Hand Gesture Recognition Based on Various Deep Learning YOLO Models

Soukaina Chraa Mesbahi, Mohamed Adnane Mahraz, Jamal Riffi, Hamid Tairi
Laboratory of Computer Science, Signals, Automation and Cognitivism (LISAC)-Department of Computer Science-Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco

***Abstract*—Some varieties of sign languages are used by deaf or hard-of-hearing people worldwide to interact with others more effectively, consequently sign language's automatic translation is expressive and important. Significant improvements in computer vision have been made recently, notably in tasks based on object detection using deep learning. By locating things in visual photos or videos, the genuine cutting-edge one-step object detection approach greatly provides exceptional detection accuracy. With the help of messaging or video calling, this study suggests a technique to get beyond these obstacles and enhance communication for such persons, regardless of their disability. To recognize motions and classes, we provide an enhanced model based on Yolo (You Look Only Once) V3, V4, V4-tiny, and V5. The dataset is clustered using the suggested algorithm, requiring only manual annotation of a reduced number of classes and analysis for patterns that aid in target prediction. The suggested method outperforms the current object detection approaches based on the YOLO model, according to experimental results.**

Keywords—Neural network; deep learning; YOLO; object detection; hand gesture

I. INTRODUCTION

In human-computer interaction (HCI), the hand represents a necessary role as a medium of interaction [1]. Verbal and non-verbal communication can be roughly categorized as forms of communication. A gesture is a type of nonverbal communication in which the movement of the hand, face, or other body parts conveys a specific message [2]. The hand is the most aspect of body language used to make gestures for creating communication. There are two types of hand gestures: static and dynamic. The postures in which stable shapes of the hands are represented by static hand gestures and the dynamic hand gestures contain a series of image sequences. For many computer vision applications [3], including hand action analysis, human-computer interaction, sign language identification, virtual reality, and driver hand behavior monitoring, hand gesture recognition and detection in cluttered situations is a key task. The main goal [4] of this research is to identify the challenge of detecting and recognizing static hand gestures when the hand adopts positions to convey particular meanings. Due to the numerous hand configurations and angles with respect to the image sensor, this problem exhibits a high level of complexity, making it challenging to recover the hand shape. Otherwise, in many applications, such as driver hand monitoring and hand gesture commands to prevent driver distraction, sign language recognition for deaf and speech-impaired people, and many more applications, identifying static hand motions plays a vital role . The gesture of a hand and the location of its fingertips are necessary information for a computer to comprehend the state of the interaction medium. Recognizing hand gestures is evenly essential to interpret sign language [5].

Hearing-impaired try to find basic necessities similar to normal human beings like learning, writing, teaching, communicating, reading which may not be easy for them. There are several forms of communication in the world in which people communicate with each other. One of these means of communication is sign language. Sign Language is a natural conversation that frequently hearing-impaired people utilize for communication. Occasionally, to let hearing-impaired people communicate easily with normal people, this Sign Language has to be maintained by technology to identify the sign language [6].

The recognition of sign language represents the technology that makes the computer to recognize the sign utilized with the signer and reform it to text with the help of some algorithms.

Speaking and writing are not used to communicate spoken languages; instead, facial expressions, body language, and hand gestures are used. Due to the peculiarity that these languages are expressed by visual rather than aural means, this clearly sets them apart from other languages and creates a unique language barrier because sign languages have quite different ways of expressing and interpreting ideas. Furthermore, spoken languages are more common than sign languages. Research on the translation of sign language is not as advanced as that of spoken language due to the dearth of texts that permit research access to language. For instance, there isn't a ready-to-use digital sign language translation tool that can translate between spoken and sign languages as well as vice versa.

