applied sciences

MDPI

Article

Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model

Abdullah Mujahid [1], Mazhar Javed Awan [2], Awais Yasin [3], Mazin Abed Mohammed [4], Robertas Damaševičius [5,*], Rytis Maskeliūnas [6] and Karrar Hameed Abdulkareem [7]

[1] Department of Computer Science, University of Management and Technology, Lahore 54770, Pakistan; abdullahmujahidali1@gmail.com
[2] Department of Software Engineering, University of Management and Technology, Lahore 54770, Pakistan; mazhar.awan@umt.edu.pk
[3] Department of Computer Engineering, National University of Technology, Islamabad 44000, Pakistan; awaisyasin@nutech.edu.pk
[4] Information Systems Department, College of Computer Science and Information Technology, University of Anbar, Anbar 31001, Iraq; mazinalshujeary@uoanbar.edu.iq
[5] Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland
[6] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania; rytis.maskeliunas@vdu.lt
[7] College of Agriculture, Al-Muthanna University, Samawah 66001, Iraq; Khak9784@mu.edu.iq
* Correspondence: robertas.damasevicius@polsl.pl

check for updates

Citation: Mujahid, A.; Awan, M.J.; Yasin, A.; Mohammed, M.A.; Damaševičius, R.; Maskeliūnas, R.; Abdulkareem, K.H. Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. *Appl. Sci.* **2021**, *11*, 4164. https://doi.org/10.3390/app11094164

Academic Editor: Athanasios Nikolaidis

Received: 3 April 2021
Accepted: 29 April 2021
Published: 2 May 2021

Abstract: Using gestures can help people with certain disabilities in communicating with other people. This paper proposes a lightweight model based on YOLO (You Only Look Once) v3 and DarkNet-53 convolutional neural networks for gesture recognition without additional preprocessing, image filtering, and enhancement of images. The proposed model achieved high accuracy even in a complex environment, and it successfully detected gestures even in low-resolution picture mode. The proposed model was evaluated on a labeled dataset of hand gestures in both Pascal VOC and YOLO format. We achieved better results by extracting features from the hand and recognized hand gestures of our proposed YOLOv3 based model with accuracy, precision, recall, and an F-1 score of 97.68, 94.88, 98.66, and 96.70%, respectively. Further, we compared our model with Single Shot Detector (SSD) and Visual Geometry Group (VGG16), which achieved an accuracy between 82 and 85%. The trained model can be used for real-time detection, both for static hand images and dynamic gestures recorded on a video.

Keywords: convolutional neural network; hand gesture; digital image processing; YOLOv3; artificial intelligence

1. Introduction

The interaction between humans and computers has increased widely, while the domain is witnessing continuous development, with new methods derived and techniques discovered. Hand gesture recognition is one of the most advanced domains in which computer vision and artificial intelligence has helped to improve communication with deaf people but also to support gesture-based signaling systems [1,2]. Subdomains of hand gesture recognition include sign language recognition [3–5], recognition of special signal language used in sports [6], human action recognition [7], pose and posture detection [8,9], physical exercise monitoring [10], and controlling smart home/assisted living applications with hand gesture recognition [11].

Over the years, computer scientists have used different computation algorithms and methods to help solve our problems while easing our lives [12]. The use of hand gestures in different software applications has contributed towards improving computer and human interaction [13]. The progress of the gesture recognition systems plays a vital role in the development of computer and human interaction, and the use of hand gestures in various domains is growing more frequent. The application of the use of hand gestures can now be seen in games [14], virtual reality [15,16] and augmented reality [17], assisted living [18,19], cognitive development assessment [20], etc. The recent development of hand gesture recognition in different sectors has grabbed the attention of industry, too, for human-robot interaction in manufacturing [21,22], and control of autonomous cars [23].

The main aim of this real-time hand gesture recognition application is to classify and recognize the gestures. Hand recognition is a technique in which we use different algorithms and concepts of various techniques, such as image processing and neural networks, to understand the movement of a hand [24]. In general, there are countless applications of hand gesture recognition. For example, for deaf people who cannot hear, we can communicate with their familiar sign language.

There are many object detection algorithms that help to detect and determine what the gesture is that each algorithm targets. This paper explores a few algorithms and detects which algorithm is better than the others, providing better accuracy with fast and responsive results. To achieve this detection, You Only Look Once (YOLO) v3 and Single Shot Detector (SSD) algorithms were used to evaluate the structure and mechanism deduction of hand gesture recognition.

YOLO is a convolutional neural network algorithm, which is highly efficient and works tremendously well for real-time object detection [25,26]. A neural network not only helps in feature extraction, but it can also help us understand the meaning of gesture, and help to detect an object of interest. A similar approach was adopted in [27], but the main difference is that in [27] it was done through Light YOLO, which is completely different from YOLOv3. Light YOLO is preferred for applications built on RaspberryPi. It achieves a good frame per second (fps) rate as compared to YOLOv3. However, the accuracy of YOLOv3 is 30% better when compared to Light YOLO, and as the application is not focused on the Internet-of-Things (IoT) product, YOLOv3 is preferred.

The main contributions of this study are summarized as follows:

- A lightweight proposed model where there is no need to apply as much preprocessing which involves filtering, enhancement of images, etc.;
- A labeled dataset in both Pascal VOC and YOLO format;
- This is the first gesture recognition model that is dedicated to the mentioned gestures using YOLOv3. We use YOLOv3 as it is faster, stronger, and more reliable compared to other deep learning models. By using YOLOv3, our hand gesture recognition system has achieved a high accuracy even in a complex environment, and it successfully detected gestures even in low-resolution picture mode;
- The trained model can be used for real-time detection, it can be used for static hand images, and it also can detect gestures from video feed.

The organization of our study is as follows: Section 2 presents the related work and reviews the latest studies on hand gesture recognition. The materials and methods that were used in this study are in Section 3. Section 4 presents the results of the Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. The discussion of the results of the proposed Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model are discussed in Section 5. Finally, Section 6 concludes the study directions and future works.

2. Related Work

There are various studies on hand gesture recognition as the area is widely expanding, and there are multiple implementations involving both machine learning and deep learning methods aiming to recognize a gesture that is intonated by a human hand. Further, some papers are reviewed to understand the mechanism of the hand gesture recognition technique.

The study [28] demonstrated that with insignificant computational cost, one of the normal and well-known designs, CNN, accomplished higher paces of perceiving components effectively. The proposed strategy focused only on instances of gestures present in static images, without hand location, and followed the instances of hand impediment with 24 gestures, utilizing a backpropagation algorithm and segmentation algorithm for preparing the multi-layer propagation, and for the backpropagation calculation having its influence, sorting out the blunder proliferated on the contrary request [29]. Moreover, it utilized a convolutional neural organization, and sifting incorporated a distinctive division of image division and recognition.

