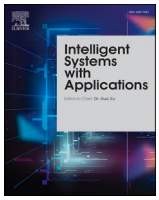
Unmasking Anomalies in Road-Scene Segmentation

Intelligent Systems with Applications 17 (2023) 200174

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications



Fighting for a future free from violence: A framework for real-time detection of "Signal for Help"

Sarah Azimi *, Corrado De Sio, Francesco Carlucci, Luca Sterpone

Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy

ARTICLE INFO

Keywords:
Signal for help
Gesture recognition
Convolutional neural networks

ABSTRACT

In April 2020, by the start of isolation all around the world to counter the spread of COVID-19, an increase in violence against women and kids has been observed such that it has been named The Shadow Pandemic. To fight against this phenomenon, a Canadian foundation proposed the "Signal for Help" gesture to help people in danger to alert others of being in danger, discreetly. Soon, this gesture became famous among people all around the world, and even after COVID-19 isolation, it has been used in public places to alert them of being in danger and abused. However, the problem is that the signal works if people recognize it and know what it means. To address this challenge, we present a workflow for real-time detection of "Signal for Help" based on two lightweight CNN architectures, dedicated to hand palm detection and hand gesture classification, respectively. Moreover, due to the lack of a "Signal for Help" dataset, we create the first video dataset representing the "Signal for Help" hand gesture for detection and classification applications which includes 200 videos. While the hand-detection task is based on a pre-trained network, the classifying network is trained using the publicly available Jesture dataset, including 27 classes, and fine-tuned with the "Signal for Help" dataset through transfer learning. The proposed platform shows an accuracy of 91.25% with a video processing capability of 16 fps executed on a machine with an Intel i9-9900K@3.6 GHz CPU, 31.2 GB memory, and NVIDIA GeForce RTX 2080 Ti GPU, while it reaches 6 fps when running on Jetson Nano NVIDIA developer kit as an embedded platform. The high performance and small model size of the proposed approach ensure great suitability for resource-limited devices and embedded applications which has been confirmed by implementing the developed framework on the Jetson Nano Developer Kit. A comparison between the developed framework and the state-of-the-art hand detection and classification models shows a negligible reduction in the validation accuracy, around 3%, while the proposed model required 4 times fewer resources for implementation, and inference has a speedup of about 50% on Jetson Nano platform, which make it highly suitable for embedded systems. The developed platform as well as the created dataset are publicly available.

1. Introduction

The story of a 16-year-old girl from North Carolina reported missing on November 22, 2021, has shined a spotlight on a hand signal. Two days after her parents reported the teenager missing, a driver noticed a girl in a passing car making hand gestures of "Signal for Help". The police were able to arrest the man who had kidnapped the girl, and fortunately, the girl returned safely to her family. The gesture used by this teenager was created by the Canadian Women's foundation as a "Signal for Help" to alert others of being in danger. However, this solution works only if it is observed and recognized by human-being present at moment. Therefore, many times it might happen that a person in danger is trying to ask for help but is not seen by anyone.

This works aims to take a step forward to find a solution for this problem, by proposing a platform for real-time automatic detection of "Signal for Help", easily to be exploited by Closed Circuit Television (CCTV) and security videos in public places, such as public transport, supermarket, or even roads.

Many researchers focused on the real-time automatic detection of violence through surveillance videos, relying on computer vision techniques for the detection of violence-related features, have shown promising results (Ramzan et al., 2019). However, considering the

* Corresponidng author.
E-mail addresses: sarah.azimi@polito.it (S. Azimi), corrado.desio@polito.it (C. De Sio), s304910@studenti.polito.it (F. Carlucci), luca.sterpone@polito.it (L. Sterpone).

https://doi.org/10.1016/j.iswa.2022.200174
Received 1 November 2022; Received in revised form 7 December 2022; Accepted 29 December 2022
Available online 8 January 2023

detection of "Signal for Help", such methods that are based on the detection of characteristics of scenarios involved in violence (e.g., blood or gun) are not valid. Indeed, the detection and classification of "Signal for Help" is closer to the platforms that are dedicated to vision-based hand gesture recognition.

Currently, Convolutional Neural Networks (CNNs) are the state-of-the-art solution for image-based and video-based hand gesture recognition. However, when considering real-time gesture recognition platforms, it is important to fulfill requirements such as classification accuracy, recognition time, and scalability. Moreover, in developing a recognition algorithm for an embedded system, resource efficiency becomes another important aspect to take into account. Therefore, most of the previous works dedicated to hand gesture recognition exploiting CNN are not suitable for real-time detection scenarios since they are focused on increasing the offline classification accuracy while other methods do not apply to embedded systems and applications due to the high resource usage.

Considering CNN platforms, another important aspect to investigate is the availability of a suitable dataset. Typically, for having results with high accuracy, a vast dataset is required. For example, many works that are dedicated to hand gesture recognition exploit vast datasets such as EgoGesture Dataset (Zhang et al., 2018) and NVIDIA Dynamic Hand Gesture Dataset (Molchanov et al., 2016). Though, regarding the "Signal for Help", due to the recent creation of this signal, not even one dataset is available yet and, there are not many videos available online for creating a dataset exploiting already existing videos.

In this work, we propose a CNN-based lightweight, fast and accurate platform, suitable for embedded systems with low available resources, for real-time detection of "Signal for Help". This study generated a dataset of "Signal for Help" gesture videos which are used for training as well as validating and testing the developed CNN-based recognition platform.

1.1. The main contributions

The main contributions of this work are divided into two parts as follows:

- The first part is oriented toward the creation of the very first open-source dynamic "Signal for Help" video dataset thanks to the active contribution of the students of Politecnico di Torino. The dataset is publicly available for everyone's usage.
- The second part is dedicated to the development of the platform which is able to perform real-time "Signal for Help" hand gestures through surveillance videos. In order to propose a model appropriate to real-world scenarios, efforts have been made for designing a system that works in real-time conditions by evaluating a video stream and is suitable for systems with low computational resources, such as embedded and video surveillance distributed systems. To do so, we exploit two CNN architectures, namely MediaPipe and MobileNet, to take advantage of their small models and good performance for hand gesture recognition, then develop a single, high-performance, and lightweight hierarchical recognition architecture. We would like to highlight that even though the CNN-based platform for the detection and recognition of hand gestures is widely studied, this work is oriented over the very first developed platform for real-time detection of "Signal for Help". Therefore, proposing a novel and more importantly necessary application for hand gesture detection and recognition techniques. We truly hope that we can make a change, even though a small one, toward a safer city.

