Sigma J Eng Nat Sci, Vol. 42, No. 1, pp. 99−115, February, 2024

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2024.00012

Research Article

Comparative analysis on real-time hand gesture and sign language recognition using convexity defects and YOLOv3

Md KHALİLUZZAMAN[1,*], Khadijatul KOBRA[1], Shabnaj LİAQAT[1], Shahidul Islam KHAN[1]

[1]*Department of CSE, International Islamic University Chittagong, Chattogram, 4318, Bangladesh*

ARTICLE INFO

Article history
Received: 12 December 2021
Accepted: 24 July 2022

Keywords:
Hand Gesture Recognition; Convexity Defect; Feature Extraction; Deep Learning; YOLO

ABSTRACT

The purpose of this paper is to help people with auditory and speech disabilities to communicate with others and for controlling computers and machines. This paper proposes two different methods for identifying six distinctive hand gestures and sign language for divergent environmental conditions. The first method is based on the hand feature extraction i.e., convexity defects. For that, initially, the hand region is detected by HSV skin color conversion process. Contour and convex hull of hand are extracted from the hand region. Finally, convexity defects are determined to identify the hand gestures. The second method is deep learning based YOLOv3 model that uses DARKNET-53 convolutional neural network (CNN) as its backbone. The model is trained on a large annotated dataset. Experimental results reveal that the deep-leaning method outperforms the hand feature approach and obtain 98.92% and 95.57% accuracy for deep learning and hand feature-based model respectively.

Cite this article as: Khaliluzzaman M, Kobra K, Liaqat S, Khan SI. Comparative analysis on real-time hand gesture and sign language recognition using convexity defects and YOLOv3. Sigma J Eng Nat Sci 2024;42(1):99−115.

INTRODUCTION

Human-Computer Interaction (HCI) is a subject that links computer science and language technology to decipher human gestures with the help of mathematical algorithms [1]. Using vision-based system, many works has been done on real time hand gesture recognition. From any physical movement or position, gestures can precede. Hand gesture applications require the end-user to be an expert and well qualified at operating and interpreting the object of different gestures. Hand Gesture detection can be used for a far-reaching number of purposes, namely Sign Language Recognition, Directional hint through pointing, for socially assistive robotics, Remote control, Alternative computer interfaces, Immersive Game technology, Virtual Controllers, Affective Computing. Hand Gesture Recognition is a crucial Human-Computer Interface (HCI) for intercommunication between living beings and machine systems [2]. So to construct HCI more traditional and polite, it would be advantageous to give computers the ability to identify conditions in the same manner as a human. Approaches of this research work are proposed as being motivated to build a more robust and reliable hand gesture recognizing system,

***Corresponding author.**
*E-mail address: khalil@iiuc.ac.bd; khalilcse021@gmail.com; khalil_021@yahoo.co.in; khalil@cse.iiuc.ac.bd

This paper was recommended for publication in revised form by Regional Editor Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

to establish a hand gesture detector that can be used for several environmental surroundings and most of the detection accuracies are not much satisfactory.

Of all the AI techniques, machine learning is most commonly used for big data processing [3]. It is a self-adapting algorithm that gets increasingly better analysis and patterns with experience or newly added data. Deep learning is an efficient tool in handling raw data, which also can get the needed characteristics for a specific identification task. Currently, deep learning algorithms are largely used for object detection as they show great accuracy and speed when trained with large datasets. YOLOv3 [4], a recent modification of a current detector algorithm -YOLO- You Only Look Once, is such an algorithm.

This work sets on recognizing six hand gestures in four different environmental conditions and aims to do so while having the maximum speed and efficiency for both the models. These Recognized gestures will help in communicating with deaf people as well as allow the computer to give commands. The analysis of hand gestures used in this paper is explained in the block diagram exhibited in Figure 1.

The first mentioned model converts the acquired RGB image frame to HSV skin color image and obtains the hand's region of interest (ROI). After thresholding, contour and convex hull of hand is to be detected. Afterward, the convexity defect of the hand is computed. By calculating the number of convexity defects and area of the hand, hand gesture is recognized.

In the deep learning-based model i.e., YOLOv3, images are passed to the convolutional neural network (CNN) after they are labelled with appropriate bounding boxes using data annotation tool. Records including values of bounding boxes and image names of annotated images

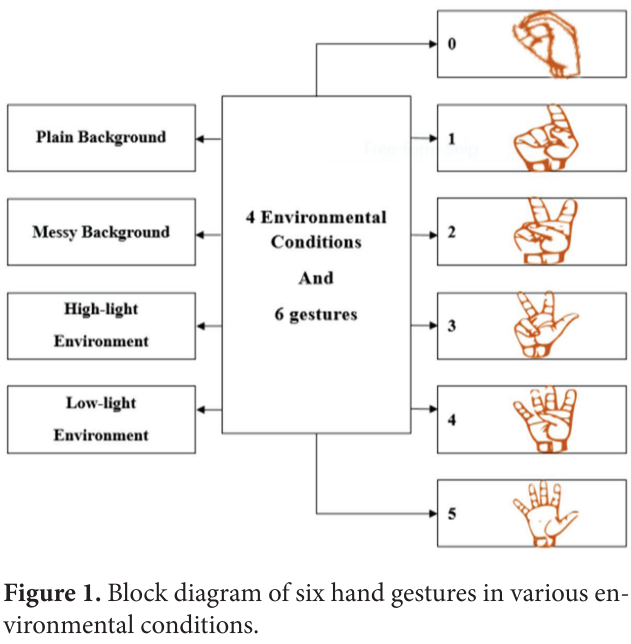


Figure 1. Block diagram of six hand gestures in various environmental conditions.

are saved in text format which is used for training the model. Transfer learning [5] is used in this model as it is the most useful method for training models on similar object detections. It is where the weights of the pre-trained models are used as the starting points for training model [6]. This reduces the time for training process, lowers the generalization errors and increases classification accuracy. Through transfer learning on the custom dataset, CNN is prepared for the desire output. Later in the testing session, the YOLOv3 model creates an SXS grid on the given image. The output of the model encoded candidate bound boxes from three different scales and the boxes defined the context of the anchor box, thoroughly chosen based on the object size analysis in the dataset. Any anchor box that does not confidently define an object i.e., the probability of all classes below a threshold value is disregarded. Finally, to find the best-fitted box among all remained bounding box, non-maximum suppression (NMS) [7] is applied.

The main purpose of this study is to propose two methods based on hand gesture recognition to help people with disabilities easier to communicate with the world. They can also be used for many applications, such as controlling robots. The further sections are discussed as follow: related works are discussed in section 2, the two proposed methods on real-time hand gesture and sign language recognition using the hand feature and deep learning model respectively are described in section 3, experimental outputs of both the methods are presented in section 4 and section 5 consists of the conclusion of the paper.

RELATED WORKS

Hand gesture applications require the end-user to be an expert and well qualified at operating and interpreting the object of different gestures. There are several viable quantities of hand gestures, hence, for specific applications; a collection of gestures is used to complete its procedures. Hand gesture consists of unique characteristics of the different hand movements. In this regard, the feature extraction processes should be capable of obtaining features from the critical angle of the hand. Many researchers utilized these difficult characteristics to recognize hand gestures from different environmental conditions. Many researchers used the hand-crafted feature and deep learning-based algorithms to recognize hand gesture in the last few decades.

Works Related to Hand Feature-based Method

According to Z. Ren et al. [8], it converges on developing a robust part-based hand gesture recognition model with the help of a Kinect sensor. They have used finger detection and distance metric called Finger-Earth Mover's Distance.The proposed gesture recognition technique shows an average accuracy of 93.2% for a dataset of 10 different gestures. But the research has used a very complex method to recognize gestures. The method will become more complex on more subjects. In the work of X. Li et al.