Among these techniques and methods there is a popular method which is used by several other detection applications, and that is Hidden Markov Models (HMM). There are different detection variants that are often used by an application, which we have come across, and the application this paper refers to checks and works with all of it by learning and looking towards other papers. This application deals with all three mediums: image, video, and webcam [30]. Going through a general paper written by Francois, in which he refers to an application that detects posture, the detection is through video and uses the HMM. The process that the paper of Francois and other researchers worked on is related to this application, which this paper refers to first by extracting the information from either image, video, or real-time detection using webcams. These features are detected by the three methods mentioned, but the main concern of all the methods, whether they are CNN-related, RNN-related, or using any other technique, is that all of them use fitting techniques, and these techniques refer to the bounding box that this paper discussed. The box represents the information that is detected, from which they gain a confidence value, and the value that is the highest is the output of what image is being displayed. Besides this, all other information varies depending upon the method that others are using; despite that, some other devices and techniques that are related to segmentation, general localization, and even fusion of other different partials help achieve the tasks of detecting and recognizing.

Nyirarugira et al. [31] proposed Social Touch Gesture Recognition using CNN. He utilized various calculations such as Random Forest (RF), boosting algorithms, and the Decision Tree algorithm to recognize gestures, utilizing 600 arrangements of palmar images with 30 examples utilizing convolutional neural organization, and finding an ideal method on the premise of the framework of an 8×8 network. The casing length is variable anyway so the outcome decides the ideal casing length. It is performed using a dataset as of late assessed with different subjects that perform changing social gestures [32]. A system that gathers contact gestures in a practically constant manner using a deep neural organization is favorable to present. The results showed that their strategy performed better when differentiated, and the previous work was reliant on leave-one-subject-out cross-endorsement for the CoST dataset. The proposed approach presents two points of interest differentiated from those in the current writing, acquiring an accuracy of 66.2% when perceiving social gestures.

In the study of Multiscale CNNs for hand detection, excited by the headway of article recognition in the field of PC vision, numerous techniques have been proposed for hand-distinguishing proof in the latest decade [33]. The most untroublesome procedure relies upon the recognition of skin tone, which works on hands, faces, and arms, yet moreover has issues because of the affectability for brilliant changes. The contributing components of this multifaceted nature fuse substantial hindrance, low objectivity, fluctuating illumination conditions, various hand gestures, and the incredible coordinated efforts among hands and dissents or various hands. They further presented a Multiscale Fast R-CNN approach to manage to correctly recognize human hands in unconstrained pictures. By merging staggered convolutional features, the CNN model can achieve favored results over the standard VGG16 model, accomplishing practically 85% of 5500 images for the testing and 5500 for the preparing set.

Saqib et al. [34] used a CNN model augmented by edit distance for the recognition of static and dynamic gestures of Pakistani sign language, and achieved 90.79% accuracy. Al-Hammadi et al. [35] proposed a 3DCNN model to learn region-based spatiotemporal features for hand gestures. The fusion techniques to globalize the local features learned by the 3DCNN model were used to improve the performance. The approach obtained recognition rates of 98.12, 100, and 76.67% on three color video gesture datasets.

Do et al. [36] proposed a multi-level feature LSTM with Conv1D, the Conv2D pyramid, and the LSTM block. The proposed method exploited skeletal point-cloud features from skeletal data, as well as depth shape features from the hand component segmentation model. The method achieved accuracies of 96.07 and 94.40% on the Dynamic Hand Gesture Recognition (DHG) dataset with 14 and 28 classes, respectively. The study extracted diversity of dynamic hand gestures from 14 depth and 28 skeletal data through the LSTM model with two pyramid convolutional blocks. The accuracy of 18 classes was 94.40%. Elboushaki et al. [37] learned high-level gesture representations by using Convolutional Residual Networks (ResNets) for learning the spatiotemporal features from color images, and Convolutional Long Short-Term Memory Networks (ConvLSTM) to capture the temporal dependencies between them. A two-stream architecture based on 2D-ResNets was then adopted to extract deep features from gesture representations.

Peng et al. [38] combined a feature fusion network with a ConvLSTM network to extract spatiotemporal feature information from local, global, and deep aspects. Local feature information was acquired from videos by 3D residual network, while the ConvLSTM network learned the global spatiotemporal information of a dynamic gesture. The proposed approach obtained 95.59% accuracy on the Jester dataset, and 99.65% accuracy on the SKIG (Sheffield Kinect Gesture) dataset. Tan et al. [39] proposed an enhanced, densely connected convolutional neural network (EDenseNet) for hand gesture recognition. The method achieved 99.64% average accuracy on three hand gesture datasets. Tran et al. [40] suggested a 3D convolution neural network (3DCNN) that could extract fingertip locations and recognize hand gestures in real-time. The 3DCNN model achieved 92.6% accuracy on a dataset of videos with seven hand gestures.

Rahim et al. [41] analyzed the translation of the gesture of a sign word into text. The authors of this paper performed the skinMask segmentation to extract features along the CNN. Having a dataset of 11 gestures from a single hand and 9 from double hands, the support vector machine (SVM) was applied to classify the gestures of the signs with an accuracy of 97.28%. Mambou et al. [42] analyzed hand gestures associated with sexual assault from indoor and outdoor scenes at night. The gesture recognition system was implemented with the combination of the YOLO CNN architecture, which extracted hand gestures, and a classification stage of bounding box images, which lastly generated the assault alert. Overall, the network model was not lightweight and had a lower accuracy.

Ashiquzzaman et al. [43] proposed a compact spatial pyramid pooling (SPP) a CNN model for decoding gestures or finger-spelling from videos. The model used 65% fewer parameters than traditional classifiers and worked 3$\times$ faster than classical models. Benitez-Garcia et al. [44] employed a lightweight semantic segmentation FASSD-Net network, which was improved over Temporal Segment Networks (TSN) and Temporal Shift Modules (TSM). They demonstrated the efficiency of the proposal on a dataset of thirteen gestures focused on interaction with touchless screens in real-time.

Summarizing, the biggest challenge faced by the researchers is designing a robust hand gesture recognition framework that overcomes the most typical problems with fewer limitations and gives an accurate and reliable result. Real-time processing of hand gestures also has some limitations, such as illumination variation, background problems, distance range, and multi-gesture problems. There are approaches to hand gesture recognition that use non-machine-learning algorithms, but there is a problem in that the accuracy varies, and in different environments such as light, it overlaps one gesture with another, which makes the approach less flexible and unable to adapt independently, when compared to the machine-learning approach. Therefore, the machine-learning approach was used for developing the system.

3. Material and Methods

This section is dedicated to the materials and the methods that were used in this study to achieve the gesture recognition that this paper aimed for. Section 3.1 explains the dataset and all the information related to the material. Section 3.2 deals with the algorithm and the methods that we used to solve the problem.