The paper is organized as follows: Section 2 provides an overview of the related works. Section 3 is dedicated to elaborating on the importance of fighting against violence, especially after the COVID-19 pandemic. Section 4 reports the methodology for creating the first video dataset of "Signal for Help". Section 5 describes the developed "Signal for Help" recognition platform, while Section 6 is dedicated to the experiment performed on the developed platform. Finally, conclusions and future works are drawn in Section 7.

2. Related works

This section is providing an overview of the previous strategies and methods developed for the recognition of emotion in particular violence as well as the recognition of different hand gestures through static images and videos.

Emotion is known as one of the most fundamental methods of human communication with a significant impact on our life. Therefore, many research works are focused on the development of techniques for the automatic recognition of different emotions (Kiruthiga & Rajavel, 2021). In particular, detecting violence through videos that involve techniques such as computer visions for objects and motion detection and classification has gained high interest.

Due to the complexity and diversity of violence patterns, many approaches have been developed (Fu et al., 2019). The work in Fu et al. (2016) proposes a model to detect fights based on extracting motion acceleration, and motion region while achieving 78% accuracy on their custom dataset. In Sudhakaran and Lanz (2017), the authors proposed a model consisting of a series of convolutional layers followed by a max pooling operation for extracting discriminable features and consequently, classifying the videos as violent and non-violent ones, resulting in very high accuracy, around 97%. In Chen et al. (2011), the authors focused on the detection of violence by finding regions with skin and blood and analyzing these regions for fast motions. As can be seen, most of the research works focused on detecting violent scenes by recognizing some violence-related characteristics such as blood, gunshot, explosion, and so on (Chen et al., 2011). However, it is a challenging task to define effective and discriminative features representing violence due to the variation of the human body and the type of violence. Additionally, methods led to a high rate of false detection.

Considering our case which is using "Signal for Help" hand gesture discretely, such methods that act on detecting violence by typical characteristics are not valid and we need to move toward platforms that perform hand gesture detection and recognition.

Hand gestures as a nonverbal communication way are considered to be an essential aspect of Human-Machine Interaction (HMI). Recently, many research works have been dedicated to hand gesture recognition (Agrawal & Gupta, 2016; Tan et al., 2021; Wahid et al., 2018), mainly aiming at tasks such as gesture control within a vehicle, recognition of signals from sports referees (Žemgulys et al., 2018), and sign languages (Smith et al., 2018; Ashwini et al., 2020). These tasks have been performed by exploiting CNN which results in great success (Wang et al., 2016; Simonyan & Zisserman, 2014; Athilakshmi et al., 2018). While detection of static gestures in which the hand performs static poses, without any movements, has been investigated by many researchers (Elliott et al., 2021[1]), dynamic hand gesture recognition where changes in fingers shape and flexure angles modify the status of the hand has gained less attention.

In Athilakshmi et al. (2018), the Support Vector Machine algorithm has been investigated for the detection of basketball referee signal based on static images dataset with the achieved accuracy of 97.5% while a CNN is used in Tan et al. (2021b) to classify the American Sign Language (ASL) obtaining average accuracy of 98.5%, trained on different static image datasets (Rahman et al., 2019). Due to the great success of CNN architectures for the detection of static hand gestures, there has been a growing trend to apply them also to the detection of dynamic hand gestures (Tran et al., 2015), exploiting 3D convolution and 3D pooling, taking a sequence of video frames as inputs.

[1] https://github.com/Polito-Reconfigurable-Computing/SFH/tree/main/dataset.

Fig. 1. The "Signal for Help" hand gesture.

While several works address detection and classification separately (Molchanov et al., 2015; Ohn-Bar & Trivedi, 2014), real-time hand gesture recognition requires simultaneous application of detection and classification to the continuous stream of videos. In Köpüklü et al. (2019), the authors proposed architecture consists of two CNN models, ResNet as a hand detector and ResNext as an offline working classifier, evaluating the accuracy of the architecture on two publicly available datasets, EgoGesture and NVIDIA Dynamic Hand Gesture Datasets, which shows 94.04% of classification accuracy in offline mode. Even though the proposed architecture obtained a high accuracy rate, it is not efficient for being implemented on an embedded system with limited resources. Moreover, the efficiency of the platform is evaluated on the datasets which are oriented on a single hand at a close distance with a camera with a plane background which is not matched with the purpose of our method.

In our work, we propose a hierarchical architecture, composed of two lightweight CNN models, suitable for embedded applications with low available resources and low power requirements. It is the first architecture dedicated to the real-time video-based recognition of "Signal for Help". Moreover, we created the very first video-based dataset for "Signal for Help" in which we focused on including videos covering different possible scenarios, with the plane and not plane background, while the hand is located in the close and far distance (maximum 4 m) from the camera recording the video.

3. Violence during COVID 19

This section is elaborating on the occurrence of violence during COVID-19 which led to the creation of the "Signal for Help" hand gesture.

One out of three women worldwide experience either physical, sexual, or emotional violence. Since the epidemic of COVID-19, reports show that the rate of violence against women and girls has increased drastically, a phenomenon that has been named The Shadow Pandemic.

Due to the social isolation resulting from the COVID-19 pandemic, it became difficult for those who are at risk of abuse or violence to safely reach out for help. To counter this problem, the Canadian Women's Foundation thought of a novel solution, the use of a hand gesture named Signal for Help.

The signal is performed by holding your hand up with your thumb tucked into your palm, then folding your fingers down, symbolically trapping your thumb in your fingers, as represented in Fig. 1. It was originally created as a tool to combat the global rise in domestic violence cases resulting from isolation related to the COVID-19 pandemic. Later on, the signal became popular on TikTok and Twitter, with millions of audience members. Even though the isolation due to COVID-19 is fading

Fig. 2. Examples of "Signal for Help" dataset videos representing the heterogeneity of the dataset: (a) single hand with dark natural light and plane background (b) single hand with bright natural light and urban background (c) single hand with dark artificial light rotating 90° (d) single hand with bright artificial light rotating 0° (e) multiple hands with plane background with distance close to the camera (f) multiple hands with plane background with maximum distance from the camera (g) multiple hands with urban background (h) single hand with an urban background.