[9], a Fuzzy C-Means (FCM) clustering method is applied. The aimed method does the whole process by dividing it into four main parts. It operates on a training-classification approach. The paper obtains recognition accuracy of 85.83%. Though the described method shows good gains, the identification efficiency decreases almost immediately when exposed to bright light or when the gap separating the end-user and the camera is higher than 1.5 meters. S. Sharma et al. [10] proposes a useful method in favor of blind peoples by combining two recognition systems that are hand gestures using YCbCr color space and face recognition system using Haar Cascade Classifiers. An accuracy of 95.2% is obtained on hand gestures. However, it recognizes only few hand gestures and it has not been tested with different backgrounds. L. Yun et al. [11] Introduces a hand gesture identification system that uses multi-feature fusion and template matching. This approach identifies the hand-shaped contour area and samples by taking the maximum contour based on skin color feature by extracting angle count, skin color angle and non-skin color angle, combining the Hue invariant moments features of the largest hand-shaped area. The technique of matching Euclidean distance templates for hand gesture classification and recognition is applied. But the system is tested in complex backgrounds with only bright light. According to A. Pradhan et al. [12], an easy yet effective way for identifying the hand gesture is by representing the active and in-active fingers using the distinct sequence of binary value 0 and 1 respectively for each gesture. Although using a binary pattern accelerates the effectiveness of the classification process, the model is not trained for different environmental conditions and the input image for testing has to be captured from the same distance as the training phase. To draw out Human-Computer Interaction specially to do automated mouse controller actions, R. M. Prakash et al. [13] presented an immediate gesture recognition with a fingertip detection algorithm. To detect the hand area region, growing segmentation is applied and the exposure of fingertip was done using convex hull. This model succeeds in finding the fingertips efficiently along with identifying five distinct signs. But the model is not trained for different backgrounds and light levels. This paper by Y. Xu et al. [14] aims to use contour and convex defects to locate the fingertips for recognizing the static hand gestures one to five. Euclidean distance metric was chosen for the comparison analysis. The limitation of this paper is that it does not figure out the vertical rotation of the hand gesture which is why most of the number 4 gestures are detected as number 3 by mistake.

Works Related to Deep Learning-based Method

As stated by S. Hussain et al. [15], a vision-based hand gesture identification technique was used to recognize eleven gestures. The process was strengthened for both static and dynamic gestures with VGG16 - a CNN architecture as the pre-trained model and shows an accuracy of 93.09%. However, the model's accuracy drops when the images is taken under bright light condition. As said by Q. Zhang et al. [16], this proposed work suggested a continuous dynamic recognition algorithm of four hand gestures formed on Channel State Information (CSI) and YOLOv3 with the average recognition accuracy of 94%. Data collection uses a CSI-based radio frequency method. To extract gray value images, adaptive weighted fusion, Kalman filtering, threshold segmentation, and data conversion are used. Finally, grayscale images are employed to train and classify the YOLOv3 object detection algorithm. But the system is trained to identify only four gestures. D. Jiang et al. [17] uses the skeletonization algorithm which lessens the shooting angle and the impact of the environment on the recognition effect and improves the accuracy of gesture recognition in complex environments. The proposed model was trained using the ASK gesture database. The test outcomes reveal that this method outruns the SVM method, dictionary learning + sparse representation, CNN method, and other methods by 96.01%. This work shows that CNN is a reliable pick for gesture recognition. Yet the hindrance of the system is that the targeted user ought to have a Kinetic sensor to be able to read the gesture. In the work of B. Benjdira et al. [18], cars were detected using two latest CNN algorithms - Faster R-CNN and YOLOv3. The work shows that though these two CNN models are similar in the precision metric, yet in case of sensitivity and processing time YOLOv3 beats Faster R-CNN. The data for this system can be increased by adding lighting interfaces such as - morning, evening, night as well as various environmental circumstances such as – crowded traffic, winter, summer, etc. The author U. Tanmaie et al. [19] uses YOLOv2 to detect and classify hand gestures and ResNet-50 as feature extractor. The model is trained and tested with 2750 images for 10 different gestures. And it gives an accuracy of 100% and 97.46% with images that have zero noise and human noise respectively. In this paper, for some similar hand postures, the detection accuracy decreases.

Although region-based methods show great efficiency, it takes a long time to identify objects. Still, the detection speed of YOLOv3 is above all. There are numerous ways to recognize gestures using deep learning. However, in terms of speed and accuracy, YOLOv3 is better than all existing methods.

METHODOLOGY

This study focuses on two different proposed methods which can recognize gestures in four environmental surroundings. The first method is based on image processing technique where it uses HSV (Hue Saturation Value) color space to detect the hand region. Gaussian blur is used to eliminate noises from the frame. Once the hand is detected and features of contour and convex hull are extracted, numbers of convexity defects are obtained. Finally, the hand gesture is recognized based on convexity defects. The second method is based on YOLOv3 model that uses

DARKNET-53 convolutional neural network as its backbone. The model is trained on large, annotated dataset and can recognize gestures with a high accuracy.

Proposed Method for Hand Gesture Recognition Using Hand Feature-based Method

The proposed model of recognizing hand gestures using image processing follows an algorithmic way. In image processing technique, the gesture is recognized using convexity defect. Images that are taken as input are captured by the webcam. A frame is allocated for the input image within which the presence of hand will be scanned. This will capture the region of interest (ROI). The proposed methodology is described in this section. It consists of six steps, i.e. (1) input image frame, (2) detecting hand's ROI using HSV color model, (3) preprocessing, (4) detecting contour and convex hull by thresholding, (5) feature extraction based on area and convexity defects, and (6) classifying the gesture based on the extracted features. The workflow of the proposed method is illustrated in Figure 2.

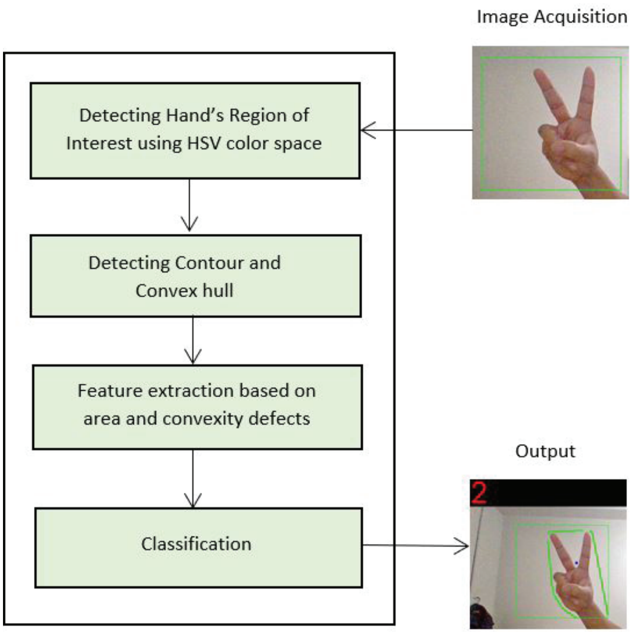


Figure 2. Workflow diagram of the proposed hand feature model using convexity defect.

Image Acquisition

Firstly, video is captured through webcam. In this method, the data is collected in real-time using the webcam. The hand is to be kept in a rectangle frame of shape, i.e., 300 × 300. The video is captured in four different backgrounds. Some of the input images that are captured from input frame are shown in Figure 3.

Detecting Hand's Region of Interest Using HSV Color Space

For skin detection, input image frame is converted to HSV color space. Maximum and minimum values of R, G, B are found and their difference D is calculated.

$$d_{max} = \max(R, G, B) \tag{1}$$

$$d_{min} = \min(R, G, B) \tag{2}$$

$$D = d_{max} - d_{min} \tag{3}$$

Hue is calculated by utilizing the equation of (4) to (7).

$$\text{If } d_{max} = 0,\ H = 0 \tag{4}$$

$$\text{If } d_{max} = R,\ H = 60 \times \frac{G-B}{D} \tag{5}$$

$$\text{If } d_{max} = G,\ H = 60 \times \frac{B-R}{D} + 120 \tag{6}$$

$$\text{If } d_{max} = B,\ H = 60 \times \frac{R-G}{D} + 240 \tag{7}$$

Saturation is calculated by utilizing the equation of (8) and (9).

$$\text{If } d_{max} \neq 0,\ S = \frac{D}{d_{max}} \tag{8}$$

$$\text{If } d_{max} = 0,\ S = 0 \tag{9}$$

Value is calculated using (10).

$$V = d_{min} \tag{10}$$

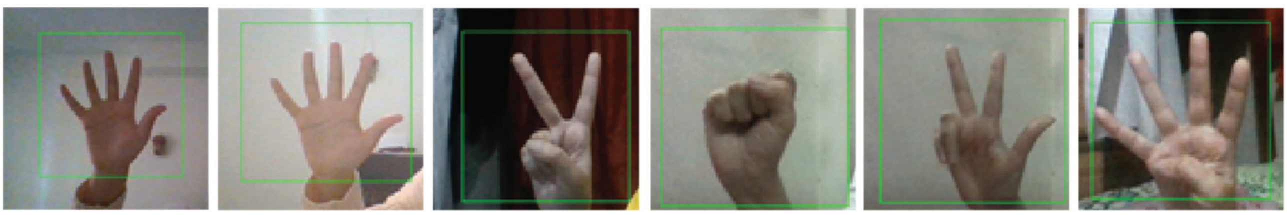


Figure 3. Sample of captured input images form input frames.