3.1. Dataset

Our dataset consisted of 216 images. These images were further classified into 5 different sets. Each set held an average of 42 images which were labeled using the YOLO labeling format. The dataset was labeled using a labeling tool, which was an open-source tool used to label custom datasets. We used the YOLO format, which labels data into text file format and holds information such as the class ID and class to which it belongs. Our classes started from 0 to 4, where 0 class ID is labeled as 1 and 4 class ID is labeled as 5. There were a total of 5 sets that our application detected, which were finger-pointing positions of 1, 2, 3, 4, and 5. Figure 1 displays the hand gestures in our collected dataset.

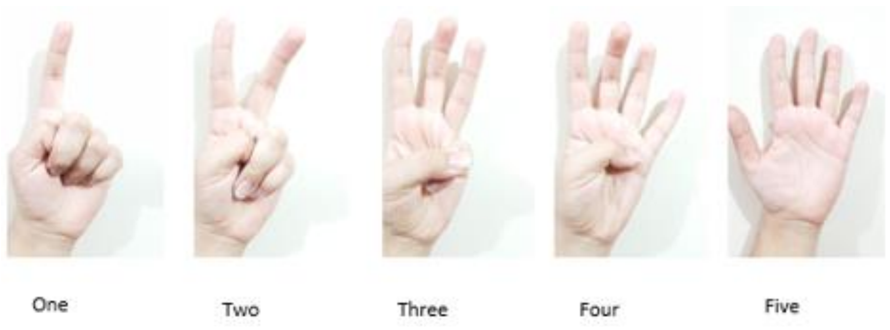

Figure 1. Hand gesture set to be detected.

3.2. Data Preprocessing

Data preprocessing is an important part of the before training and testing phase. Using a YOLO configuration with a total of 200+ images of the dataset, where 30 were used for the affine transformation by increasing 2-fold, each image was duplicated for reading and training, both left and right hand, by flipping it horizontally and sometimes taking the respective image of those hands for making the set more accurate. Furthermore, an additional 15 images were taken and labeled for the testing set. The data preprocessing step is important before moving towards post-processing, because we have to look at what type of data we have collected and which part of it will be useful for the purpose of training, testing, and for obtaining better accuracy. Table 1 shows the instances of five classes with the features of the YOLO-labeled data.

Table 1. Summary of data obtained from YOLO-labeled data file.

Class ID	X-Cord	Y-Cord	Width	Height
0	0.531771	0.490234	0.571875	0.794531
1	0.498437	0.533203	0.571875	0.905469
2	0.523438	0.579297	0.613542	0.819531
3	0.526563	0.564453	0.473958	0.819531
4	0.498611	0.587891	0.977778	0.792969

The above example is the representation of how these files will look when we label our dataset for training it on the desired model. Each line contains 5 different attributes, and all these attributes have their importance. Looking at the left first, we have class ID, followed by the next two, which are the labeled box co-ordinates of a gesture with the x-axis and y-axis values, followed by the width and height of that annotated image.

Figure 2 represents the correlation heat map of the YOLO-labeled dataset. Here, we can see the diversity of values among different labels, explaining the concentration and depth of our images represented in the form of a multi-dimensional matrix.

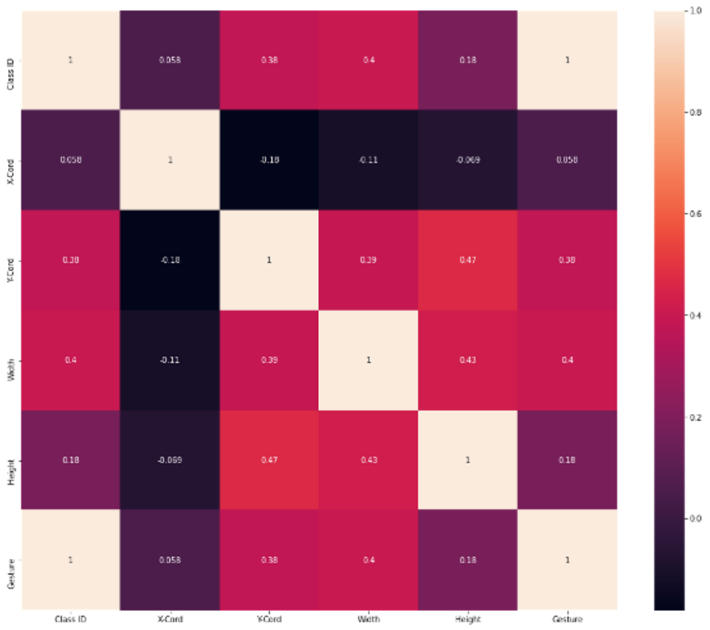

Figure 2. Heat map of the YOLO-labeled dataset.

3.3. *Proposed Method*

To understand the method which we are proposing, we need to look at the diagram presented in Figure 3 to understand better, generally, what and how our application is detecting objects. It is a kind of a general overview of the application that we have developed. The training process first requires collecting the dataset, and after that the next step is to label it, so we use YOLO annotation to label our data, which gives us some values that are later explained in the model process. After that, when the data is labeled, we then feed it to the DarkNet-53 model, which is trained according to our defined configuration. The image is captured through the camera that can be an integrated (primary) camera or it can be any external (secondary) camera. Other than that, the application can also detect gestures from a video input as well.

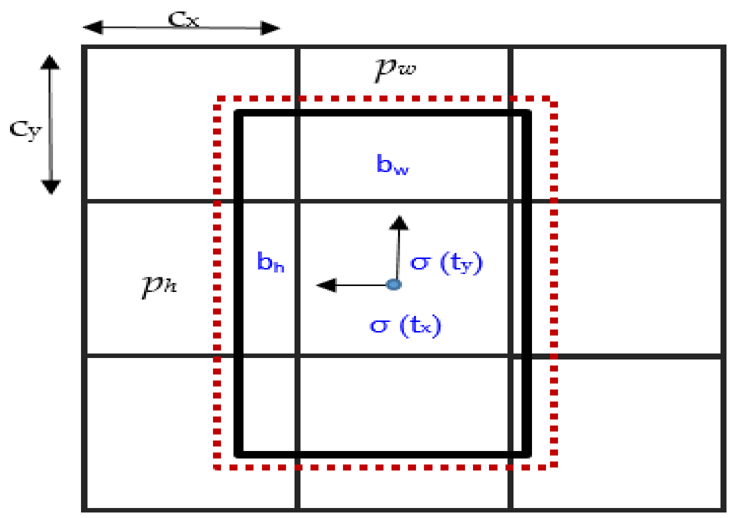

Figure 3. The bounding box highlighting positions of the values for further classification.

After capturing real-time objects with the help of the OpenCV [45] module, which is an open-source computer vision library, we can capture images and after that, we send them frame by frame from the real-time objects. Because of incorrect filtering, our dataset of collected images currently has a variable number of images. These images are labeled according to the classes we have stored for our YOLO algorithm, so we have successfully attained the coordinate and the class for our image set. After that, we can now set towards the training section. We then pass this set to our training algorithm, which is a deep neural network model YOLO.