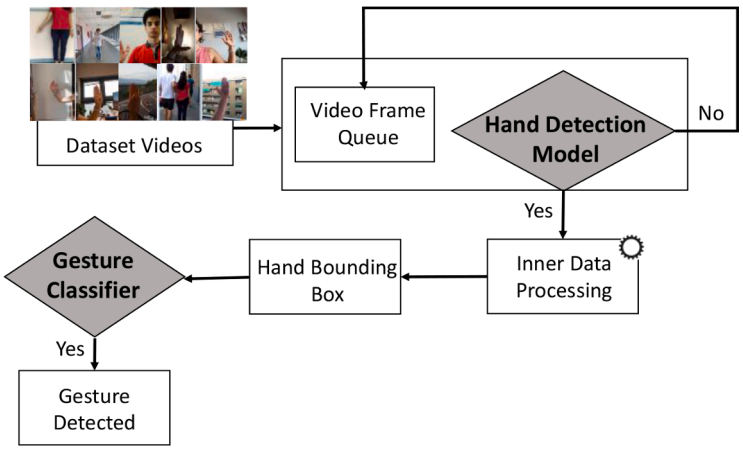


Fig. 3. The developed workflow for detection and classification of "Signal for Help".

out, unfortunately, violence against women, domestic or even on streets and public spaces continues to occur and the use of Signal for Help is becoming a valuable tool to fight against the rise of violence. The problem is that the signal works if people see it or are aware of its meaning, which is not common since most of the time the signal is transmitted discretely, without drawing attention. Therefore, we would like to aim to solve this problem, by developing a platform that is suitable for exploiting the vast amount of surveillance videos recorded by surveillance cameras, available in every corner of smart cities to perform real-time automatic detection of "Signal for Help" to make it possible to inform the right authorities and reach out for help.

4. "Signal for Help" dataset creation

The steps for the creation of the very first "Signal for Help" video dataset and the features of the created dataset are detailed in this section.

Several hand gesture datasets had been created mostly for being applied to hand gesture recognition applications. However, regarding the "Signal for Help", since this hand gesture is created recently, no database is yet available. We aimed to create the first database of "Signal for Help" hand gestures thanks to the contribution of the students of Politecnico di Torino. The creation of the dataset has been performed in three phases: Mining, Validation and Filtration.

Firstly, in the Mining phase for collecting "Signal for Help" gesture videos, we prepared a call for collaboration in which we asked students to send us videos consisting of the "Signal for Help" gesture. During the collection of the videos, we have asked for variations of the following parameters: (a) distance from the camera from 0 to 5 to 4 m (b) lightening and background conditions (c) number of hands present in the video. We started with collecting simple scenario videos, with the presence of one hand executing the "Signal for Help" gesture, close distance to the camera with the plane lightning background. Then, we moved toward more complex videos with the presence of more than one person, with a maximum distance of 4 m from the camera in the urban background which is the scenario closest to what is expected in real cases. Fig. 2 shows a representative sample of the collected videos including different scenarios.

Secondly, we moved to the validation phase, to select the videos which are executed correctly at the mining stage. We identified the videos with the wrong gesture, videos with a hand executing a feature but not completely visible in the frame, and videos with the face visible in the video while the lighting conditions and distance from the camera were not considered since it is not feasible accurately evaluate these features.

In the final phase, we performed the filtration of the inappropriate videos distinguished in the validation stage by removing the videos from the dataset. Eventually, around 250 videos were received during the mining phase which led to the current "Signal for Help" dataset, including about 200 videos consisting of "Signal for Help" hand gestures. The dataset is collected indoors as well as outdoors with considerable variation in lighting, including artificial and natural light. The created "Signal for Help" dataset is available online[1].

5. The developed "Signal for Help" recognition platform

This section is oriented over the detailed elaboration of the developed platform for real-time recognition of the "Signal for Help" gesture, the implemented hand detection and gesture recognition architecture as well as the training steps of these algorithms.

The goal of this paper is oriented toward the development of a computing platform for real-time "Signal for Help" hand gesture recognition suitable for being integrated with surveillance videos. We attempt to address this challenge using a light deep learning-based model, exploring the innovative design of two architectures of MediaPipe and MobileNet for hand gesture detection and hand gesture classification. The developed platform is open-source and available online.[2]

5.1. The implemented hardware/software platform

The current section elaborates on the implemented platform for recognition of "Signal for Help".

The developed platform acts in three steps: *hand detection, inner data processing*, and gesture classification. A conceptual schema is illustrated in Fig. 3. Please consider that the input of the platform is a 3-channel video.

The hand detection step is fully implemented using the MediaPipe framework. MediaPipe is used for checking the existence of hands in the frames of the video and if one or more hands are detected, they are provided to the next step. The MediaPipe output has been extended to provide a binary output to notify the hand detection to be used for enabling the second step of the platform. The frames where a hand is detected are manipulated by the inner data processing phase to make them suitable to be used as input for the gesture classification step. In particular, the input of the third step is a queue of time-ordered and processed video frames. The classifier, based on a MobileNet network, checks the existence of the "Signal for Help" hand gesture and labels the detected hand accordingly.

[2] https://github.com/Polito-Reconfigurable-Computing/SFH/tree/main/src.

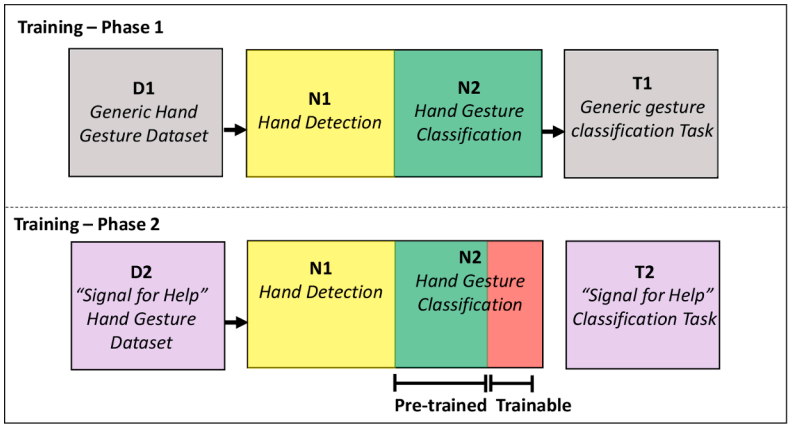


Fig. 4. The training of flow of the "Signal for Help" Recognition.