After obtaining the above values for each pixel, image can be obtained in the HSV color space as shown in Figure 4(a). The lower-intensity and upper-intensity boundaries of the pixel of the hand are set to get the skin color. The HSV value of the upper boundary is 0, 48, and 80 respectively and the lower boundary is 27, 255, and 255 respectively. If the values calculated are within the range of boundary, then the object is detected as hand. After skin detection, the skin color image is converted to binary image where the pixels of the hand regions are white and background are black pixels as shown in Figure 4(b). There are gaps in the image which are then filled using morphological operations such as dilation and erosion. Once the dark spots are filled, Gaussian blur is used to remove noises in the image. Finally, an appropriate shape of hand is obtained as shown in Figure 4(c).

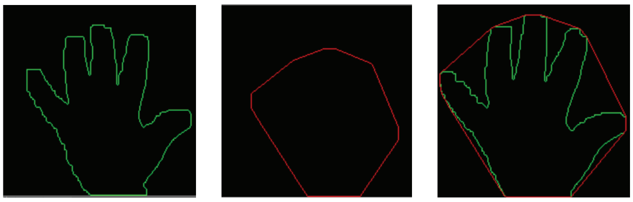
a b c

Figure 4. Example of HSV image, binary image and morphological operation: (a) HSV image (b) binary image (c) binary image after morphological operation.

Detecting Contour and Convex Hull

Contour is a figure represented by collection of all the steady points on the boundary, having the alike color or intensity. The contours measure is an excellent medium to form analysis and object identification. The contour is depicted on the boundary of the object's image which is gained once by thresholding. It is used to obtain the convex hull of the object so that convexity defects can be obtained later [20]. The contour of the hand is generated by the contour detection algorithm in OpenCV.

A convex hull is a collection of successive points in Euclidean space connected to a contour. Convex hulls are drawn around the contour. The convex hull of an object is the lowest boundary that allows the object to be completely bound or wrapped. It works as an envelope around the region of interest [21]. The algorithm for contour and convex hull detection is presented in Algorithm 1. The process of detecting contour and convex hull is depicted at Figure 5. The contour of the hand is shown in Figure 5(a), convex hull of hand image is shown in Figure 5(b) and Figure 5(c) shows the output after contour and convex hull of hand is obtained.



(a) (b) (c)

Figure 5. Experimental process of detecting contour and convex hull: a) contour, b) convex hull, and c) contour and convex hull.

Feature Extraction Based on Area and Convexity Defects

The area between convex hull and contour are convexity defects. The area of convexity defects has three points. These are starting point of contour (A1), ending point of contour (A2) and concave point (A3) as shown in Figure 6. The distance between all these points i.e., A1, A2, and A3 are obtained, i.e., a, b and c. Then using cosine rule, the angle (θ) is calculated. Finally, the distance between convex hull and the concave point is obtained, which is the depth of convexity defect. The portion of convex hull that is not covered by hand is known as area ratio. It is needed to determine the gesture of zero and one. If the angle is less than or equal to 80 and the distance is greater than 40, then there is a convexity defect. Similarly, all the defects are calculated in the hand. Once the number of convexity defect is obtained,

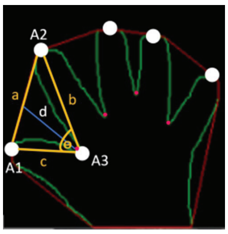


Figure 6. Process of convexity defect detection: image of convexity defect on hand gesture.

Algorithm 1: Contour and Convex Hull Detection

Input	:	Image that has been thresholded once.
Output	:	Image of object with Contour and Convex Hull
Step 1	:	In the thresholding image, all the points of the boundary are connected until a boundary is formed around the object.
Step 2	:	Based on the contour points of the tip of the object (i.e., hand), a convex hull is drawn around the object.
Step 3	:	Stop.

Algorithm 2: Gesture detection based on convexity defect

Input : Number of convexity defects, D and Area ratio of hand, R
Output : Name of the Gesture
Step 1 : If $D = 0$ and $R < 10$ then
Print "zero"
Else if $D = 0$ and $R > 10$ then
Print "one"
Else if $D = 1$ then
Print "two"
Else if $D = 2$ then
Print "three"
Else if $D = 3$ then
Print "four"
Else if $D = 4$ then
Print "five"
endif
Step 2 : Stop

the gesture of hand is classified. Figure 6 shows the detailed image of obtaining convexity defects.

Classification of Hand Gesture Based on Convexity Defect and Area Ratio of Hand

Finally, the hand gesture is classified based on the number of convexity defects and the area of the hand obtained. Firstly, if the contour area of hand is less than 2000, the object in the frame is not recognized as hand. Secondly, if the area ratio of contour and convex hull is less than 10 and there is no convexity defect, the gesture is classified as zero; else it is classified as one. Finally, if the number of convexity defects is 1, 2, 3 and 4, the gesture is classified as two, three, four and five respectively. The gesture detection procedure based on the convexity defects (D) and area ratio (R) of hand is presented in Algorithm 2.

Proposed Sign Language Recognition Using Deep Learning Based YOLOv3

YOLO (You Only Look Once) is a super-fast deep learning object detection implementation. YOLO detects the object quickly by looking at the image only once, replacing the very time-consuming sliding window method used before [22]. YOLOv3, like YOLOv2, uses dimension classes for generating anchor boxes. Since YOLOv3 is a singular network, the loss of objectivity and classification still needs to be calculated separately from the network. YOLOv3 uses a separate logistic classifier for each class in place of the regular Softmax layer to make the classification a multi-label classification. YOLOv3 predicts boxes on three different scales. Thus there will be three gains the ability to predict objects more accurately on different scales. YOLOv3 was introduced with a novel feature extractor, DarkNet-53.

The proposed framework works four different steps that are: (1) setting up the Darknet, (2) preparing the Darknet, (3) training under YOLOv3, (4) Testing under YOLOv3. The workflow of the proposed model is illustrated in Figure 7.

Data Pre-processing

To do the pre-processing task, data was gathered according to the desired environmental conditions. Then experimented with four different conditions and obtained output for six different gestures.

Data Collection

The dataset of sign language from Kaggle [23] that consists of American Sign Language images is collected. A customized image dataset is also collected. The model is trained and tested with both the datasets. Samples of the dataset collected from Kaggle is shown in Figure 8 and customized

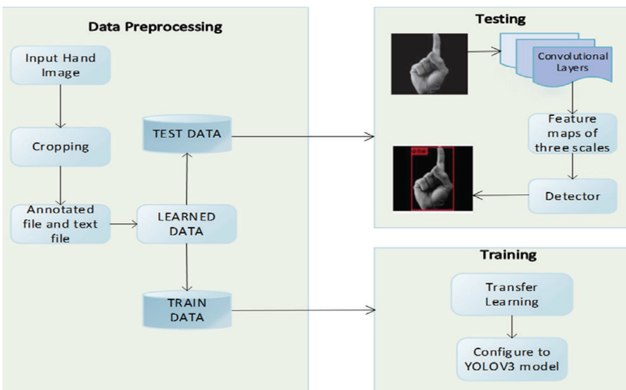


Figure 7. Workflow diagram of the proposed model using YOLOv3.

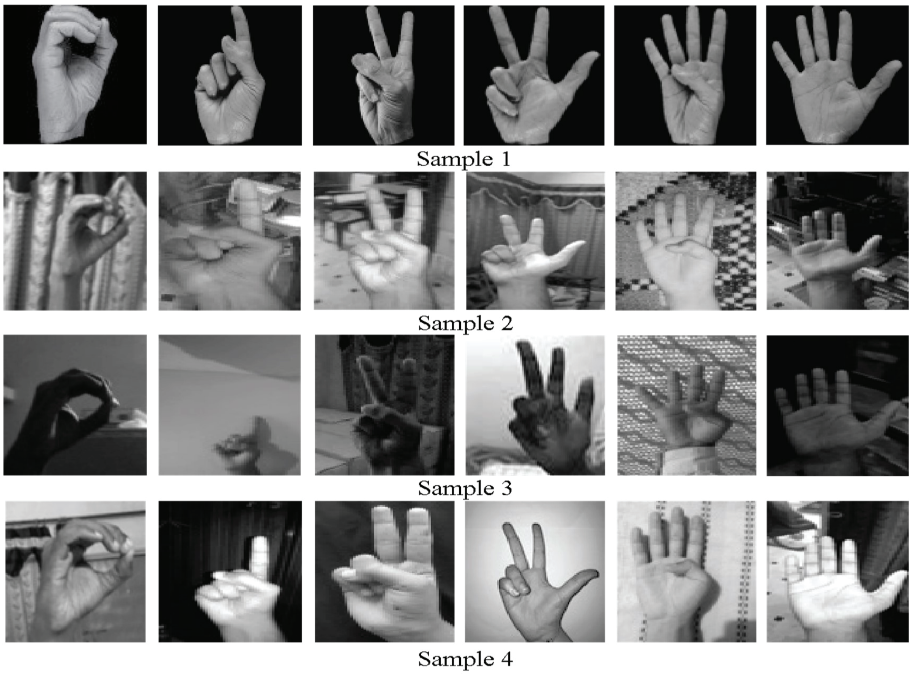


Figure 8. Sample images of all classes in different environmental conditions: sample 1) plain background, sample 2) messy background, sample 3) low light, and sample 4) bright light.