Language can be also the construction of mimic, gesture, the finger-spelling, and hand sign, in addition to the hand position. By using their bodies, particularly their hands, fingers, and arms, hearing-impaired people can interact nonverbally by using sign language. In the field of computer vision, it is crucial to be able to recognize patterns in movies or images. An essential task in the field of computer vision [7] is to identify indications in movies or images. Constantly understanding what signers are seeking to communicate or describe needs recognizing the numerous hand gestures they use. Arabic Sign Language, American Sign Language, Indian Sign Language, Indonesian Sign Language, and others all have diverse sign language structures [8]. One of the main areas of study in computer vision and machine learning is object detection. Lately, object detection becomes the key to solving real-world problems in applications such as face detection, object Tracking, video surveillance, autonomous vehicles, face detection, pedestrian detection, etc. [9]. Object detection presents an important task of detecting a custom object in images or video, etc. These images or videos contain many objects or a few objects at multiple positions. This task is accomplished by providing the list of different objects that are in the image, providing the object's coordinates and information about the object's location in the image. Supplementary information comprises a bounding box that designates the location as well as the probability with which the object was detected. Supplementary information comprises a bounding box that designates the location as well as the probability with which the object was detected.

The practice of classifying data so that a model can make decisions and take action is known as data annotation. For a range of applications [10], including those that rely on machine learning to analyze images and robotic vision, computer vision, facial and hand identification, image annotation is crucial. To train these solutions, metadata must be attributed to the images in the form of captions, identifiers, or keywords. Image annotation increases precision and accuracy by adequately training these systems. Convolutional neural networks with regional learning are currently popular in detecting work. For object localization, RCNN, Fast RCNN, and Faster RCNN were developed [11]. Recently, the idea of You Only Look Once (YOLO) was used for localizing an area of interest. This work on hand gesture identification focuses mostly on classifying and identifying the gestures. As a technique, hand recognition uses a number of algorithms and ideas from other disciplines, like neural networks and image processing, to discover the movement of a hand. There are a number of object detection methods that help to identify the class and gesture that each algorithm is targeting. This study compares various algorithms and determines which one provides faster, more accurate results than the others. You Only Look Once (YOLO) v3, YOLO v4, Yolov4-Tiny Darknet, and YOLO v5 algorithms were used to analyze the structure and mechanism deduction of hand gesture recognition in order to realize this detection.

For several kinds of hand motions, we suggest a new dataset in this work. Our dataset includes everyday activities, people from various backgrounds and nations, as well as various lighting conditions. For 50,000 photos in our dataset, bounding box annotations are present.

The following are this paper's primary points:

- This work on hand gesture recognition primarily aims to categorize and identify the gestures.
- This paper examines several algorithms, You Only Look Once (YOLO) v3, You Only Look Once (YOLO) v4-Tiny darknet, and You Only Look Once (YOLO) v5 to evaluate the structure and mechanism deduction of hand gesture recognition.
- Our main objective is to describe the datasets, evaluation measures, and experimental setup that we employed for our evaluation.

The remainder of this paper is organized as follows. Brief reviews of several related research on hand detection and gesture identification are included in Section II. The proposed system is fully described in Section III along with an explanation of each component's purpose along with the dataset used. In Section IV the evaluation metrics, experimental setup, and comparison of the obtained experimental results are discussed. Section V discusses the results obtained. Finally, Section VI presents the conclusion.

We also provide a bounding box labeled dataset for object detection methods with this dataset, which contains over 40.000 carefully labeled photos.

II. Related Work

Hand gestures can be recognized in a variety of data sources, including video and photographs, wearable sensors, etc. There are several types of research works on hand gesture recognition, The earliest technique for hand gesture recognition makes use of hand gloves with cables, sensors, LED markers, or other devices [12]. These techniques only provide accurate results when illumination conditions are stable, but classifying hands is a highly challenging problem. Many characteristics, including skin tone and velocity, have been suggested for the detection of hand motions [13], articulated models, hand crafted spatio-temporal descriptors, and trajectory based information. Convolutional neural networks' present success is inspired by deep feature based techniques, and researchers have developed a number of object identification and recognition techniques based on CNNs [14]. These techniques have been created and used for hand detection as a result.

Though, results from image recognition can be applied to tasks in various areas of computer vision, such as object detection using the methods YOLO, R-CNN, fast R-CNN, and faster R-CNN, or semantic segmentation using U-Net [15].