Table 1
The Training Parameters for MobileNet with Jesture used for training and the "Signal for Help" Dataset used for fine tunning.

Parameters	Jesture (Training)	Signal for Help (Fine Tuning)
Dataset Size	148,092	200
Classes of the Dataset [#]	27	2
Epochs [#]	39	39
Batch Size	40	18
Dampening	0.9	0.9
Weight Decay	0.01	0.01
Training Time [h]	26	2
Accuracy [%]	91.71	91.25

5.2. Hand detection model

The first step of the framework is dedicated to the detection of hands in the videos exploiting the MediaPipe Hand platform. MediaPipe is an open-source framework developed by Google, originally developed for real-time analysis of videos and audio on YouTube. MediaPipe Hand is a pre-trained model shipped with the MediaPipe framework that is dedicated to tracking hand palms and fingers. MediaPipe Hand consists of two pipelined CNN models dedicated to Palm Detection and Hand Landmark. It has many applications one of which is dedicated to the recognition of the shape and motion of hands. This operation is performed in two stages: the first stage, referred to as the palm detector, detects the hands in an image and outputs their bounding box. The second step, cascading the first one, provides the key point of the hand.

In our proposed approach, we have exploited the MediaPipe Hand for detecting when one or more hands are in the video for enabling the next step. Frames of the video are singularly evaluated and forwarded to the next step. Since MediaPipe is tiny, light, and efficient, it can be effectively implemented on embedded Internet-of-Things (IoT) devices such as mobile phones or smart cameras, while significantly decreasing the computational load on the second stage of the platform by filtering out worthless input.

5.3. Inner data processing

If the second and last steps are enabled by the hand detection model, the inner data processing step must manipulate the input video to be made compliant with the classifier input requirements. Each frame is scaled, preserving the original aspect ratio, pixel centered, and cropped to a size of 112 × 112. The video is downsampled to 10 frames per second. The time-consecutive frames are accumulated in a 16-element queue and they are provided to the SFH detection step.

5.4. Real-time gesture classification model

In the last stage of the platform, we adopted the MobileNet model. MobileNet is an architecture presented in 2017 by Google (Howard et al., 2017). It is a CNN widely adopted for mobile vision applications, especially classification tasks, suitable to be run on light systems. In particular, it exploits convolution factorization for reducing the number of weights. Since the goal of our platform is real-time "Signal for Help" detection, we adopted the 3D version of the MobileNet architecture

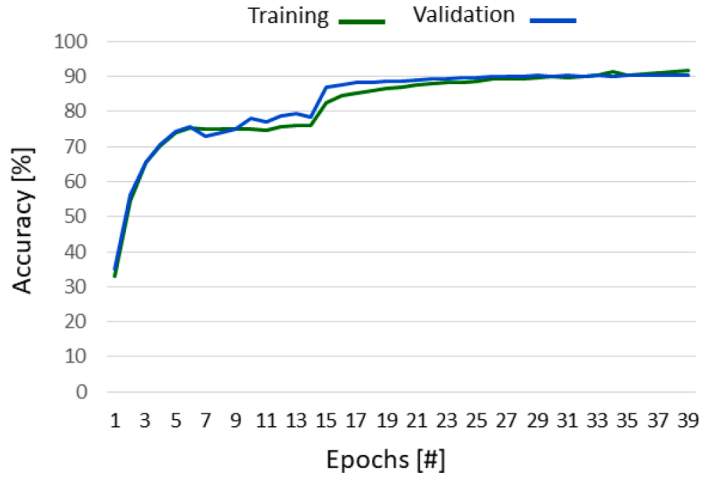


Fig. 5. Accuracy obtained after different epochs for MobileNet trained with Jester Dataset.

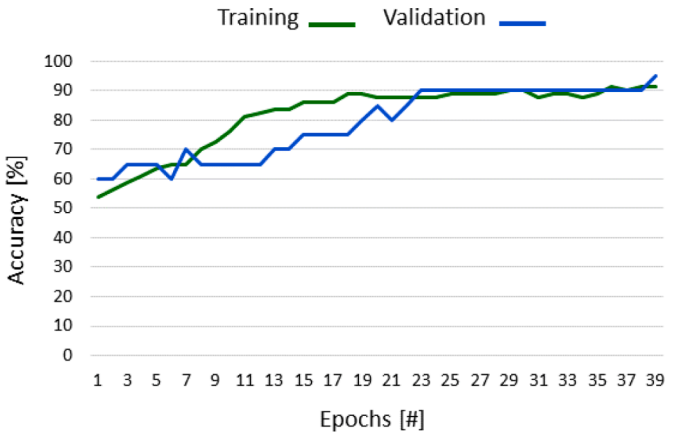


Fig. 6. Accuracy obtained after different epochs for MobileNet fine-tuned with the "Signal for Help" Dataset.

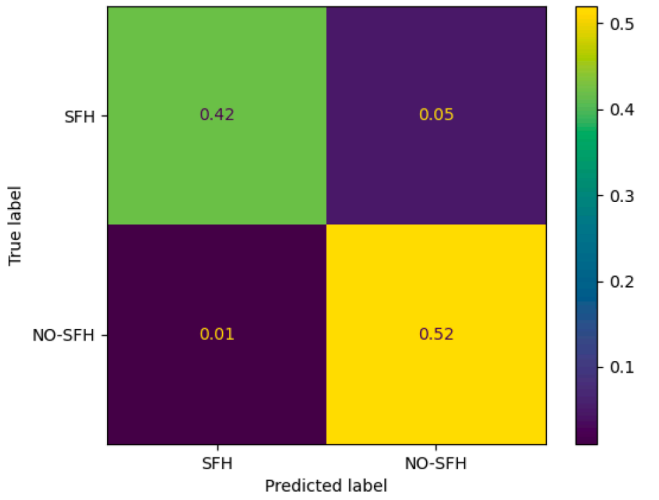


Fig. 7. Confusion Matrix for the developed Platform with the "Signal for Help" dataset.

presented in Kopuklu et al. (2019). Indeed, 3D CNNs have proven to better perform when dealing with video frames compared to traditional CNNs. This stage is in charge of evaluating if the "Signal for Help" appears in the video. The input of the stage is a collection of 16 time-consecutive frames. Since the output is binary, the last layer has been replaced by a two-neuron layer in the final version of the network.