Figure 9. Sample images of customized dataset.

dataset is shown in Figure 9. Moreover, the study is done on four different conditions i.e., plain background, messy background, in low light and bright light for six unique gestures, i.e., 0, 1, 2, 3, 4 and 5. 3000 and 900 images were annotated for training and testing the proposed model respectively. Figure 8 shows input images of 0, 1, 2, 3, 4, 5 in plain background in sample 1, messy background in sample 2, low light in sample 3, and bright light in sample 4 respectively.

Labeling or Annotation

This work focused on the development of static sign language. After collecting the images, the images of each gesture were sav-+9*-4ed in their respective folders. Each folder contained a text file named 'classes.txt', which lists the class names being labeled e.g., 0, 1, 2, 3, 4, and 5. Then, after selecting the desired image, a rectangular box around the image was drawn to show the hand area and labeled as one of the class names 0, 1, 2, 3, 4, or 5. Finally, a corresponding text file of the image is saved in the same directory. The text file contains the class name i.e., object-id and coordinates of the bounding box i.e., center-X, center-Y, width, and height of the image. The flowchart for annotating the images is given in Figure 10.

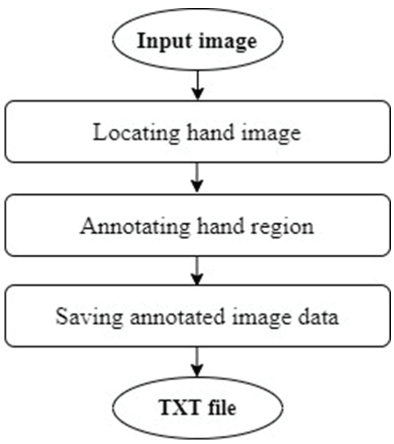


Figure 10. Workflow diagram for annotating the training and testing images.

Configure Training

In this paper, the feature of annotated custom images was extracted using the Darknet-53 framework, and YOLOv3 was used for identification. Table 1 shows the changes made to the configuration file for training the models. Two new files with data and name have been created. The data file contains the path to train.txt, test.txt, name file, and backup folder in Cloud VM via Google Drive. The backup folder is where all the trained weights will be stored after every 1000 iterations. The 'name file' contains all the names of the classes, such as 0, 1, 2, 3, 4, and 5. These two files were copied from Google Drive to Cloud VM.

Feature Extractor: Darknet-53

Darknet-53 is a combination of 53 convolutional layers and 5 maximum pooling layers. Not only is it deeper than YOLOv2 but it also has residual or shortcut connections. Darknet 53 is 1.5 times faster, more reliable than ResNet-101, and is as reliable as ResNet-152 yet two times quicker. Batch normalization and dropout operations are added after each convolutional layer to prevent over-fitting. In DarkNet-53, all convolutional layers use the Leaky ReLU activation function. However, the last layer uses a linear activation function. Adding the residual units to Darknet-53 has enabled YOLOv3 to avoid gradient disappearance by increasing its network depth. As shown in Figure 11, Darknet-53 consists of five residual blocks, using the idea of a residual neural network for reference.

Training Under YOLOv3

Annotated custom images were trained using the Darknet framework. The training process was performed

Table 1. Initialization parameters of the custom YOLOv3 network

Hyperparameters	Value
Momentum	0.9
Batch Size	64
Subdivision	64
Learning rate	0.001
Max Batches	12000
Decay	0.0005
Kernel size	3 × 3 filter size
Activation function	Leaky ReLU for all convolutional layers and linear for final layers
channels	3
Loss function	Binary cross-entropy

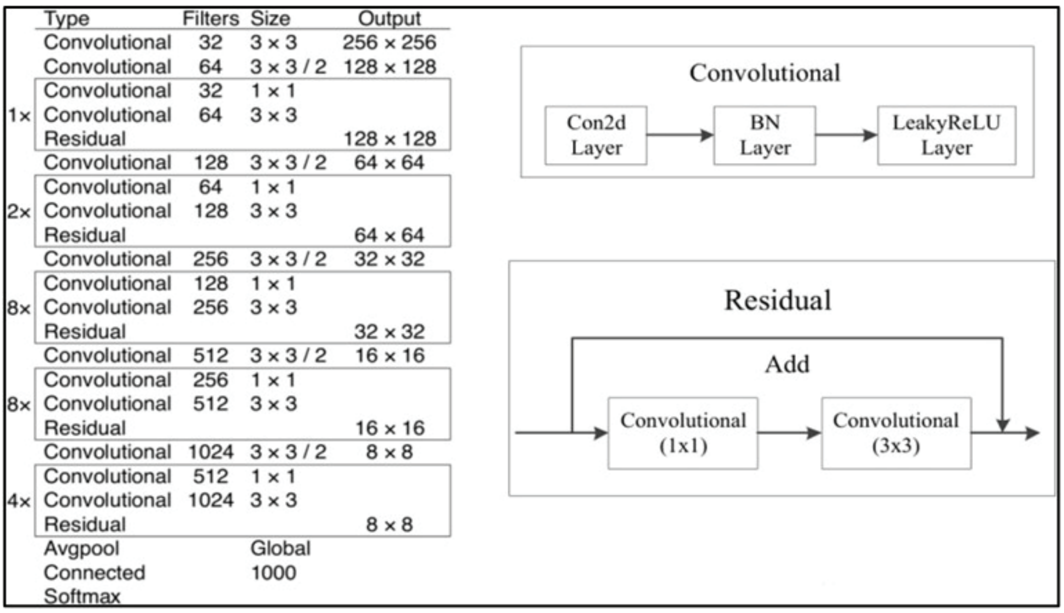


Figure 11. Darknet-53 architecture [24].

using the convoluted weights of a pre-trained model on the COCO dataset to make it easier to train the object detectors on larger datasets. The training process was performed using pre-trained convolutional weights, "darknet53. conv.74" and was compiled into the darknet framework. Training continues until there is an average stable loss. For this study, the average loss at 1400 iterations went below 0.1. The training was stopped when the average loss was 0.0121 at 9200 repetitions as it was the lowest average loss. The loss curve is shown in Figure 21.

Testing Under YOLOv3

After the training process was completed, the model testing process was done to determine the accuracy of the trained weights. In the Google Drive backup folder, the trained weights will be saved in the ".weights" file after every 1000 repetitions. The model has now been tested using 900 annotated images and the most accurate weights have been identified. Next, the model was tested using images from the Kaggle dataset. The performance of the model shown in Table 2 is measured using different performance measure metrics like F1-Score, intersection over union (IoU), and mean average precision (mAP) for each class.

EXPERIMENTAL ANALYSIS

The detailed experimental review of the model being proposed based on hand feature extraction and deep learning process for hand gesture and sign language recognition is presented in this section.

Hand Feature-Based Recognition

The work has been done on python environment as it is a powerful programming language for object-oriented programming. Here, OpenCV library of 4.2.0.34 version and numpy of 1.18.2 version are used. The experiments were conducted on Intel(R) Core(TM) i5-8265U CPU @ 1.60Hz 1.80 GHz pre-processor. It has a RAM of 8 GB. The web camera that was used to capture hand from real time was 0.9 MP. Experiments were performed in four different conditions and have obtained output for six different gestures.

Some processing example of six different hand gesture recognition from four divergent environments based on hand feature is presented in Figure 12. And Figure 13 shows some examples of gesture recognition in different orientation of hand and different combination of fingers. Figure 12(a, b and c) represent the binary hand image, contour and convex hull of hand and gesture label with convexity defect respectively. Sample 1, 2, 3, 4, 5 and 6 represents gesture zero, one, two, three, four and five respectively in Figure 12. The experimental results reveal that the hand feature method determines the convexity defect from different environmental conditions effectively and recognizes the hand gesture efficiently.