By Roy et al. [16] it was recommended to employ a two-stage hand detector on the basis of the region-CNN (R-CNN) and Faster R-CNN frameworks. To increase the robustness of the deep features, Le et al. [17] suggested a novel technique that incorporated local and global context information. By aggregating several scale feature maps, they expanded the region-fully convolutional network (R-FCN) and faster R-CNN. On two difficult datasets, the performance of this method was adequate. Tokenization is a pre-processing method that Orbay et al. [18] suggested improves the success of translations. If supervised data is available, tokens can be learned from sign videos. Annotated data is, however, hard to come by and expensive to annotate at the gloss level. To find semi-supervised tokenization methods without the burden of extra labeling, adversarial, multitasking, and transfer learning were used. To undertake a more thorough examination, it offers numerous experiments that compare all the approaches in various contexts. In order to take use of the parallelism that all sub problems share, Oscar and colleagues [19] suggested a technique that exploits sequence limitations inside each separate stream and combines them by explicitly enforcing synchronization points. Using the hybrid method, embed strong CNN-LSTM models in each HMM stream. This makes it possible to identify traits that don't have enough discriminative power on their own. Utilizing the sequential parallelism to train sign language, mouth shape, and hand shape classifiers, the approach is then applied to the area of sign language recognition.

Convolutional neural networks were suggested by Gruber et al. [20] for the classification of sign language number motions. The collection contains recordings of 18 distinct people that were made with the Kinect v2 device. In this study, just depth datastream was employed. Classic VGG16 architecture was used for a classification challenge, and its outcomes were compared with the chosen baseline approach and other examined architectures. Research and development of the assistive mobile information robot prototype was presented by Ryumin et al. [21].

The single-handed gesture detection system, the technical description of the robotic platform architecture, and the navigation algorithm are all based on a database of elements used in Russian sign language.

A unique approach based on fusing conventional hand-crafted features with a CNN was developed by Chevtchenko et al. [22]. They tested their approach using depth and grayscale photographs, where the background is eliminated using depth information and the hand is taken into consideration to be the nearest object to the camera.

CNNs were used by Liang et al. [23] to extract features from point clouds that a depth sensor had recorded. Since the first object detection methods in computer vision were advised, object annotation in digital images has generally been taken into consideration. Numerous studies have focused on accelerating the annotation of picture datasets for object detection tasks. Multiple techniques for bounding box annotation were advised by Papadopoulos et al. The annotator just requires checking the label intended by the network in their bounding box verification approach [24] with an accept / reject decision by humans.

Learning intelligent dialogs that take into account the benefit of a trained network to build a bounding box on the image was advised by Konyushkova et al. [25]. To validate the bounding box suggested by the detector in each image, a human annotator is required. The first step in fully annotating the initial batch of images from the unlabeled dataset is hand annotation. Drawing bounding boxes and assigning class names to photos is an entirely manual process that requires human intervention.

To address the issues with RCNN, some writers [26] suggest Fast RCNN, where each and every image is fed only once to the CNN, and feature maps are created using a selective search technique. To shorten the time required to detect, Ren et al. developed a Faster R-CNN modification to the Fast RCNN extension [27]. In order to localize the hand position in a background with no clutter, Soe and Naing [28] used the Faster R-CNN technique using the Caffe framework. Using the NUS dataset, Pisharady et al. [29] showed the segmentation strategy to detect the hand posture and achieved 93% accuracy.

Convolutional Neural Network (CNN) technology is the foundation of YOLO [30], which may produce quick and accurate object recognition. The state-of-the-art object detection technique is very quick from beginning to end. YOLO is frequently used to forecast object detection tasks like real-time pedestrian detection, mask detection, and traffic sign recognition. After the hand position has been localized using YOLOv3, the hand gesture is fed to CNN so that the motion can be detected. The YOLO (You Only Look Once) approach predicts the detected object in the input photos after only one viewing by the neural network. It operates by dividing the input image into several grids with predetermined grid sizes, and then calculating the likelihood that each grid contains the target object [31]. In a single algorithm run, it predicts every class and object bounds that are present in the image. The YOLO algorithm is also constantly being improved in terms of accuracy, speed, and lightweight. Then, You Only Look Once (YOLO) is advised by Redmon et al. [32] to localize the area of interest. Rotation estimation was provided by Denget et al. [33] using CNN to localize the hand region. Deep attention networks for hand gesture localisation were developed by Yuan Li et al. [34].