5.5. Training methodology

MediaPipe Hand model is shipped already trained, achieving high accuracy of more than 95%, therefore, no additional training has been needed.

Differently, MobileNet requires to be trained with a suitable dataset to be capable of classifying the "Signal for Help" hand gesture. Most of the time, a CNN classifier is accurate only when a huge amount of data is available. To overcome this challenge, we relied on a transfer learning approach based on fine-tuning. In transfer learning, a large dataset with features similar to the actual dataset is exploited for creating a pre-trained model of the network. On training a model using a large dataset that is similar in order to transfer its knowledge to a smaller dataset. This concept has been exploited in our approach (Fig. 4) to compensate for the small created "Signal for Help" hand gesture. Therefore, the training phase of the MobileNet classifier is divided into two phases: firstly, a traditional training phase using an open-source hand gesture dataset, and secondly, a fine-tuning training phase based on the open-source "Signal for Help" dataset that we collected and introduced in the previous section.

The dataset selected to be used during the first phase is the Jester dataset (Materzynska et al., 2019). After the network has been trained for classifying hand gestures, we moved toward the next phase, fine-tuning the network for binary recognition of the "Signal for Help" hand gesture. To do so, the last fully connected layer of the classifier, which previously consisted of 27 neurons (since Jester has 27 unique gestures) has been replaced with a 2-neuron layer for supporting the binary classification of the "Signal for Help" gesture. After that, a new training phase has been performed using the "Signal for Help" dataset. During this phase, only the weights of the last layer are trained while the other parameters of the network are kept unchanged.

6. Experiments

This section presents the result of implementing the developed framework on the machine with Intel i9-9900K@3.6 GHz CPU as well as the Jetson Nano NVIDIA developer kit and reports a comparison between the developed framework with the state-of-the-art solutions in terms of resource usage and performance.

In order to demonstrate the efficiency of our proposed framework for real-time recognition of "Signal for Help", we carried out different experiments. We started with training the proposed platform on the "Signal for Help" dataset. Additionally, we performed a comparison with state-of-the-art models to confirm the efficiency of our proposed platform as a valuable solution for embedded lightweight systems. Please note that all the operations including the training and testing of the models are performed on a machine with Intel i9-9900K@3.6 GHz CPU, 31.2 GB memory, and NVIDIA GeForce RTX 2080 Ti GPU while verifying the efficiency of the developed framework for embedded applications with respect to state-of-the-art, the proposed models and state-of-the-art selected models are implemented on Jetson Nano Developer kit.

6.1. Platform training results

As it has been mentioned before, we exploited the already trained MediaPipe model for performing the hand detection task. Therefore, we started the training phase directly from the second stage, the MobileNet hand gesture classification model.

As noted, due to the small size of the created "Signal for Help" dataset, we have applied the transfer learning technique. Therefore, training of the network is performed in two phases. As the first phase, the Jester dataset (Materzynska et al., 2019) is exploited. It contains 27 different classes, intended for training machine learning models to recognize human hand gestures. It consists of 148,092 videos. A ratio of 8:1:1 has been used for splitting the dataset in training, validation, and testing sets. The dataset also includes two "no gesture" classes to help the network distinguish between specific gestures and unknown movements. The training videos of the Jesture dataset were used for training the MobileNet classifier.

MobileNet process 16 frames at once, encoding both spatial and temporal information. We trained the MobileNet model with Jester dataset for 40 epochs, on a machine with an NVIDIA GeForce RTX 2080 Ti GPU, which required 26 h. We have used an SGD optimizer with a learning rate that started at 0.1 and has been divided by 10 at the 15th, 25th, and 35th epochs, cross-entropy loss function, a batch size of 64, a dampening of 0.9, and weight decay at 0.001.

Please note that the average length of videos of Jester is around 3 s, 36 frames. Therefore, we decided to downscale the videos to 10 frames per second, in order to meet the expected rate for our architecture.

In order to improve the prediction of MobileNet and prevent overfitting, we applied data augmentation. To do so, each image is cropped to 112 × 112 and rescaled by a random parameter between 0.5 to 1.