Further, we discuss the methodology how YOLO deals with the network desired output which is achieved by using a formula which takes different co-ordinates, such as pw, ph, tx, ty, tw, th, cx, and cy. These are the variables which we use for the bounding box dimensions. Obtaining the values of the boundary box (x-axis, y-axis, height, and width) is described by Equation (1).

$$\begin{gathered} b_x = \alpha\left(\mathrm{t}_x\right) + \mathrm{Cx} \\ b_y = \alpha\left(\mathrm{t}_y\right) + \mathrm{Cy} \\ b_w = p_w e^t w \\ b_h = phe^t h \end{gathered} \tag{1}$$

where bx, by, bw, and bh are the box prediction components, x and y refer to the center co-ordinates, and w and h refer to the height and width of the bounding box. Equation (1) used in the YOLOv3 algorithm shows how it extracts the values from the image in the bounding box, and below is the diagram of how these values are extracted from the bounding box.

From Figure 3, we can understand how each value of the bounding box from the algorithm provides us with the co-ordinates of the center, x and y. From the prediction here the next important thing comes, which is the sigmoid function, which we have already discussed above, which filters out data except for the main part which is going to be recognized.

From Figure 4, we can understand the backend of the algorithm. When we pass an input, it first comes into the input layer, and after that it is further rendered and passes into the hidden layers, that are several in number and size, and are interconnected convolutions. These convolutional layers determine a special value, which is the value of confidence. In our case, if the value of the confidence threshold is greater than 0.5, then we assume that the application has successfully determined what object it has encountered.

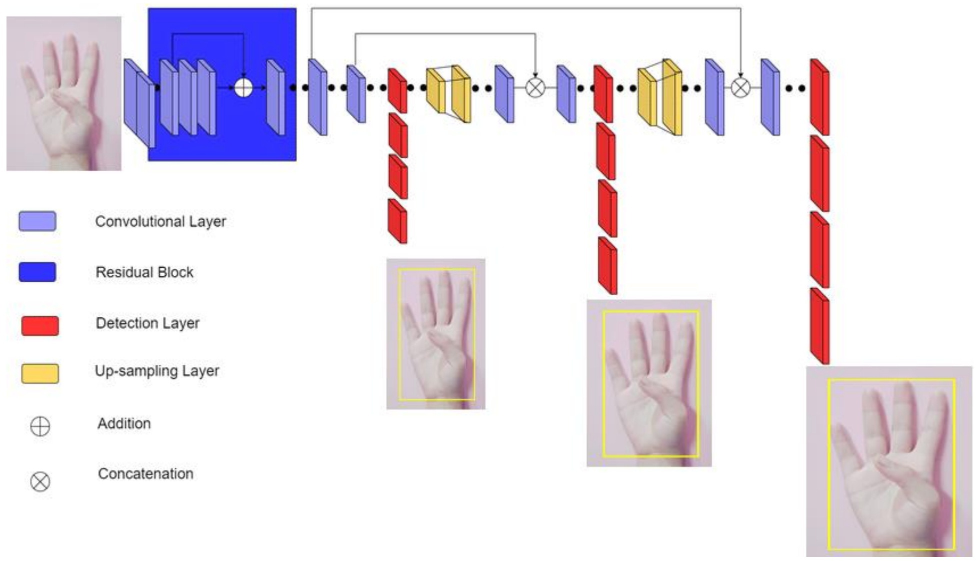

Figure 4. Architecture of the neural network for hand gesture recognition.

3.4. Implementation

Figure 5 explains how the YOLO algorithm plays its part when it acquires the image with the help of the OpenCV module. The image is then passed to the YOLO network, which then further identifies the required target. After doing that it sends it forward to the feature map prediction block, where it further extracts the information which is required to identify the gesture, then, after predicting it, sends it to the decoding part, where the output predicted is mapped onto the image and then displayed, as shown in Figure 5.

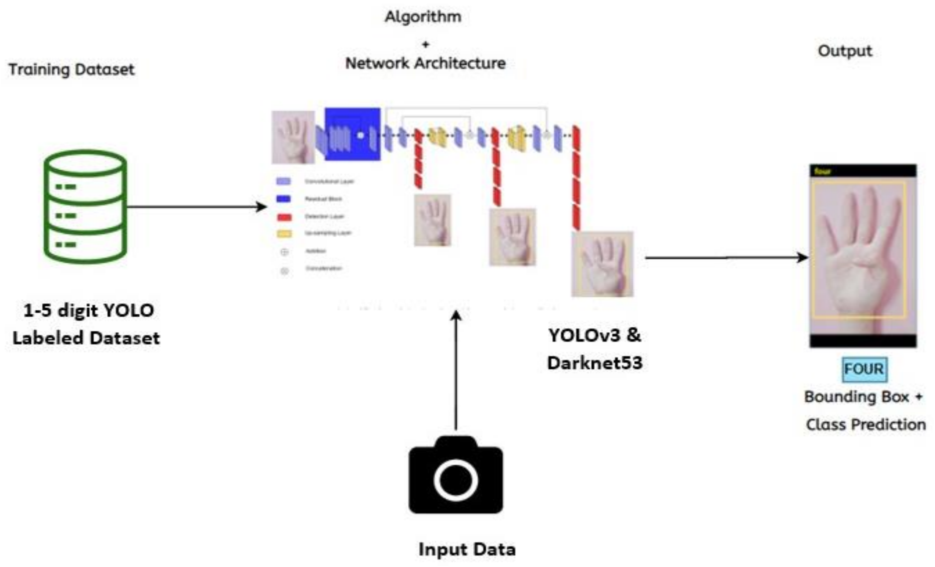

Figure 5. In-depth architectural flow of how the application deals with the recognition and detection of the input image with a gesture using YOLOv3 network.

We changed the configuration of YOLO and defined the activation according to the stride and the pad. For the YOLOv3 model, we set the mask to 0.5. The learning rate was set to 0.001, and the value of jitter was set to 0.3.

For the exploratory study of the best alternatives to implement the gesture recognition system, we use some other models to check which algorithm works best for gesture detection which will be discussed in the next section.

The developed neural network model is summarized in Table 2. Our input layer is just a typical CNN, which has convolutional layers, and other than that, it has a special layer, which is a max-pooling layer, and a very simple layer, which is the output layer. The general architecture of the application, which includes both the training and testing set, can be seen in Figure 6.

Table 2. Model Summary.