Shinde et al. used YOLO, which can precisely identify and locate the group frames or even single frames of human movements in the video, to complete the recognition and location of human motions [35]. Based on the enhanced YOLO-v4 [36], Yu et al. proposed a face mask recognition and standard wear detection algorithm.

YOLOv5 was employed in some recent experiments to detect various items. Some recent research looked at replacing the manual inspection procedure with the YOLOv5 during the COVID-19 phase to check for the social distancing proposed by Shukla et al. [37] and face mask by Yang et al. [38] from video and still photos. The model developed by Wang et al. [39] for the detection of safety helmets and tree leaves has been applied in a few other researches. Again, in a number of studies, the YOLOv5 surpassed the R-CNN and other YOLO in terms of speed and accuracy. Then, these features are treated by an algorithm that identifies the specific hand gesture, such as Support Vector Machines (SVM) [11], Conditional Random Fields (CRF), Hidden Markov Models (HMM), and Convolutional Neural Networks (CNN).

After deep learning methods were established, CNN became a more widely used technique for replacing previous methods in object recognition and classification tasks. One of the most difficult issues in this area of computer vision is object detection. Localizing various items in a scene and labeling their bounding boxes are the goals of object detection. The most crucial strategy for solving this issue is to use already trained classifiers to give bounding boxes in scene names [40].

III. Materials and Methods

The techniques and resources that were used in this study to achieve the hand gesture recognition that this paper focused on are assigned to this part. Fig. 1 depicts the suggested hand gesture recognition flowchart and the methodology that was used.

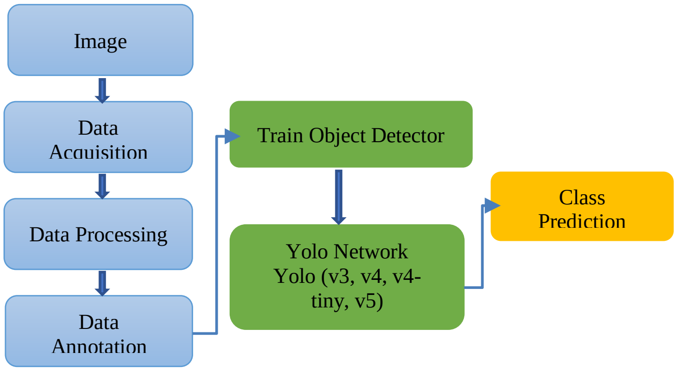

Fig. 1. The flowchart of the proposed hand gesture recognition.

The proposed method was divided into different steps. Firstly, the hand images were collected from the image database for tiny hand gesture recognition and underwent data augmentation to create a hand motions dataset. The captured image is passed through an annotation format to draw a Bounding boxbased hand detector to extract hand regions. Bounding boxes were manually drawn around specific objects in the images to annotate them. The hand region is then extracted once the hand has been recognized and transferred to a better Yolo (You Only Look Once) deep learning model. This model was then optimized and trained on the created datasets. The dataset has seven different gesture classes, such as a Fist, I, Pointer, Palm, Ok, Thumb up and Thumb down. To verify the detection performance, evaluation metrics were produced. The best model was ultimately chosen for the best hand detection across many images.

A. *Dataset Collection*

Data 1: This dataset [41] includes 1400 motions made by 14 distinct individuals, whom each made 10 different gestures and repeated them 10 times. The dataset includes a variety of distinct gestures that were captured using both the Leap Motion and Kinect devices, enabling the development and testing of hybrid gesture recognition systems that utilize both sensors, as suggested in the study, or the comparison of the two sensors.

Data 2: The dataset [42] [43] [44] includes a variety of static motions that were captured with the Creative Senz3D camera. While this camera works well for short range depth collection, its depth range is constrained, and its far range noise level is significant. It has been used to evaluate the performance of a Multi-Class SVM gesture classifier that was trained on fictitious data produced HandPoseGenerator.