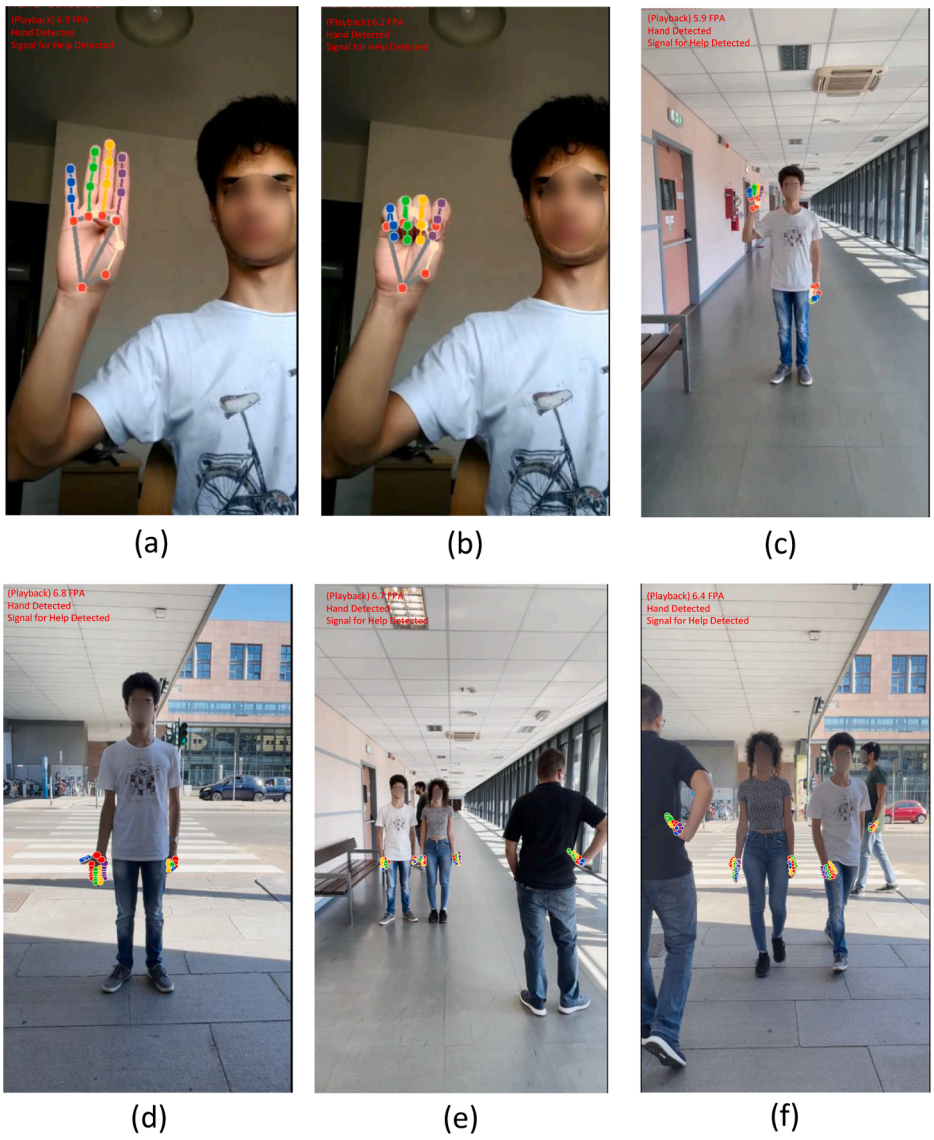


Fig. 8. Different scenarios of the testing videos executed on the developed platform (a) close distance from the camera with the plane background (b) far distance from the camera with the plane camera (c) far distance from the camera with the urban background (d) multiple hands far distance from the camera in plane background (e) multiple hands far distance from the camera in urban background.

Table 2
Accuracy Comparison of the developed platform with different versions of the "Signal for Help" dataset.

Dataset Version	Accuracy [%]	Accuracy Reduction [%]
Original	91.25	–
Low Resolution	86	5.75
Black & White	91	0.27
Low Resolution and Black &White	85	6.84
With Noise	76	16.71

During the validation and testing, the image is scaled down, preserving its aspect ratio, and cropped to 112×112 image size.

At the end of the first phase of the training, MobileNet achieved 91.71% accuracy trained on 27 classes of the Jester dataset.

The second phase consists of applying transfer learning techniques by freezing the pre-trained model of MobileNet and fine-tuning the last layer using the "Signal for Help" dataset. Following the Jesture division, we kept the same ratio, 8:1:1 for dividing the dataset into training, validation, and testing sets. The model converges after 39 epochs, with a learning rate of 0.0005 and a batch size of 40. After the second training phase, the model achieved 91.25% accuracy on the "Signal for Help" test set. The training parameters used for MobileNet during the training and fine-tuning phases are reported in Table 1, while Figs. 5 and 6 represents the trend of the accuracy during the epochs of the training and fine-tuning steps, respectively. As can be expected and observed from this figure, the accuracy increases with the increase of the epochs for both the training and validation sets.

6.2. Real-time detection and classification results

After the training phase, we move forward to verify the efficiency of the developed workflow. We evaluate the execution performance of our proposed workflow on each video of the testing set of the "Signal for Help" dataset. Our system runs on average at 62 fps when there is no gesture, therefore, only the MediaPipe hand detector is active, while in the presence of the gesture, therefore, both the MediaPipe hand detector and MobileNet classifier are active, it runs at 16 fps. Testing the videos in the testing set of the dataset, in 94% of the cases, a true alarm has been detected while the 5% of false alarms and 1% of missing alarms were

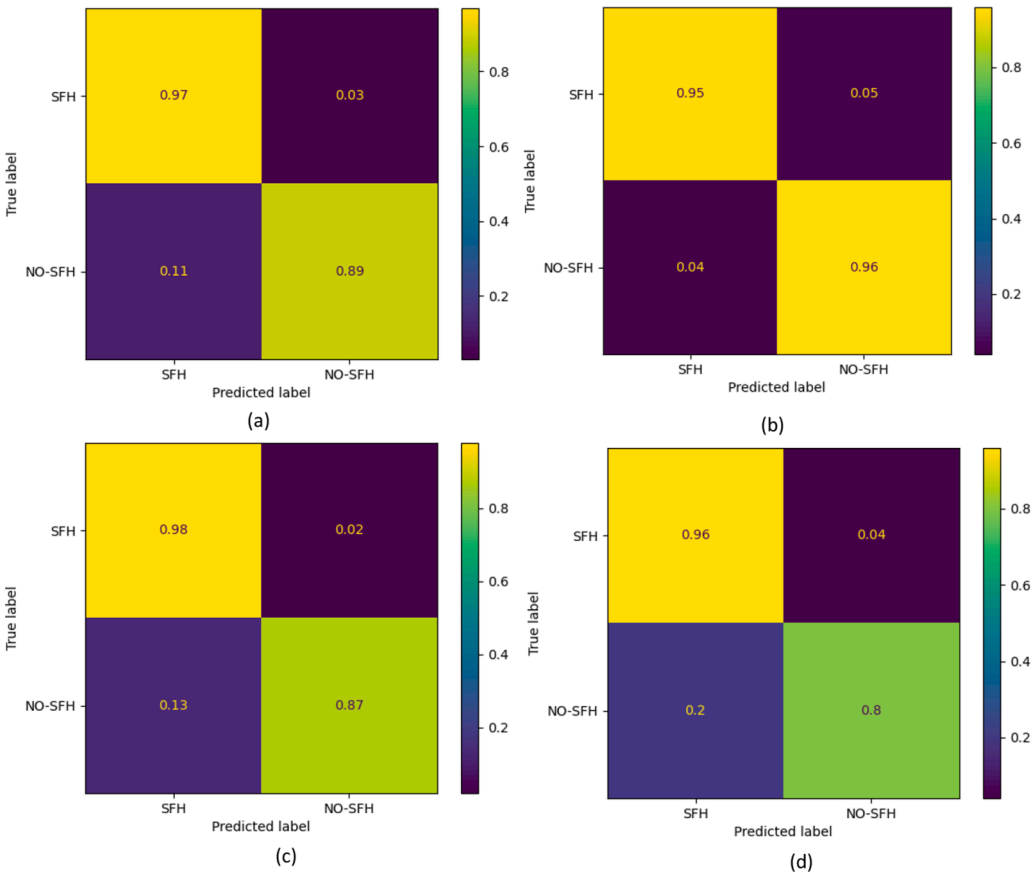


Fig. 9. Confusion Matrix for the developed Platform with different modified versions "Signal for Help" dataset (a) Low Resolution (b) Black & White (c) Low Resolution and Black & White (d) with Noise.