Figure 14 demonstrates the experimental examples for six different hand gestures recognition in four different conditions. Sample 1 shows the output for gesture zero, one, two three, four and five in plain background. Experiments in plain background have performed with great accuracy. As there was no other object and the background consisted

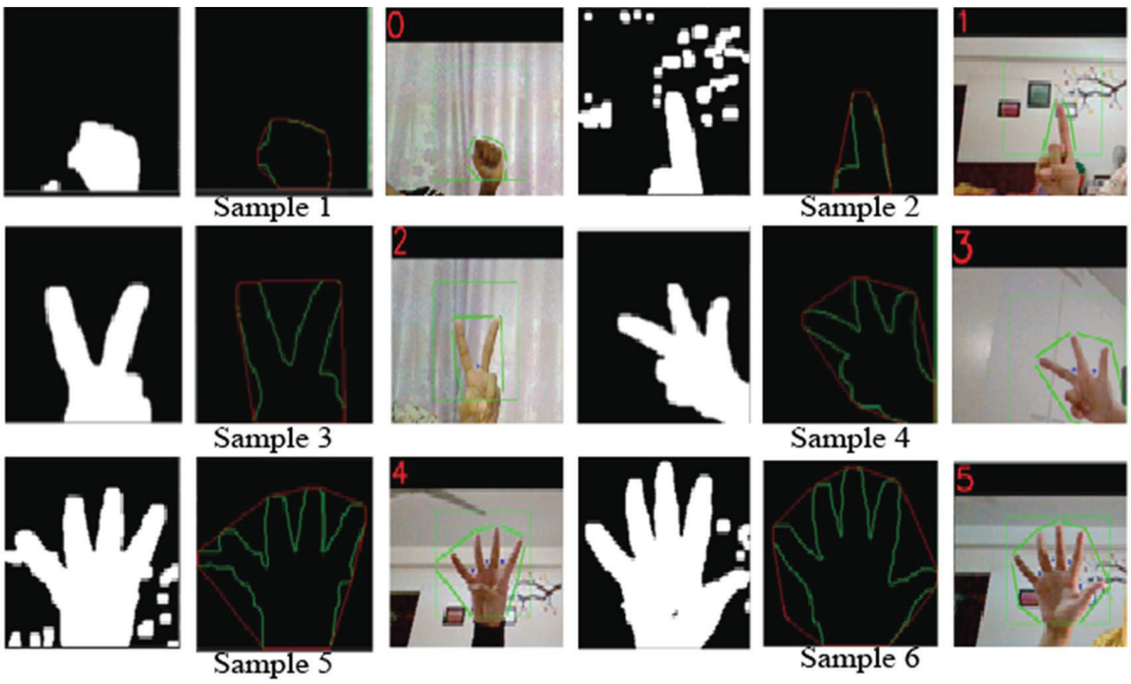


Figure 12. Processing example of hand feature-based gesture recognition: a) binary hand image, b) hand image with contour and convex hull, and c) gesture label with convexity defects.

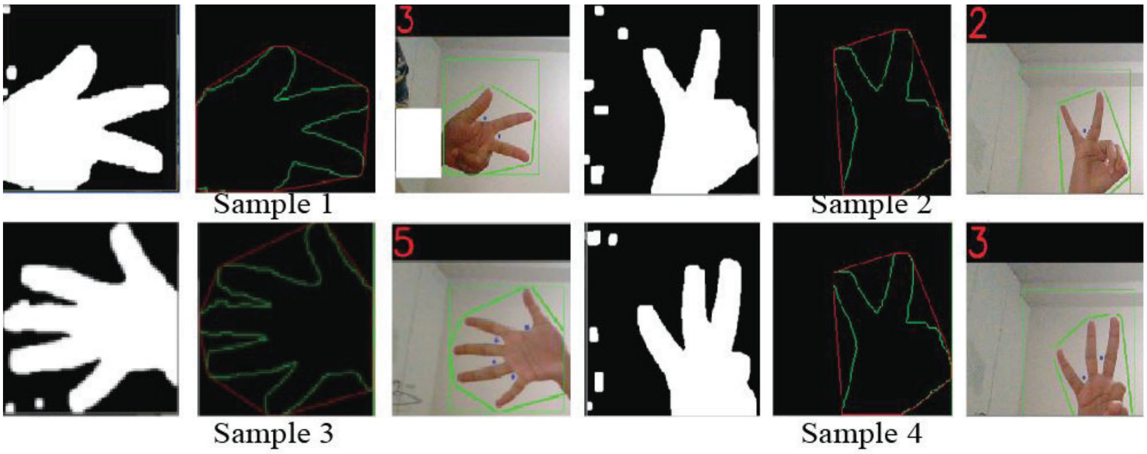


Figure 13. Processing example of hand feature-based gesture recognition in different orientation of hand and different combination of fingers: a) binary hand image, b) hand image with contour and convex hull, and c) gesture label with convexity defects.

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 14. Hand feature convexity defect based experimental results of six gestures in four conditions: a) gesture zero, b) gesture one, c) gesture two, d) gesture three, e) gesture four, and f) gesture five.

of single colour, hand gesture was easy to recognize. Sample 2 demonstrates the experimental examples for gesture zero, one, two, three, four and five that were carried in messy background where there were many objects in the background. It was a highly noisy condition. Hence it was difficult to detect hand. Sample 3 shows the output for gesture zero, one, two three, four and five in poor light condition where the hand is not seen properly. Finally sample 4 shows the output for gesture zero, one, two three, four and five in bright light where the hand was clearly visible. It had greater accuracy as there was very low noise.

In this model of detecting hand gesture, 30 image frames were tested. The targeted gestures were six, i.e., zero to five in ASL (American Sign Language) with the four environmental conditions i.e., plain background, messy background, low light and bright light. Each gesture type was collected from three testers. The average accuracy of the hand feature-based recognition for hand gesture detection is 95.57% as shown in Figure 15.

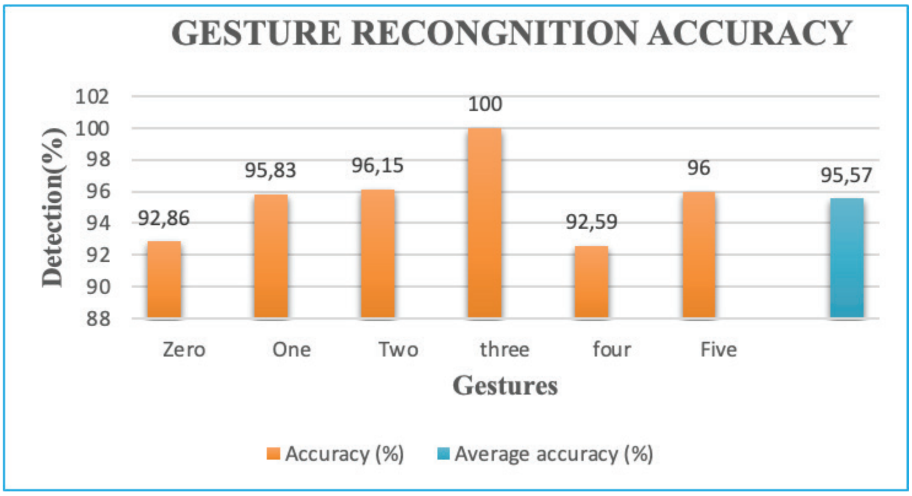


Figure 15. Hand feature-based hand gesture recognition accuracy graph.

Deep Learning-Based Recognition of Sign Language

RCNN, Faster RCNN, YOLO, YOLOv2 and YOLOv3 are all object detection algorithms; however, YOLOv3 has a greater speed in detecting objects among all of them. The model identifies 80 distinct objects in images and videos, yet it is almost as accurate as Single Shot Multi-Box (SSD) [25], however, much faster than the latter. Moreover, YOLOv3 replaces the soft-max function with a self-determining logistic classifier. Instead of utilizing mean square error in calculating the classification loss, YOLOv3 uses binary cross-entropy loss for each specific label [4]. That lessens the complexity of calculation by avoiding the soft-max function completely. Besides, YOLOv3 uses K-means clustering to produce anchor boxes, rather than thoroughly applying anchor boxes at the last detection, YOLOv3 employs a multi-scale anchor mechanism that is used to enhance the detection accuracy for small objects. The recognition of the proposed YOLOv3 model is accomplished by dividing the input image into SxS grid cells [26]. If the center of an object falls into a grid cell, the grid cell is responsible for detecting the object. Each grid cell predicts the position information of bounding boxes [27] and calculates the confidence scores corresponding to these bounding boxes. After a single forward pass CNN, the YOLO network creates multiple bounding boxes for the same identified object. And finally, to erase the overlapping bounding boxes and keep only the correct one, a non-maximum suppression (NMS) is used. YOLOv3 feature extractor, Darknet-53 is deeper than YOLOv2 also contains residuals or shortcut connections. It is likely to be Feature Pyramid Network (FPNs) [28].