B. *Data Acquisition*

In this paper, the hand images were collected from the image database for tiny hand gesture recognition. This dataset [43] has been collected from forty participants; each individual was invited to make seven different gestures. Each instantiation of a gesture consists of around 1400 color frames, and the gestures are carried out in various places throughout the image. The majority of the people have complex backgrounds, with the remaining 50% having basic backgrounds. Backgrounds that are thought to be complicated are extremely crowded, and the lighting varies greatly. The human face and body make up the majority of each image, whereas the hand gestures that need to be categorized only take up 10% of the total number of pixels. We establish a dataset for different person classification and detection. Our dataset contained a total of 3600 images. These images were further divided into seven different classes. Each class comprises an average of 500 images which were labeled and annotated using the bounding box. Our classes started from palm to thumb down, which were finger-pointing different positions. The hand gestures in our dataset are shown in Fig. 2.

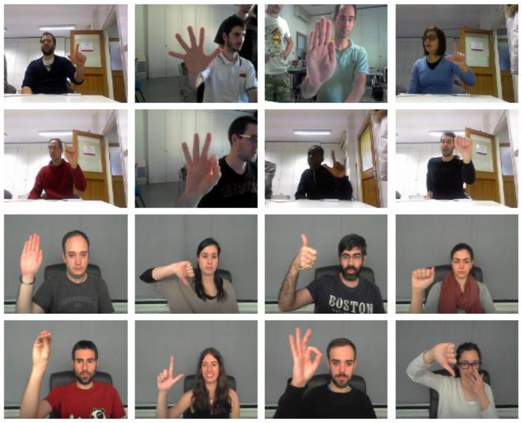

Fig. 2. Sample instances of each class from the dataset with the changes in hand position, shape, and scale.

C. *Data Pre-processing*

By artificially increasing the dataset, data augmentation is a key strategy for creating variations of the training and testing datasets. This step consists to utilize the augmentation techniques such as brightness transformation, randomly altering rotation, motion blur, blurring, and the scale of an input image necessitates that a model contemplates what an image subject looks like in a diversity of positions. Each image was repeated for reading and training, both for the left and right hand, by flipping it horizontally, and sometimes capturing the respective image of those hands to make the set more accurate, using a YOLO setup with a total of images from the dataset.

Additionally, each image for the testing set was captured and labeled. Before moving on to post-processing, it is vital to perform data pretreatment so that we can determine the type of data we have collected and which portions will be relevant for training, testing, and improving accuracy. This part presents the system or methods used to classify, select, and process as well as analyze data and its recognition of characters is discussed. The following methodology is employed to collect data in the form of images, preprocess the data, and then feed the processed data to our model.

1) Manual annotation: The annotation procedure, the training and validation set images were originally 240×240 pixels in size. We utilized the internet tool Roboflow to construct the bounding boxes for each image (www.roboflow.com). This page facilitates making data labels and annotating in the desired format. The images were annotated using the Roboflow Annotate, which is a self-serve annotation tool, and that greatly accelerates the transition from untrained and deployed computer vision models to raw images. After manually drawing and categorizing bounding boxes, this tool makes it possible to change just one annotation or label throughout the whole dataset.

2) Object detection: The object detection model is trained in this section. We concentrate on the most recent deep learning-based object detection models, albeit any detector can be used. In the following part, we'll go into more detail about our training methods. To determine the existence, quantity, and placement of objects in a picture, object detection models are used. Drawing a bounding box around each object of interest in each image was necessary for the image annotation model, which enables us to determine the precise location and quantity of objects in an image. In contrast to image classification, where the class placement within the image is irrelevant because the entire image is designated as one class, the class location is a parameter in addition to the class. Bounding boxes and polygons are examples of labels that can be used to annotate objects inside a picture. Find the existence of things in an image using a bounding box and the types or classes of the objects you find.

a) Input: An image that includes one or more items, like a photo.

b) Output: A class label for each bounding box as well as one or more bounding boxes (each defined by a point, width, and height).

3) Image data labeling with bounding box: We have also produced a dataset with bounding box labeling so that we may utilize the characteristics of the deep learning detection technique. In order to reduce the difficulty and expense of labeling, we randomly choose a few images from each class in the dataset and choose to label the bounding boxes. The most popular annotation shape in computer vision is the bounding box.