Layer (Type)	Output Shape	Parameters
Conv2d_133 (Conv2D)	(None, 126, 254, 64)	1792
Max_pooling_132 (Max Pooling)	(None, 63, 127, 64)	0
Conv2d_134 (Conv2D)	(None, 61, 125, 64)	36928
Max_pooling_133 (Max Pooling)	(None, 20, 41, 64, 0)	0
Conv2d_135 (Conv2D)	(None, 18, 39, 64)	36928
Max_pooling_134 (Max Pooling)	(None, 6, 13, 64)	0
Conv2d_136 (Conv2D)	(None, 4, 11, 64)	36928
Max_pooling_135 (Max Pooling)	(None, 1, 3, 64)	0
Flatten_33 (Flatten)	(None, 192)	0
Dense_151 (Dense)	(None, 128)	24704
Dropout_36 (Dropout)	(None, 128)	0
Dense_152 (Dense)	(None, 64)	8256
Dense_153 (Dense)	(None, 32)	2080
Dense_154 (Dense)	(None, 8)	264

Total params: 147,880. Trainable params: 147,880. Non-trainable params: 0.

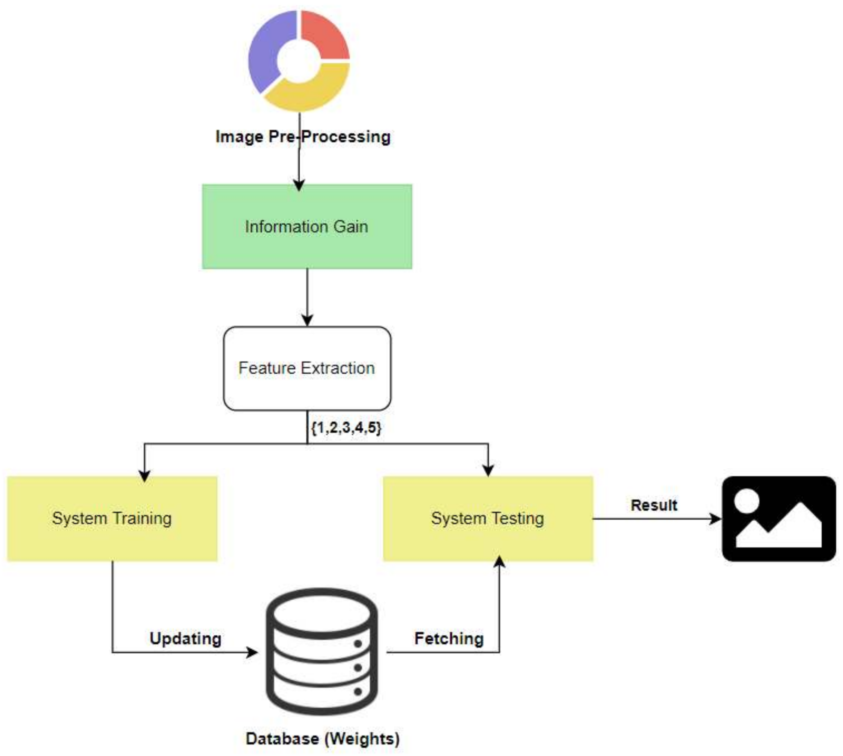

Figure 6. General architecture of the developed hand gesture detection system.

4. Results

In this section, we present, discuss, and evaluate our results.

4.1. Environmental Setup

To carry out this experiment Python 3.7 was used to train the algorithm on the personal computer with local 4GB GPU. Other important parameters are presented in Table 3. The training of the neural network model on our dataset took more than 26 h on a GPU of 24 GB.

Table 3. Parameters and values for environment setup.

Hyper Parameter	Value
Learning Rate	0.001
Epochs	30
No. of Classes	5
Algorithm	DarkNet-53 and YOLOv3
Optimizer	Adam
Activation	Linear, Leaky
Filter Size	[64,128,256,512,1024]
Mask	0–8
Decay	0.0005

4.2. Results and Performance of YOLOv3

Figure 7 shows the output of the developed application that performs real-time hand gesture recognition. As you can see, the bounding box is a little bit large, which is because it was left intentionally as 416 × 416 for covering the maximum part, then scattered, and all unnecessary information was removed from the image so we could obtain a larger coverage area. This also helps in the zoom case when the object is too large, as it will cleverly identify which gesture is being represented, and it then returns the class ID with the best match.

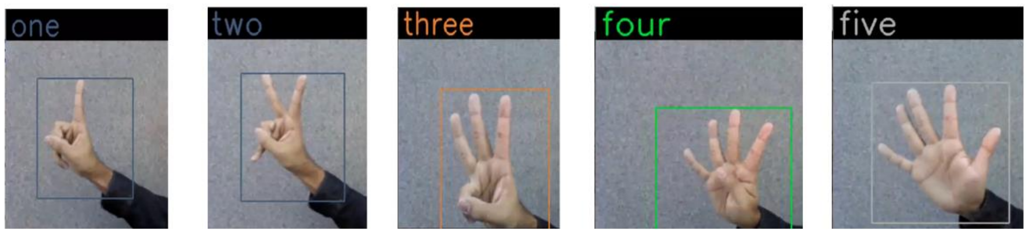

Figure 7. Gesture detected using the proposed YOLOv3-based methodology.

Figure 8 shows the training performance of the proposed deep learning model. The results are 98% correct, as measured by experiments. The experiment is in real time because we trained our model with YOLO and Pascal VOC [46] configurations and tested it with live images. We added the YOLO-annotated labels into a CSV file and re-ran training tests to confirm the accuracy of the model by plotting the curve, as we can find the accuracy of an individual gestation in the real-time experiment. The fault was seen in the transitions between one gesture to another, because the application is in real-time, so the system can detect dynamic objects too, as proven by our experiments.

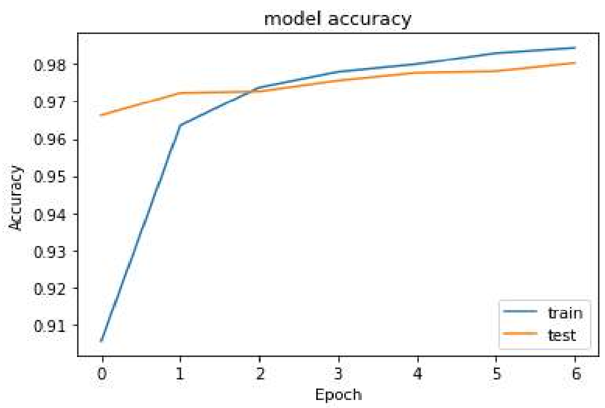

Figure 8. The performance of the proposed model.

We used several different algorithms to test which was the best method for the gesture recognition application. We compared YOLOv3 [47] with other deep learning models

(VGG16 [48] and SSD [49]). In the end, YOLOv3 produced the best results and was chosen, with an accuracy of 97.68% during training and an outstanding 96.2% during testing. DarkNet-53 was used to train the dataset, and for the detection, YOLOv3 was used. As we know, in real-time experiments it is not quite possible to replicate the gestures exactly, so in order to determine the accuracy of the model, so we generated a CSV file of a few YOLO-annotated labels and re-ran the code with some modifications for the accuracy curve.

