossRef]
15. Chudasama, V.; Kar, P.; Gudmalwar, A.; Shah, N.; Wasnik, P.; Onoe, N. M2FNet: Multi-modal Fusion Network for Emotion Recognition in Conversation. *arXiv* **2022**, arXiv:2206.02187.
16. Hironobu, F.; Tsubasa, H.; Takayoshi, Y. Deep learning-based image recognition for autonomous driving. *IATSS Res.* **2019**, *43*, 244–252. [CrossRef]
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
18. Sahiner, B.; Chan, H.; Petrick, N.; Wei, D.; Helvie, M.A.; Adler, D.D.; Goodsitt, M.M. Classification of mass and normal breast tissue: A convolution neural network classifier with spatial domain and texture images. *IEEE Trans. Med. Imaging* **1996**, *15*, 598–610. [CrossRef]
19. Park, S.; Hwang, J.; Kwak, N. 3D Human Pose Estimation Using Convolutional Neural Networks with 2D Pose Information. In *Computer Vision—ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science*; Hua, G., Jégou, H., Eds.; Springer: Cham, Switzerland, 2016; Volume 9915. [CrossRef]
20. Ponzio, F.; Macii, E.; Ficarra, E.; Di Cataldo, S. A Multi-modal Brain Image Registration Framework for US-guided Neuronavigation Systems. In Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies—BIOIMAGING, (BIOSTEC 2017), Porto, Portugal, 21–23 February 2017; pp. 114–121. [CrossRef]
21. Aijaz, A.R.; Furqan, R.; Arif, M.; Abdulaziz, A.; Ziyad, A.; Ajaz A.; Hessa, A.; Gyu Sang, C. An Efficient CNN Model for COVID-19 Disease Detection Based on X-Ray Image Classification. *Complexity* **2021**, *2021*, 6621607. [CrossRef]
22. Michalski, P.; Ruszczak, B.; Tomaszewski, M. Convolutional Neural Networks Implementations for Computer Vision. In *Biomedical Engineering and Neuroscience. BCI 2018. Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2018; Volume 720. [CrossRef]
23. Blaivas, M.; Blaivas, L.; Philips, G.; Merchant, R.; Levy, M.; Abbasi, A.; Eickhoff, C.; Shapiro, N.; Corl, K. Development of a Deep Learning Network to Classify Inferior Vena Cava Collapse to Predict Fluid Responsiveness. *J. Ultrasound Med.* **2021**, *40*, 1495–1504. [CrossRef]
24. Ni, J.C.; Shpanskaya, K.; Han, M.; Lee, E.H.; Do, B.H.; Kuo, W.T.; Yeom, K.W.; Wang, D.S. Deep Learning for Automated Classification of Inferior Vena Cava Filter Types on Radiographs. *J. Vasc. Interv. Radiol.* **2020**, *31*, 66–73. [CrossRef]
25. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv* **2013**, arXiv:1311.2524.
26. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
27. Wu, X.; Tan, G. Zhu, N.; Chen, Z.; Yang, Y.; Wen H.; Li, K. CacheTrack-YOLO: Real-Time Detection and Tracking for Thyroid Nodules and Surrounding Tissues in Ultrasound Videos. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 3812–3823. [CrossRef] [PubMed]

28. Mesin, L.; Albani, S.; Policastro, P.; Pasquero, P.; Porta, M.; Melchiorri, C.; Leonardi, G.; Albera, C.; Scacciatella, P.; Pellicori, P.; et al. Assessment of Phasic Changes of Vascular Size by Automated Edge Tracking-State of the Art and Clinical Perspectives. *Front. Cardiovasc. Med.* **2022**, *8*, 775635. [CrossRef] [PubMed]
29. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
30. Wang, C.; Bochkovskiy, A.; Liao, H. Scaled-YOLOv4: Scaling Cross Stage Partial Network. *arXiv* **2020**, arxiv:2011.08036.
31. Fairbrother, J.; Letchford, A.N. Projection Results for the k-Partition Problem. *Discret. Optim.* **2017**, *26*, 97–111. [CrossRef]
32. Shaifee, M.J.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video. *J. Comput. Vis. Imaging Syst.* **2017**, *3*. [CrossRef]
33. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8. [CrossRef]
34. Yan, J.; Lei, Z.; Wen, L.; Li, S. Z. The Fastest Deformable Part Model for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2497–2504. [CrossRef]
35. Lempitsky, V.; Kohli, P.; Rother, C.; Sharp, T. Image segmentation with a bounding box prior. In Proceedings of the IEEE IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 277–284. [CrossRef]
36. Wang, G.; Ding, H.; Li, B.; Nie, R.; Zhao, Y. Trident-YOLO: Improving the precision and speed of mobile device object detection. *IET Image Process.* **2022**, *16*, 145–157. [CrossRef]
37. Mesin, L.; Giovinazzo, T.; D'Alessandro, S.; Roatta, S.; Raviolo, A.; Chiacchiarini, F.; Porta, M.; Pasquero, P. Improved repeatability of the estimation of pulsatility of inferior vena cava. *Ultrasound Med. Biol.* **2019**, *45*, 2830–2843. [CrossRef]
38. Ermini, L.; Seddone, S.; Policastro, P.; Mesin, L.; Pasquero, P.; Roatta, S. The Cardiac Caval Index: Improving Noninvasive Assessment of Cardiac Preload. *J. Ultrasound Med.* **2022**, *41*, 2247–2258. [CrossRef]
39. Mesin, L.; Policastro, P.; Albani, S.; Petersen, C.; Sciarrone, P.; Taddei, C.; Giannoni, A. Non-Invasive Estimation of Right Atrial Pressure Using a Semi-Automated Echocardiographic Tool for Inferior Vena Cava Edge-Tracking. *J. Clin. Med.* **2022**, *11*, 3257. [CrossRef]
40. Ng, A.; Swanevelder, J. Resolution in ultrasound imaging. *Contin. Educ. Anaesth. Crit. Care Pain* **2011**, *11*, 186–192. [CrossRef]