Table 3
Comparison between MediaPipe and ResNet-50 Hand Detection Models.

CNN Models	MediaPipe (Proposed Solution)	ResNet-50 (State-of-the-art Solution)
Parameters [#]	2M	23M
Performance	~30 fps	~12 fps
Accuracy [%]	~96%	~97%

Table 4
Comparison between MediaPipe and ResNet-50 Hand Detection Models.

Dataset Type	Training Time [h]		Validation Accuracy [%]	
	MobileNet (Proposed)	ResNeXt-101 (state-of-the-art)	MobileNet (Proposed)	ResNeXt-101 (state-of-the-art)
Jesture	26	33.4	91.71	93.8
Signal for Help	2	3.7	91.25	94.9

Table 5
Comparison between MobileNet and ResNext-101 classifier in terms of complexity and performance. Performance evaluated on Jetson Nano NVIDIA developer kit.

CNN models	MobileNet (Proposed Solution)	ResNeXt-101 (State-of-the-art Solution)
Parameters [#]	14.1M	48.75M
Layers [#]	28	101
Performance	~6 fps	~4 fps

observed as it is represented in Fig. 7 which reports the confusion matrix of the developed platform exploiting "Signal for Help" test set. Fig. 8 represents some cases in which the "Signal for Help" have been detected through video, in real-time. As can be seen, the shown cases represent different scenarios, outdoor, indoor with natural and artificial light, in the presence of single or multiple hands.

Moreover, to verify the efficiency of our method considering more realistic scenarios, we mimic the CCTV videos by modifying the dataset, creating four new versions of the dataset: "Low Resolution", "Black & White" and "Low Resolution and Black & White" and "with Noise". We test the developed platforms with the four newly modified datasets. Table 2 reports the accuracy of the developed platform in testing with each of these four datasets while Fig. 9 represents the Confusion Matrix of the platform for each test. As expected, the results show a reduction in accuracy with respect to the original dataset, ranging from 0.27 to 16.71% reduction.

6.3. Comparison with state-of-the-art

In order to confirm the efficiency of our proposed solutions for the embedded system with respect to the state-of-the-art models, we have implemented our proposed solution as well as two chosen state-of-the-art models on the Jetson Nano developer kit while we investigated the performance of the proposed framework by comparing them with the state-of-the-art models. We have chosen ResNet-50 as the hand detection algorithm while ResNeXt-101 was chosen as the hand classification algorithm as the two most common state-of-the-art solutions.

The same as the MediaPipe model, we chose the pre-trained ResNet-50 hand detection model. We used the Jetson Nano NVIDIA developer kit with a 128-core NVIDIA Maxwell GPU, Quad-Core ARM core @1.43 GHz, and 2 GB of memory for training and testing of the chosen models.

Table 3 shows the comparison between the accuracy of ResNet-50 with respect to the MediaPipe model used in our proposed platform on the Jesture dataset.

As can be observed in the table, MediaPipe required fewer resources which is more suitable for embedded applications while the acquired accuracy from MediaPipe and ResNet-50 is almost the same.

Moving toward the classification model, we trained the ResNeXt-101 classification model following the same strategy as the training of the MobileNet classifier. Therefore, as the first step, we trained the ResNeXt-101 with the Jesture dataset. Secondly, we applied transfer learning techniques, freezing the pre-trained model of ResNeXt-101 and fine-tuning the last layer using the "Signal for Help" dataset.

As reported in Table 4, we trained ResNext with Jester dataset for 40 epochs, on a machine with an NVIDIA GeForce RTX 2080 Ti graphic card, which required 33.4 h. We have used the SGD optimizer and cross-entropy loss with a learning rate that started at 0.1 and has been divided by 10 at the 15th, 25th, and 35th epochs with a batch size of 46, dampening of 0.9, and weight decay at 0.001. During the first phase, ResNext achieved 93.8% accuracy.

We continued the training of the ResNeXt-101 through the second phase on the "Signal for Help" dataset. The model converges after 10 epochs, with a learning rate of 0.01, batch size of 128, and down-sampling of 5. After the second training phase, the model achieved 94.9% accuracy on the "Signal for Help" test set.

As can be observed in Table 5, ResNext has slightly higher accuracy with respect to MobileNet. However, please notice that considering the final goal which is developing a "Signal for Help" recognition platform which is light-weighted and required limited resources, it is important to take into account the complexity and resource constraints of each model. Therefore, we moved forward by performing a comparison between the characteristics of the two models. As it is reported in Table 5, ResNext-101 is known as a complex and deep CNN which requires many resources. In fact, the number of parameters defining the ResNeXt model is 4 times more than MobileNet while they both operate at the same performance rate which confirms the choice of MobileNet as a light-weight accurate classification model of our proposed "Signal for Help" recognition workflow.

We would like to highlight that the proposed solution with respect to state-of-the-art provides slightly lower accuracy on the detection and classification of "Signal for Help" while the resource usage is drastically, around 4 times, lower, and inference has a speedup of about 50%, which makes the developed platform a perfect candidate for embedded systems with low available resources and low power.

7. Conclusion

This paper presents the development of the very first real-time platform for the detection of "Signal for Help" hand gestures applicable to an embedded system with limited resources available. The proposed platform is composed of two 3D CNN-based models, MediaPipe and MobileNet for the detection of the hand in the video and classification of the hand gesture which has been merged to create one single hierarchical gesture recognition architecture. Moreover, thanks to the contribution of students of Politecnico di Torino, we have created the first dynamic dataset of "Signal for Help", including 200 videos, available publicly. The developed platform has been implemented on two machines, a machine with Intel i9-9900K@3.6 GHz CPU and Jetson Nano Developer Kit as an embedded system with a performance rate of 16 fps and 6 fps, respectively, showing 91.71% accuracy. Even though the accuracy of the developed platform is in the same range as the state-of-the-art solutions for hand gesture recognition platforms, the resource usage is 4 times less and inference has 50% speedup, which provides a great option for embedded systems.

8. Future works

In the future, we plan to, first of all, enrich the created dataset to be able to train a more accurate and efficient CNN-based platform that is complemented with different real scenarios. Secondly, we are negotiating with some public and private organizations in Italy in order to put the developed platform into use, exploiting the surveillance cameras available in the urban areas of big cities in Italy. In this regard, we are collaborating with a team of urban designers who are performing a deep investigation into the features of the security cameras of smart cities in order to identify a suitable location for integrating the developed platform with the surveillance cameras.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

We wish to confirm that there are no known conflicts of interest associated with this publication and therehas been no significant financial support for this work that could have influenced its outcome.