Parameters for Performance Measurement

While performing classification predictions, four categories can result. Those terms are True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). The performance of this model is measured by various performance measurement metrics such as Precision, Recall, F1-Score, IoU, and mAP. Precision lets us know how many of the positive predictions are actually true, estimated as (11). Recall shows us how many true positive cases were able to accurately predict through model and calculated using (12). F1-Score refered to the harmonic mean of precision and recall present in (13). Accuracy indicates the rate of accurately predicted cases to the total predicted cases that can be enumerated using (14). Intersection over Union (IoU) is a metric used to measure the accuracy of a model on a particular dataset. IoU is obtained using the predicted bounding boxes by the help of (15). Mean Average Precision (mAP) is a metric used to determine the performance of a model in object detection over a dataset of N images, is computed by (16).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{12}$$

$$\text{F1}-\text{Score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recal} + \text{Precision}} \tag{13}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{14}$$

$$\text{IoU} = \frac{\text{area of intersection}}{\text{area of union}} \tag{15}$$

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{16}$$

RESULTS AND DISCUSSION

Experimental examples of six different hand gestures recognition in different backgrounds are shown in Figure 16, Figure 17 and Figure 18. In each Fig, samples 1, 2, 3 and 4 show the output in plain backgrounds, messy backgrounds, low light conditions and bright light conditions respectively. Figure 16(I) illustrates each of the four conditions for gesture zero and Figure 16(II)

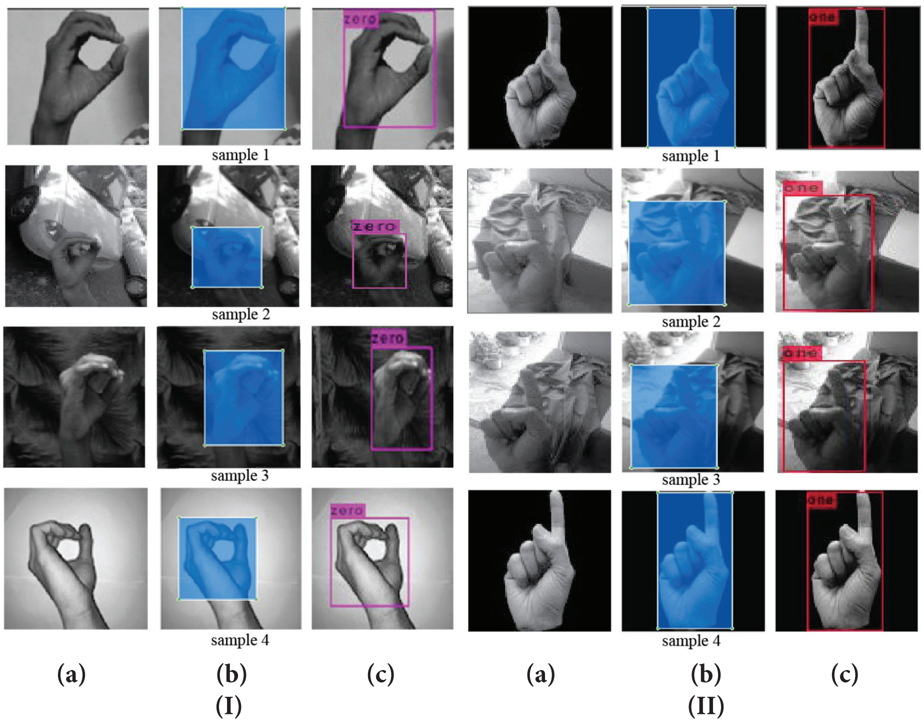


Figure 16. Deep learning based outputs of gesture (Zero in (I) and One in (II)) for four different conditions: sample 1) plain backgrounds, sample 2) messy backgrounds, sample 3) low light conditions, and sample 4) bright light conditions respectively.

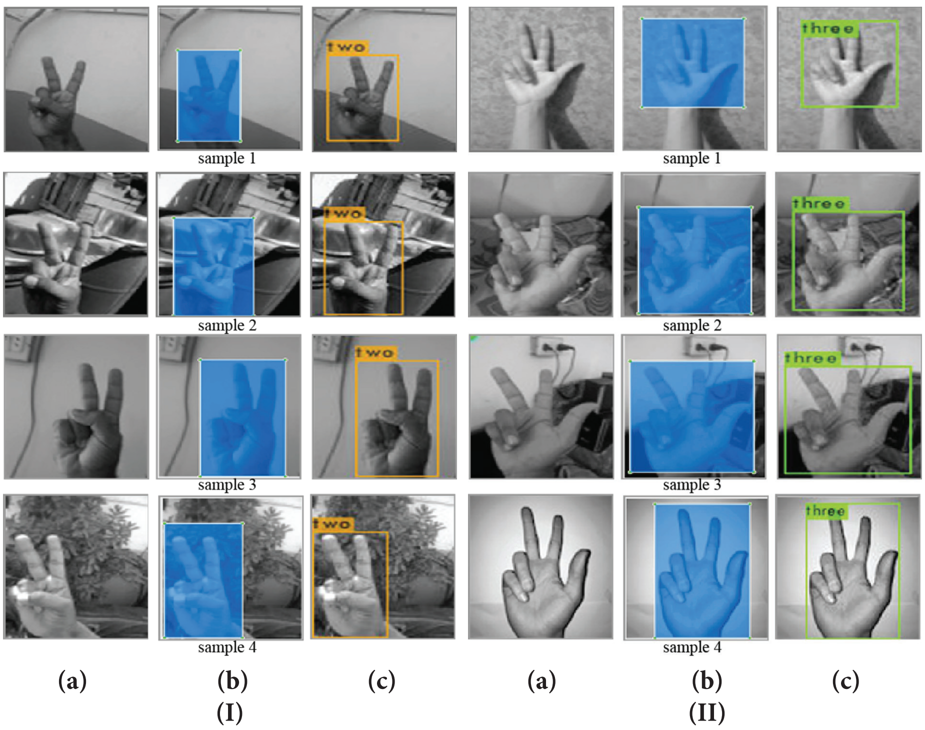


Figure 17. Deep learning based outputs of gesture (Two in (I) and Three in (II)) for four different conditions: sample 1) plain backgrounds, sample 2) messy backgrounds, sample 3) low light conditions, and sample 4) bright light conditions respectively.

illustrates the same for gesture one. Figure 17(I) shows the same for gesture two and Figure 17(II) gives output for gesture three. Finally, Figure 18(I) and Figure 18(II) shows each of the four conditions for gesture four and five respectively.

The proposed deep learning model is also tested with the customized color images. Some processing examples are demonstrated in Figure 19. Sample 1, 2, 3, 4, 5 and 6 shows processing example for gesture zero, one, two, three, four and five respectively.

Performance Statistics

The statistics of the proposed model's performance is illustrated in Table 2 that shows the TP (True Positive), FP (False Positive), FN (False Negative), AP (Average Precision), Average IoU, Precision, recall and mAP of every 1000 iterations.

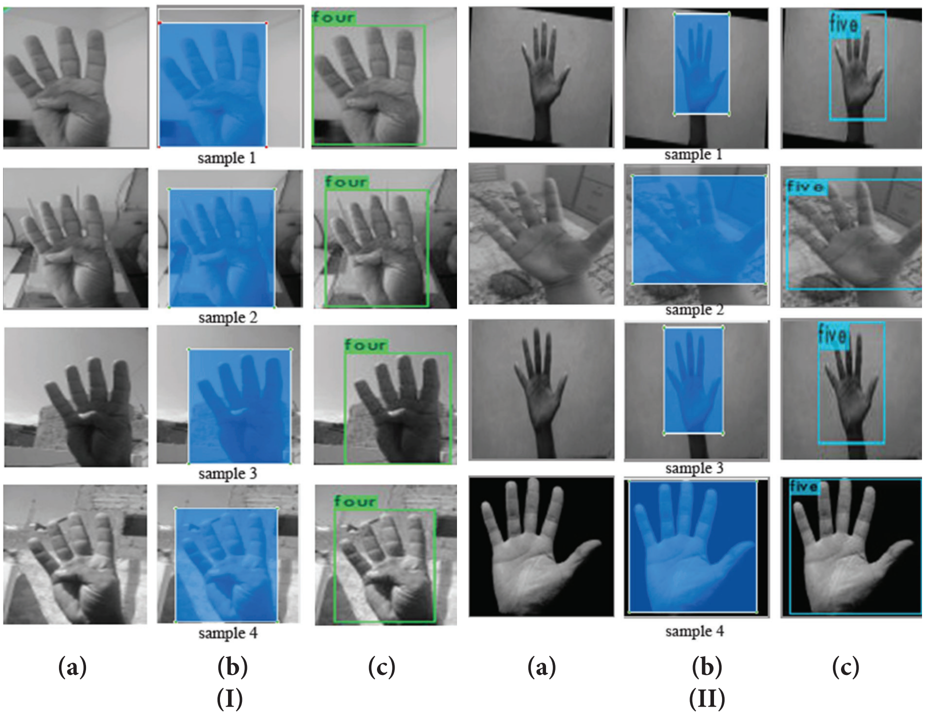


Figure 18. Deep learning based outputs of gesture (Four in (I) and Five in (II)) for four different conditions: sample 1) plain backgrounds, sample 2) messy backgrounds, sample 3) low light conditions, and sample 4) bright light conditions respectively.