Table 4 explains the accuracy of the individual models, where the stochastic gradient descent gave the lowest value because the real-time motion stochastic could not identify moving objects correctly, compared to the other algorithms. We evaluated the performance of deep learning models using the precision, recall, F-1 score, and accuracy measures, achieving an accuracy of 97.68% for YOLOv3.

Table 4. Comparison of the accuracy results for the deep learning models used for hand gesture recognition. Best results are shown in bold.

Models	Learning Rate	Images	Precision (%)	Recall (%)	F-1 Score (%)	Accuracy (%)
VGG16	0.001	216	93.45	87.45	90.3	85.68
SGD	0.1	216	70.95	98.37	82.4	77.98
SSD	0.0001	216	91.25	84.30	87.6	82.00
YOLOv3	0.001	216	94.88	98.68	96.7	97.68

Table 5 shows the comparison of the state-of-the-art works of Chen et al. [28], Nyirarugira et al. [31], Albawi et al. [32], Fong et al. [50], Yan et al. [51], and Ren et al. [52] with our proposed model, which produced better results with higher accuracy.

Table 5. Comparison of the state-of-the-art works with our proposed model.

Reference	Model	Dataset	Accuracy (%)
Chen et al. [28]	Labeling Algorithm Producing Palm Mask	1300 images	93%
Nyirarugira et al. [31]	Particle Swarm Movement (PSO), Longest Common Subsequence (LCS)	Gesture vocabulary	86%
Albawi et al. [32]	Random Forest (RF) and Boosting Algorithms, Decision Tree Algorithm	7805 gestures frames	63%
Fong et al. [50]	Model Induction Algorithm, K-star Algorithm, Updated Naïve Bayes Algorithm, Decision Tree Algorithm	50 different attributes total of 9000 data instance 7 videos	76%
Yan et al. [51]	AdaBoost Algorithm, SAMME Algorithm, SGD Algorithm, Edgebox Algorithm	5500 images for the testing and 5500 for the training set	81.25%
Ren et al. [52]	FEMD Algorithm, Finger Detection Algorithm, Skeleton-Based Matching	1000 cases	93.20%
Proposed model	YOLOv3	216 images (Train) 15 images (Test)	97.68%

5. Discussion

Real-time hand gesture recognition based on deep learning models has critical roles in many applications due to being one of the most advanced domains, in which the computer vision and artificial intelligence methods have helped to improve communication with deaf people, but also to support the gesture-based signaling systems]. In this study, we experimented with hand gesture recognition using YOLOv3 and DarkNet-53 deep learning network models [47]. The dataset, which we used, was collected, labeled, and trained by ourselves. We compared the performance of YOLOv3 with several other state-of-the-art algorithms, which can be seen in Table 5. The achieved results were good, as YOLOv3 achieved better results when compared to other state-of-the art algorithms. However, our proposed approach was not tested on the YOLO-LITE model. The YOLOv3 model was trained on a YOLO-labeled dataset with DarkNet-53. YOLOv3 has more tightness when it comes to bounding boxes and generally is more accurate than YOLO-LITE [53].

Moreover, the aim of our study was not to apply real-time hand gestures on limited computing power devices, since we collected hand images of different size and used different angles for diversity. Hence, we focused only on performance criteria rather than YOLO-LITE [54] criteria to recognize hand gestures of speed. Complex applications such as communication with deaf people, gesture-based signaling systems [55], sign language recognition [4], special signal languages used in sports [56], human action recognition [7], posture detection [57], physical exercise monitoring [10], and controlling smart homes for assisted living [58], are where GPUs could perform better through YOLOv3.

In future work we can apply mixed YOLOv3–LITE [59] on our datasets and new datasets of 1–10 numbers for all kind of applications for GPU and non-GPU based computers to achieve real-time object detection precisely and quickly. Furthermore, we can have enhanced our images through oversampling and real-time augmentation [60] as well. The major contribution is that there is no such dataset available in YOLO-labeled format and the research on, specifically, YOLOv3 and onwards requires the YOLO-labeled dataset so by our contribution that dataset will be readily available for future research and improvement as the domain of hand gesture recognition is very much wide there is a need of dataset that should be readily available in the YOLO format also.

6. Conclusions

In this paper, we have proposed a lightweight model based on the YOLOv3 and DarkNet-53 deep learning models for hand gesture recognition. The developed hand gesture recognition system detects both real-time objects and gestures from video frames with an accuracy of 97.68%. Despite the accuracy obtained there is still room for improvement in the following model, as right now the model proposed detects static gestures. However, the model can be improved for detecting multiple gestures and can be improved by detecting more than one gesture at a time. The proposed method can be used for improving assisted living systems, which are used for human–computer interaction both by healthy and impaired people. Additionally, we compared the performance and execution of the YOLOv3 model with different methods and our proposed method achieved better results by extracting features from the hand and recognized hand gestures with the accuracy, precision, recall, and F-1 score of 97.68, 94.88, 98.66, and 96.70%, respectively. For future work, we will be focusing on hybrid methods with smart mobile applications or robotics with different scenarios. Furthermore, we will design a more advanced convolution neural network with data fusion, inspired by recent works [61–63], to enhance the precision of hand gesture recognition.

Author Contributions: Conceptualization, A.M., M.J.A., A.Y., M.A.M., R.D. and K.H.A.; methodology, A.M., M.J.A., A.Y., M.A.M., R.D. and K.H.A.; software, A.M., M.J.A. and A.Y.; validation, M.J.A.; formal analysis, M.A.M.; writing—original draft preparation, A.M., M.J.A. and A.Y.; writing—review and editing, A.M, M.J.A., A.Y., M.A.M., R.D., R.M. and K.H.A.; supervision, M.A.M.; funding acquisition, R.D. and R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable for this study.

Informed Consent Statement: Not applicable for this study.

Data Availability Statement: The dataset used is collected and labeled by the authors and is available at Google Drive at: https://drive.google.com/drive/folders/1X4q59iAUcPDY9lsJaq2HMb0GLU6cKp-q?usp=sharing (accessed on 1 February 2021).