Data availability

We have shared the data on GitHub and insert the links in the article.

Acknowledgment

This project has been selected as the top three proposals during the competition "Women to Women Tech Ideas" organized by IEEE Women in Engineering in March 2022. Moreover, we would like to thank all the students of Politecnico di Torino who contributed to the creation of the "Signal for Help" dataset. All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

Agrawal, R., & Gupta, N. (2016). Real time hand gesture recognition for human computer interaction. In *Proceedings of the IEEE 6th international conference on advanced computing (IACC)* (pp. 470–475). https://doi.org/10.1109/IACC.2016.93

Ashwini, R Amutha, Rajavel, R., & Anusha, D. (2020). Classification of daily human activities using wearable inertial sensor. In *Proceedings of the international conference on wireless communications signal processing and networking (WiSPNET)* (pp. 1–6).

Athilakshmi, Rajangam, Rajavel, Ramadoss, & Jacob, Shomona Gracia (2018). A survey on deep-learning architectures. *Journal of Computational and Theoretical Nanoscience, 15*(8), 2577–2579.

Chen, L.-H., Hsu, H.-W., Wang, L.-Y., & Su, C.-W. (2011). Violence detection in movies. In. In *Proceedings of the international conference on computer graphics, imaging and visualization (CGIV)*.

Elliott, G., Meehan, K., & Hyndman, J. (2021). Using CNN and tensorflow to recognise 'Signal for Help' hand gestures. In *Proceedings of the IEEE 12th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)* (pp. 0515–0521). https://doi.org/10.1109/UEMCON53757.2021.9666484

Fu, E. Y., Va Leong, H., Ngai, G., & Chan, S. (2016). Automatic fight detection in surveillance videos. In *Proceedings of the 14th international conference on advances in mobile computing and multi media, Ser. MoMM '16* (pp. 225–234). Association for Computing Machinery. https://doi.org/10.1145/3007120.3007129.

A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand et al. "MobileNets: Efficient convolutional neural networks for mobile vision applications", in Computer vision and pattern recognition, 2017, arXiv:1704.04861.

Köpüklü, O., Gunduz, A., Kose, N., & Rigoll, G. (2019). Real-time hand gesture detection and classification using convolutional neural networks. In *Proceedings of the 14th IEEE international conference on automatic face & gesture recognition (FG 2019)* (pp. 1–8). https://doi.org/10.1109/FG.2019.8756576

Kiruthiga, P., & Rajavel, R. (2021). Audio visual emotion recognition in children. In *Proceedings of the international conference on power of digital technologies in societal empowerment, CHENCON2021*.

Kopuklu, O., Kose, N., Cunduz, A., & Rigoll, G. (2019). Resource efficient 3D convolutional neural networks. in. In *Proceedings of the international conference on computer vision*.

Materzynska, J., Berger, G., Bax, I., & Memisevic, R. (2019). The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF international conference on computer vision workshop (ICCVW)* (pp. 2874–2882). https://doi.org/10.1109/ICCVW.2019.00349

Molchanov, P., Gupta, S., Kim, K., & Pulli, K. (2015). Multi-sensor system for driver's hand-gesture recognition. In. In *, 1. Proceedings of the 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)* (pp. 1–8). IEEE.

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4207–4215).

Ohn-Bar, E., & Trivedi, M. M. (2014). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Transactions on Intelligent Transportation Systems, 15*(6), 2368–2377.

Rahman, M. M., Islam, M. S., Rahman, M. H., Sassi, R., Rivolta, M. W., & Aktaruzzaman, M. (2019). A new benchmark on american sign language recognition using convolutional neural network. In *Proceedings of the international conference on sustainable technologies for industry 4.0 (STI)* (pp. 1–6). https://doi.org/10.1109/STI47673.2019.9067974

Ramzan, M., Abid, A., Khan, H. U., Awan, S. M., Ismail, A., Ahmed, M., et al. (2019). A review on state-of-the-art violence detection techniques. *IEEE Access : Practical Innovations, Open Solutions, 7*, 107560–107575. https://doi.org/10.1109/ACCESS.2019.2932114

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In. In *Proceedings of the advances in neural information processing systems* (pp. 568–576).

Smith, K. A., Csech, C., Murdoch, D., & Shaker, G. (2018). Gesture recognition using mm-wave sensor for human-car interface. in. In *, 2. Proceedings of the IEEE Sensors Letters* (pp. 1–4). https://doi.org/10.1109/LSENS.2018.2810093.

Sudhakaran, S., & Lanz, O. (2017). Learning to detect violent videos using convolutional long short-term memory. In *Proceedings of the 14th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1–6). https://doi.org/10.1109/AVSS.2017.8078468

Tan, Y. S., Lim, K. M., & Lee, C. P. (2021a). Hand gesture recognition via enhanced densely connected convolutional neural network. *Elsevier Expert Systems with Applications., 175*, Article 114797. https://doi.org/10.1016/j.eswa.2021.114797

Tan, Y. S., Lim, K. M., & Lee, C. P. (2021b). Hand gesture recognition via enhanced densely connected convolutional neural network. *Elsevier Expert Systems with Applications, 175*, 1–12. https://doi.org/10.1016/j.eswa.2021.114797

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 4489–4497). IEEE.

Wahid, M., Tafreshi, R., Sowaidi, M. A., & Langari, R. (2018). Subject-independent hand gesture recognition using normalization and machine learning algorithms. in *Elsevier Journal of Computational Science*, 69–76. https://doi.org/10.1016/j.jocs.2018.04.019.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., et al. (2016). Temporal segment networks: Towards good practices for deep action recognition. In. In *Proceedings of the European conference on computer vision* (pp. 20–36). Springer.

Žemgulys, J., Raudonis, V., Maskeliūnas, R., & Damaševičius, R. (2018). *Recognition of basketball referee signals from videos using histogram of oriented gradients (HOG) and support vector machine (SVM)*. in. In . *130* pp. 953–960). https://doi.org/10.1016/j.procs.2018.04.095.

Zhang, Y., Cao, C., Cheng, J., & Lu, H. (2018). EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia, 20*(5), 1038–1050.