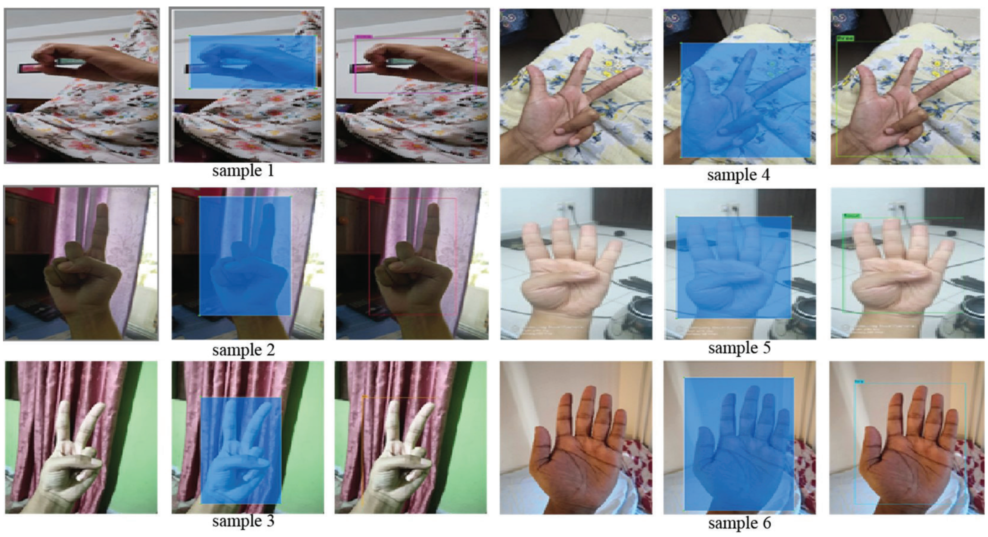


Figure 19. processing example of customized image dataset using deep learning-based model.

Table 2. Results of performance at each 1000 iterations

Iteration	1000			2000			3000			4000			5000		
Classes	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**
0	30	46	21.33	85	3	93.22	79	3	89.25	74	3	89.02	92	8	93.40
1	74	43	63.60	99	10	99.34	99	9	98.73	98	12	99.59	100	7	99.95
2	83	25	82.25	99	6	99.89	98	15	98.36	97	4	98.62	99	3	99.93
3	87	12	91.52	99	4	99.79	96	6	99.32	100	4	99.99	100	2	100.00
4	92	15	96.05	100	11	99.33	99	31	95.21	100	13	98.45	100	6	99.92
5	88	21	88.38	91	5	93.95	91	6	98.39	95	1	99.26	97	3	99.89

Iteration	6000			7000			8000			9000		
Classes	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**	**TP**	**FP**	**AP (%)**
0	83	1	93.47	72	2	85.19	84	2	92.09	92	2	97.14
1	100	3	100.00	100	5	99.92	100	10	99.88	99	9	98.96
2	100	4	99.96	99	5	99.00	100	1	99.97	100	0	100.00
3	100	1	100.00	100	1	99.99	100	1	99.99	99	8	99.90
4	100	4	99.83	100	7	99.52	100	4	99.95	99	0	99.98
5	100	1	99.97	95	6	99.31	100	2	100.00	98	0	99.98

TP = Ture positive FP = False positive AP = Average precition

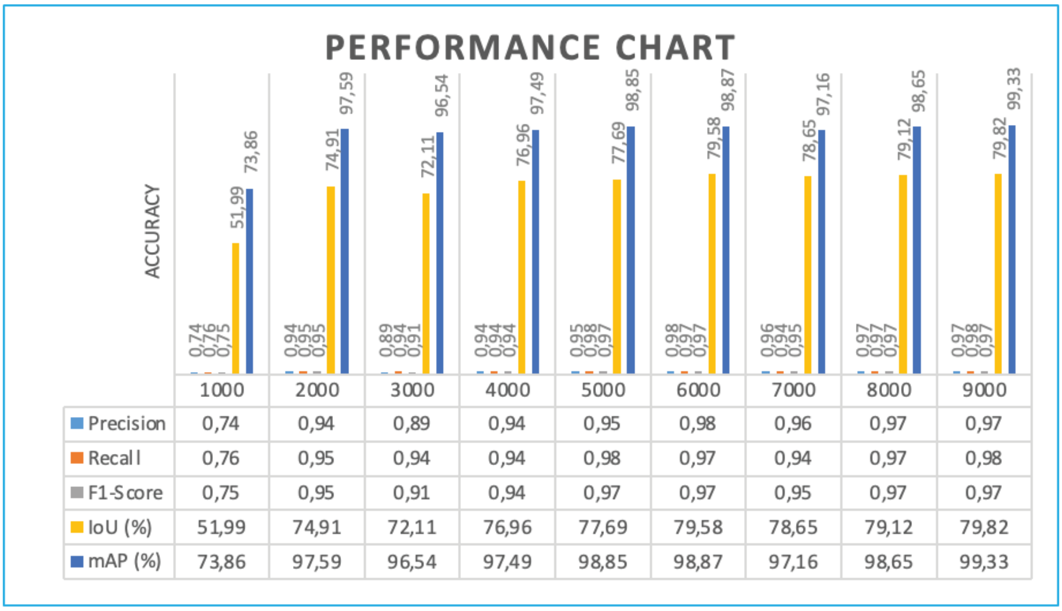


Figure 20. Deep learning based performace evalution chart.

Figure 20 shows that weights at 9000 iterations have the maximum IoU 79.82% and mAP 99.33%. It has the highest TP and the lowest FP and FN. Moreover, it shows high precision, recall, and F1 score. Hence, the weights of 9000 iterations were selected and tested with 600 test images. The loss curve in Figure 21 shows the performance of the model, where the 'Loss' is shown on the Y-axis against the 'Iteration number' from 0 to 9600 in increments of 1200 on the X-axis. After 1200 iterations, the Loss value has become almost zero, and after 2400 iterations, the value becomes saturated. The model was trained for 9200 iterations. The current average loss of the model is 0.0121 at 9200 iterations.

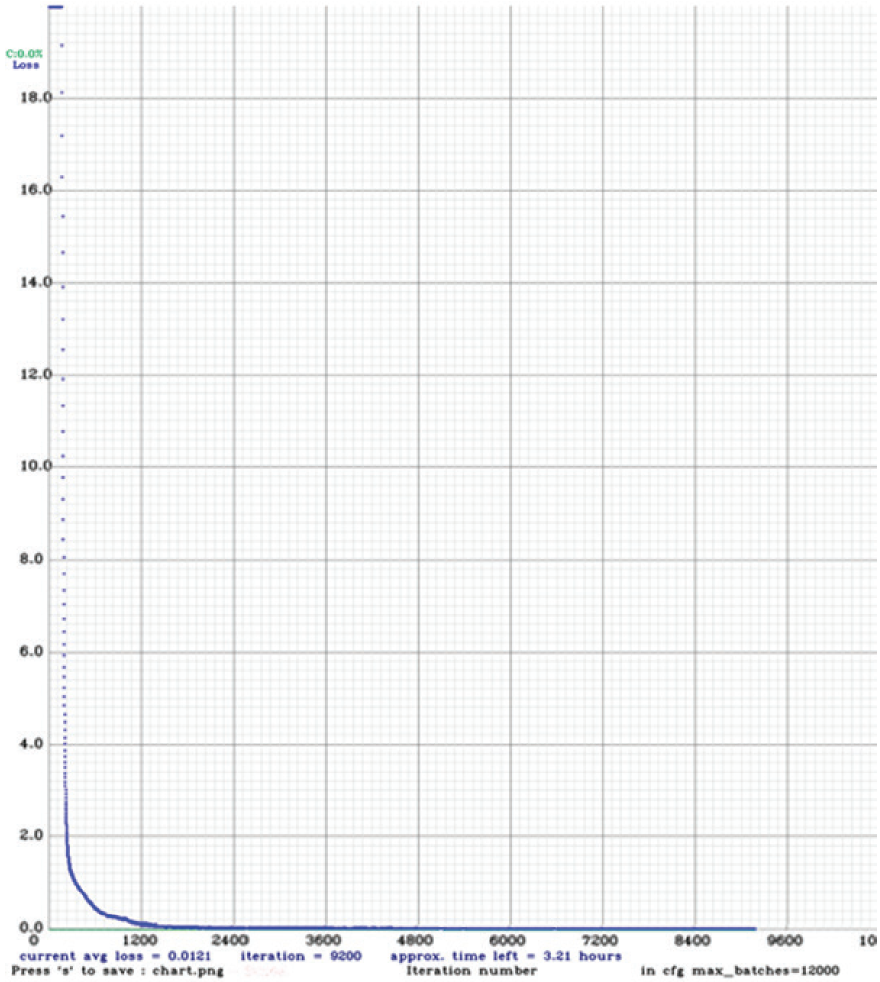


Figure 21. Loss curve.