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

References

1. Fang, Y.; Wang, K.; Cheng, J.; Lu, H. A Real-Time Hand Gesture Recognition Method. In Proceedings of the Multimedia and Expo, IEEE International Conference On Multimedia and Expo, Beijing, China, 2–5 July 2007. [CrossRef]
2. Oudah, M.; Al-Naji, A.; Chahl, J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *J. Imaging* **2020**, *6*, 73. [CrossRef]
3. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Alrayes, T.S.; Mekhtiche, M.A. Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. *IEEE Access* **2020**, *8*, 192527–192542. [CrossRef]
4. Vaitkevičius, A.; Taroza, M.; Blažauskas, T.; Damaševičius, R.; Maskeliūnas, R.; Woźniak, M. Recognition of american sign language gestures in a virtual reality using leap motion. *Appl. Sci.* **2019**, *9*, 445. [CrossRef]
5. Rezende, T.M.; Almeida, S.G.M.; Guimarães, F.G. Development and validation of a brazilian sign language database for human gesture recognition. *Neural Comput. Appl.* **2021**. [CrossRef]
6. Žemgulys, J.; Raudonis, V.; Maskeliūnas, R.; Damaševičius, R. Recognition of basketball referee signals from real-time videos. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 979–991. [CrossRef]
7. Afza, F.; Khan, M.A.; Sharif, M.; Kadry, S.; Manogaran, G.; Saba, T.; Ashraf, I.; Damaševičius, R. A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. *Image Vision Comput.* **2021**, *106*, 104090. [CrossRef]
8. Nikolaidis, A.; Pitas, I. Facial feature extraction and pose determination. *Pattern Recognit.* **2000**, *33*, 1783–1791. [CrossRef]
9. Kulikajevas, A.; Maskeliunas, R.; Damaševičius, R. Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ Comput. Sci.* **2021**, *7*, e442. [CrossRef] [PubMed]
10. Ryselis, K.; Petkus, T.; Blažauskas, T.; Maskeliūnas, R.; Damaševičius, R. Multiple kinect based system to monitor and analyze key performance indicators of physical training. *Hum. Centric Comput. Inf. Sci.* **2020**, *10*, 51. [CrossRef]
11. Huu, P.N.; Minh, Q.T.; The, H.L. An ANN-based gesture recognition algorithm for smart-home applications. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 1967–1983. [CrossRef]
12. Abraham, L.; Urru, A.; Normani, N.; Wilk, M.P.; Walsh, M.; O'Flynn, B. Hand Tracking and Gesture Recognition Using Lensless Smart Sensors. *Sensors* **2018**, *18*, 2834. [CrossRef]
13. Ahmed, S.; Cho, S.H. Hand Gesture Recognition Using an IR-UWB Radar with an Inception Module-Based Classifier. *Sensors* **2020**, *20*, 564. [CrossRef]
14. Lee, D.-H.; Kwang-Seok Hong, K.-S. Game interface using hand gesture recognition. In Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology, Seoul, Korea, 30 November–2 December 2010. [CrossRef]
15. Alkemade, R.; Verbeek, F.J.; Lukosch, S.G. On the efficiency of a VR hand gesture-based interface for 3D object manipulations in conceptual design. *Int. J. Hum. Comput. Interact.* **2017**, *33*, 882–901. [CrossRef]
16. Lee, Y.S.; Sohn, B. Immersive gesture interfaces for navigation of 3D maps in HMD-based mobile virtual environments. *Mob. Inf. Syst.* **2018**, *2018*, 2585797. [CrossRef]
17. Del Rio Guerra, M.S.; Martin-Gutierrez, J.; Acevedo, R.; Salinas, S. Hand gestures in virtual and augmented 3D environments for down syndrome users. *Appl. Sci.* **2019**, *9*, 2641. [CrossRef]
18. Moschetti, A.; Fiorini, L.; Esposito, D.; Dario, P.; Cavallo, F. Toward an unsupervised approach for daily gesture recognition in assisted living applications. *IEEE Sens. J.* **2017**, *17*, 8395–8403. [CrossRef]
19. Mezari, A.; Maglogiannis, I. An easily customized gesture recognizer for assisted living using commodity mobile devices. *J. Healthc. Eng.* **2018**, *2018*, 3180652. [CrossRef] [PubMed]
20. Negin, F.; Rodriguez, P.; Koperski, M.; Kerboua, A.; Gonzàlez, J.; Bourgeois, J.; Bremond, F. PRAXIS: Towards automatic cognitive assessment using gesture recognition. *Expert Syst. Appl.* **2018**, *106*, 21–35. [CrossRef]
21. Kaczmarek, W.; Panasiuk, J.; Borys, S.; Banach, P. Industrial robot control by means of gestures and voice commands in off-line and on-line mode. *Sensors* **2020**, *20*, 6358. [CrossRef]
22. Neto, P.; Simão, M.; Mendes, N.; Safeea, M. Gesture-based human-robot interaction for human assistance in manufacturing. *Int. J. Adv. Manuf. Technol.* **2019**, *101*, 119–135. [CrossRef]
23. Young, G.; Milne, H.; Griffiths, D.; Padfield, E.; Blenkinsopp, R.; Georgiou, O. Designing mid-air haptic gesture controlled user interfaces for cars. In Proceedings of the ACM on Human-Computer Interaction, 4(EICS), Article No. 81, Honolulu, HI, USA, 25–30 April 2020. [CrossRef]
24. Yu, H.; Fan, X.; Zhao, L.; Guo, X. A novel hand gesture recognition method based on 2-channel sEMG. *Technol. Health Care* **2018**, *26*, 205–214. [CrossRef] [PubMed]
25. Zhao, L.; Li, S. Object detection algorithm based on improved YOLOv3. *Electronics* **2020**, *9*, 537. [CrossRef]
26. Kulikajevas, A.; Maskeliūnas, R.; Damaševičius, R.; Ho, E.S.L. 3D object reconstruction from imperfect depth data using extended yolov3 network. *Sensors* **2020**, *20*, 2025. [CrossRef]
27. Ni, Z.; Chen, J.; Sang, N.; Gao, C.; Liu, L. Light YOLO for High-Speed Gesture Recognition. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018. [CrossRef]