This paper has proposed two different models of hand gesture and sign language recognition. The first model recognizes gestures using hand feature-based methods that are convexity defects and the second one recognizes using deep learning-based model that is YOLOv3. First proposed model gives great accuracy in the case of simple backgrounds and bright lighting conditions. Table 3 tests the effectiveness of the first proposed model with the existing models.

The proposed model based on image processing shows greater accuracy than all other existing models. Less complexity of this model has achieved greater accuracy. Moreover, the feature extraction method of this paper is very structured and easy. Features of the hand are efficiently extracted with higher detection accuracy. But other methods are very complex and detection accuracy is less. However, it is also able to detect gestures with less computation time in different light conditions and backgrounds.

For YOLOv3, this paper shows that the proposed YOLOv3 automatically identifies various features of hand gesture recognition to support fast, accurate and reliable

Table 3. Hand feature-based comparison between models

Method	Detection Accuracy (%)	Computation time (s)
Proposed model	95.57	1.25
Finger-Earth Mover's Distance [8]	93.2	1.95
Using fuzzy C-means clustering algorithm [9]	85.83	3.56
convexity defect [10]	95.2	2.15

Table 4. Deep learning base comparison between models

Method	Detection Accuracy (%)	Backbone Feature Extractor
Proposed YOLOv3 model	98.92	Darknet-53 (53 convolutional layers)
VGG16 [15]	93.09	VGG16 (13 convolutional layers)
YOLOV3 [16]	94.00	Darknet-53 (53 convolutional layers)
YOLOV2 [19]	97.46	Darknet-19 (19 convolutional layers)
CNN/AlexNet [17]	96.01	AlexNet (8 convolutional layers)

identification and hand image recognition. The proposed method performs with great accuracy of 98.92% in all four conditions when compared with the existing models in Table 4.

The proposed YOLOv3 model is being trained on a large dataset. It uses Darknet-53 as its backbone. Darknet-53 has 53 convolutional layers making it more efficient than all other competing backbones. It efficiently recognizes all six gestures in different environmental surroundings with higher accuracy than all other existing models. Moreover, it can accurately classify gestures from an image of both low light and cluttered background. It overcomes limitations of image resolution and can detect area of gestures from a low-resolution image. The model has achieved higher accuracy of 98.92% than all other existing models.

CONCLUSIONS

This paper has proposed two different approaches to hand gesture and sign language recognition. The first method recognizes gestures using hand featured architecture i.e., convexity defects and the second method does it using deep learning-based architecture i.e., YOLOv3. The work is done on six gestures and four different environmental conditions. Hand feature-based method can successfully recognize different gestures in different environmental conditions irrespective of the position of hand and combination of fingers. The aim of this research is to build a simple system that focuses to detect gestures effectively by reducing the complexity. Once the gesture is detected, the information can be used for different applications, such as, to control industrial robots, security and disable people. It shows an accuracy of 95.57%. The second method is based on YOLOv3. The model is trained on large annotated datasets and can effectively detect the gesture area from images of different light and background conditions. It has a very high accuracy of 98.92%. The accuracy is higher compared to the existing models.

One major contribution of this paper is that it proposes hand gesture recognition system in two ways. This research achieves recognition from both real-time and static images with better accuracy in compared to existing state of arts.

Another contribution of the paper is that it can detect gestures in four different backgrounds. The first method can detect gestures in various distance from the webcam in different backgrounds. Its simple yet efficient way of extracting hand features with high accuracy and less detection time makes it better than all other models. And the second method is trained on large and two different types of datasets. It is successfully able to classify gestures from images that has both low light and complex background. It is also able to recognize gestures from low resolution images with higher accuracy. There is certainly room for improvement in both approaches. For future work on the first proposed method using convexity defects, the aim is to upgrade the model so that it can successfully detect hand gestures with greater accuracy in more challenging backgrounds. And for the second method on deep learning, more gestures for detection could be included for future works.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Braffort A, Gherbi R, Gibet S, Richardson J, Teil D, editors. Gesture-Based Communication in Human-Computer Interaction: International Gesture Workshop, GW'99; 1999 Mar 17-19; Gif-sur-Yvette, France. Berlin: Springer; 2003. [CrossRef]

[2] Gupta S, Bagga S, Sharma DK. Hand Gesture Recognition for Human Computer Interaction and Its Applications in Virtual Reality. In: Advanced Computational Intelligence Techniques for Virtual Reality in Healthcare. Cham: Springer; 2020. p. 85−105. [CrossRef]

[3] Jing Y, Bian Y, Hu Z, Wang L, Xie XQS. Deep learning for drug design: an artificial intelligence paradigm for drug discovery in the big data era. AAPS J 2018;20:58. [CrossRef]

[4] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. 2018.

[5] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A comprehensive survey on transfer learning. Proc IEEE 2020;109:43−76. [CrossRef]

[6] Singh S, Ahuja U, Kumar M, Kumar K, Sachdeva M. Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. Multimed Tools Appl 2021;80:19753−19768. [CrossRef]

[7] Shi M, Ouyang P, Yin S, Liu L, Wei S. A fast and power-efficient hardware architecture for non-maximum suppression. IEEE Trans Circuits Syst II Express Briefs 2019;66:1870−1874. [CrossRef]

[8] 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]

[9] Li X. Gesture recognition based on fuzzy C-Means clustering algorithm [thesis]. Knoxville (TN): University of Tennessee; 2003.

[10] Sharma S, Jain S. A static hand gesture and face recognition system for blind people. In: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN); 2019 Mar; Noida, India. IEEE; 2019. p. 534−539. [CrossRef]

[11] Yun L, Lifeng Z, Shujun Z. A hand gesture recognition method based on multi-feature fusion and template matching. Procedia Eng 2012;29:1678−1684. [CrossRef]

[12] Pradhan A, Ghose MK, Pradhan M, Qazi S, Moors T, EL-Arab IME, et al. A hand gesture recognition using feature extraction. Int J Curr Eng Technol 2012;2:323−327.

[13] Prakash RM, Deepa T, Gunasundari T, Kasthuri N. Gesture recognition and finger tip detection for human computer interaction. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS); 2017 Mar; Coimbatore, India. IEEE; 2017. p. 1−4. [CrossRef]

[14] Xu Y, Park DW, Pok G. Hand gesture recognition based on convex defect detection. Int J Appl Eng Res 2017;12:7075−7079.

[15] Hussain S, Saxena R, Han X, Khan JA, Shin H. Hand gesture recognition using deep learning. In: 2017 International SoC Design Conference (ISOCC); 2017 Nov; Seoul, South Korea. IEEE; 2017. p. 48−49. [CrossRef]

[16] Zhang Q, Zhang Y, Liu Z. A dynamic hand gesture recognition algorithm based on CSI and YOLOv3. J Phys Conf Ser 2019;1267:012055. [CrossRef]

[17] Jiang D, Li G, Sun Y, Kong J, Tao B. Gesture recognition based on skeletonization algorithm and CNN with ASL database. Multimed Tools Appl 2019;78:29953−29970. [CrossRef]

[18] Benjdira B, Khursheed T, Koubaa A, Ammar A, Ouni K. Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In: 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS); 2019; Muscat, Oman. IEEE; 2019. p. 1−6. [CrossRef]

[19] Tanmaie U, Rao CS. Hand posture detection and classification using you only look once (YOLO v2) object detector. JAC 2020;13:101-106.

[20] Ganapathyraju S. Hand gesture recognition using convexity hull defects to control an industrial robot. In: 2013 3rd International Conference on Instrumentation Control and Automation (ICA); 2013; Bali, Indonesia. IEEE; 2013. p. 63−67. [CrossRef]

[21] Sayed U, Mofaddel MA, Bakheet S, El-Zohry Z. Human hand gesture recognition. Inf Sci Lett 2018;7:41-44. [CrossRef]

[22] Yang F, Chen H, Li J, Li F, Wang L, Yan X. Single shot multibox detector with kalman filter for online pedestrian detection in video. IEEE Access 2019;7:15478−15488. [CrossRef]

[23] Sign Language for Numbers. Kaggle. 2012. Available from: https://kaggle.com/muhammadkhalid/sign-language-for-numbers Accessed on Feb 06, 2024.

[24] Liu J, Wang X. Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. Front Plant Sci 2020;11:898. [CrossRef]

[25] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. SSD: Single shot multibox detector. In: European Conference on Computer Vision; 2016; Amsterdam, Netherlands. Cham: Springer; 2016. p. 21−37. [CrossRef]

[26] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; Las Vegas, NV, USA. 2016. p. 779−788. [CrossRef]

[27] Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017; Honolulu, HI, USA. 2017. p. 7263−7271. [CrossRef]

[28] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017; Honolulu, HI, USA. 2017. p. 2117−2125. [CrossRef]