28. Chen, L.; Fu, J.; Wu, Y.; Li, H.; Zheng, B. Hand Gesture Recognition Using Compact CNN via Surface Electromyography Signals. *Sensors* **2020**, *20*, 672. [CrossRef] [PubMed]
29. Colli-Alfaro, J.G.; Ibrahim, A.; Trejos, A.L. Design of User-Independent Hand Gesture Recognition Using Multilayer Perceptron Networks and Sensor Fusion Techniques. In Proceedings of the IEEE 16th International Conference on Rehabilitation Robotics (ICORR), Toronto, ON, Canada, 24–28 June 2019; pp. 1103–1108. [CrossRef]
30. Elmezain, M.; Al-Hamadi, A.; Appenrodt, J.; Michaelis, B. A hidden markov model-based isolated and meaningful hand gesture recognition. *Int. J. Electr. Comput. Syst. Eng.* **2009**, *3*, 156–163.
31. Nyirarugira, C.; Choi, H.-R.; Kim, J.; Hayes, M.; Kim, T. Modified levenshtein distance for real-time gesture recognition. In Proceedings of the 6th International Congress on Image and Signal Processing (CISP), Hangzhou, China, 16–18 December 2013. [CrossRef]
32. Albawi, S.; Bayat, O.; Al-Azawi, S.; Ucan, O.N. Social Touch Gesture Recognition Using Convolutional Neural Network. *Comput. Intell. Neurosci.* **2018**, 1–10. [CrossRef]
33. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The Application of Improved YOLO V3 in Multi-Scale Target Detection. *Appl. Sci.* **2019**, *9*, 3775. [CrossRef]
34. Saqib, S.; Ditta, A.; Khan, M.A.; Kazmi, S.A.R.; Alquhayz, H. Intelligent dynamic gesture recognition using CNN empowered by edit distance. *Comput. Mater. Contin.* **2020**, *66*, 2061–2076. [CrossRef]
35. Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Mekhtiche, M.A. Hand gesture recognition for sign language using 3DCNN. *IEEE Access* **2020**, *8*, 79491–79509. [CrossRef]
36. Do, N.; Kim, S.; Yang, H.; Lee, G. Robust hand shape features for dynamic hand gesture recognition using multi-level feature LSTM. *Appl. Sci.* **2020**, *10*, 6293. [CrossRef]
37. Elboushaki, A.; Hannane, R.; Afdel, K.; Koutti, L. MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. *Expert Syst. Appl.* **2020**, *139*. [CrossRef]
38. Peng, Y.; Tao, H.; Li, W.; Yuan, H.; Li, T. Dynamic gesture recognition based on feature fusion network and variant ConvLSTM. *IET Image Process.* **2020**, *14*, 2480–2486. [CrossRef]
39. Tan, Y.S.; Lim, K.M.; Lee, C.P. Hand gesture recognition via enhanced densely connected convolutional neural network. *Expert Syst. Appl.* **2021**, *175*. [CrossRef]
40. Tran, D.; Ho, N.; Yang, H.; Baek, E.; Kim, S.; Lee, G. Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network. *Appl. Sci.* **2020**, *10*, 722. [CrossRef]
41. Rahim, M.A.; Islam, M.R.; Shin, J. Non-Touch Sign Word Recognition Based on Dynamic Hand Gesture Using Hybrid Segmentation and CNN Feature Fusion. *Appl. Sci.* **2019**, *9*, 3790. [CrossRef]
42. Mambou, S.; Krejcar, O.; Maresova, P.; Selamat, A.; Kuca, K. Novel Hand Gesture Alert System. *Appl. Sci.* **2019**, *9*, 3419. [CrossRef]
43. Ashiquzzaman, A.; Lee, H.; Kim, K.; Kim, H.-Y.; Park, J.; Kim, J. Compact Spatial Pyramid Pooling Deep Convolutional Neural Network Based Hand Gestures Decoder. *Appl. Sci.* **2020**, *10*, 7898. [CrossRef]
44. Benitez-Garcia, G.; Prudente-Tixteco, L.; Castro-Madrid, L.C.; Toscano-Medina, R.; Olivares-Mercado, J.; Sanchez-Perez, G.; Villalba, L.J.G. Improving Real-Time Hand Gesture Recognition with Semantic Segmentation. *Sensors* **2021**, *21*, 356. [CrossRef] [PubMed]
45. Bradski, G. The OpenCV Library. *Dr Dobb's J. Softw. Tools* **2000**, *25*, 120–125.
46. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
47. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
48. Qassim, H.; Verma, A.; Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 169–175. [CrossRef]
49. Fu, C.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
50. Fong, S.; Liang, J.; Fister, I.; Fister, I.; Mohammed, S. Gesture Recognition from Data Streams of Human Motion Sensor Using Accelerated PSO Swarm Search Feature Selection Algorithm. *J. Sens.* **2015**, *2015*, 205707. [CrossRef]
51. Yan, S.; Xia, Y.; Smith, J.S.; Lu, W.; Zhang, B. Multiscale Convolutional Neural Networks for Hand Detection. *Appl. Comput. Intell. Soft Comput.* **2017**, *2017*, 9830641. [CrossRef]
52. Ren, Z.; Yuan, J.; Meng, J.; Zhang, Z. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. *IEEE Trans. Multimed.* **2013**, *15*, 1110–1120. [CrossRef]
53. Pedoeem, J.; Huang, R. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. In Proceedings of the IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2503–2510.
54. Sismananda, P.; Abdurohman, M.; Putrada, A.G. Performance Comparison of Yolo-Lite and YoloV3 Using Raspberry Pi and MotionEyeOS. In Proceedings of the 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 4–5 August 2020; pp. 1–7.
55. Połap, D. Human-machine interaction in intelligent technologies using the augmented reality. *Inf. Technol. Control* **2018**, *47*, 691–703. [CrossRef]
56. Žemgulys, J.; Raudonis, V.; Maskeliunas, R.; Damaševičius, R. Recognition of basketball referee signals from videos using histogram of oriented gradients (HOG) and support vector machine (SVM). *Procedia Comput. Sci.* **2018**, *130*, 953–960. [CrossRef]

57. Wozniak, M.; Wieczorek, M.; Silka, J.; Polap, D. Body pose prediction based on motion sensor data and recurrent neural network. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2101–2111. [CrossRef]
58. Maskeliunas, R.; Damaševicius, R.; Segal, S. A review of internet of things technologies for ambient assisted living environments. *Future Internet* **2019**, *11*, 259. [CrossRef]
59. Zhao, H.; Zhou, Y.; Zhang, L.; Peng, Y.; Hu, X.; Peng, H.; Cai, X. Mixed YOLOv3-LITE: A Lightweight Real-Time Object Detection Method. *Sensors* **2020**, *20*, 1861. [CrossRef]
60. Awan, M.J.; Rahim, M.S.M.; Salim, N.; Mohammed, M.A.; Garcia-Zapirain, B.; Abdulkareem, K.H. Efficient Detection of Knee Anterior Cruciate Ligament from Magnetic Resonance Imaging Using Deep Learning Approach. *Diagnostics* **2021**, *11*, 105. [CrossRef] [PubMed]
61. Mastoi, Q.; Memon, M.S.; Lakhan, A.; Mohammed, M.A.; Qabulio, M.; Al-Turjman, F.; Abdulkareem, K.H. Machine learning-data mining integrated approach for premature ventricular contraction prediction. *Neural Comput. Appl.* **2021**. [CrossRef]
62. Mohammed, M.A.; Abdulkareem, K.H.; Mostafa, S.A.; Ghani, M.K.A.; Maashi, M.S.; Garcia-Zapirain, B.; Oleagordia, I.; Alhakami, H.; Al-Dhief, F.T. Voice pathology detection and classification using convolutional neural network model. *Appl. Sci.* **2020**, *10*, 3723. [CrossRef]
63. Kashinath, S.A.; Mostafa, S.A.; Mustapha, A.; Mahdin, H.; Lim, D.; Mahmoud, M.A.; Mohammed, M.A.; Al-Rimy, B.A.S.; Fudzee, M.F.M.; Yang, T.J. Review of Data Fusion Methods for Real-Time and Multi-Sensor Traffic Flow Analysis. *IEEE Access* **2021**, *9*, 51258–51276. [CrossRef]