Angular boxes called bounding boxes are used to specify where an object is located inside an image. Both two-dimensional (2D) and three-dimensional (3D) models are possible (3D). Polygons or rectangular shapes were manually drawn to annotate the object's edges and to mark each of the object's vertices. The x_center, y_center, width, and height of an object's boundary show its exact location in that image. As shown in Fig. 3, the rectangular shapes are used to label different hands.

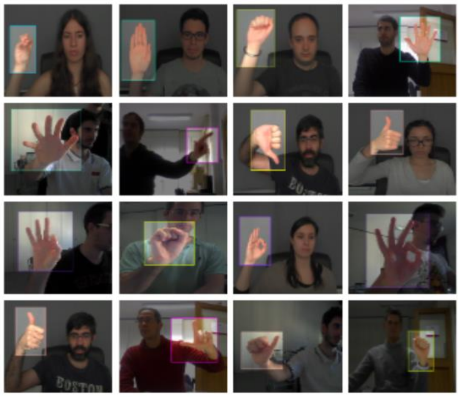

Fig. 3. Labeling different hand classes with bounding boxes.

D. Labeled Dataset

In a labeled dataset, each element of the unlabeled data is given a meaningful "label", "tag" or "class" that makes it more desired or instructive to identify it. Bounding box inference in the training detection model continues until all unlabeled images have been manually fully tagged. In our model we annotated the dataset, we introduce seven different gesture classes, such as Fist, I, Pointer, Palm, Ok, Thumb up and Thumb down.

E. The Structure of the Proposed YOLO Algorithms

1) You Only Look Once (YOLO): YOLO means You Only Look Once is a method that detects all objects in a frame or image in a single shot. Mainly, YOLO makes use of only convolutional layers, to determine which items are represented in the image, a single fully convolutional network (FCN) is used. The YOLO method divides the image into cells or grids; each cell is responsible for object localisation, estimating the number of bounding boxes, and calculating class probabilities. The dataset is collected from various people with various complex backgrounds at different positions, such as variable illumination, gesture variations, and low resolution. Labeling images is essential for good computer vision models. All the images are annotated and labeled manually with Roboflow Annotate which represents a self-serve annotation tool. In this study, we provide a dataset called "BdHand," to which we add bounding boxes to roughly 5000 images in order to make use of the potential of object detection techniques. After the first step of preprocessing and the manual annotation, the second one is training the deep learning models using modern YOLO algorithms (YOLO v3, YOLO v4, YOLO v4 Tiny, and YOLO v5). To understand the different algorithms which we are proposing, the diagram presented in Fig. 1 shows the detection of objects. At first, the first step in the training process is to gather the data, and the second is to label it. We label our dataset using YOLO annotation, which gives us certain values that are later detailed in the model process. We feed the dataset to the DarkNet-53 (YOLO v3) model afterwards, after it has been annotated with YOLO annotation.

2) YOLO v3 Model: YOLO v3 represents an improvement of the essential idea of YOLO, It enables partitioning an image into cells that are in charge of object prediction. Feature extraction networks and the use of detection at multiple scales are changes from YOLO, and the bounding boxes. YOLO v3 [45] presents a deeper architecture of a feature extractor named Darknet-53. It has 53 convolutional layers with a batch normalization layer and leaky Relu activation layer after each one. The feature maps are downsampled using a convolutional layer with stride 2 and without using any kind of pooling [46]. This aids in avoiding the loss of low-level characteristics that pooling is sometimes bed for. As illustrated in Fig. 4, our technique separates the input image using grids into an S×S region first. These cells are used to carry out operations on class probability and bounding box estimates. If the detection of an object in a grid cell is carried out by the object's center. There are now a variable amount of images in our dataset of collected images. The classes we have stored for our YOLO technique are used to label these photographs, and once that is done, we have successfully determined the class and the coordinate for our image set. Additionally, we describe the method by which YOLO manages the network-aimed output, which is achieved by using a formula that requires various coordinates. The b_x , b_y , b_w , b_h are the variables that we employ for the bounding box dimensions and are associated with (x,y) coordinates that represent the center of the box, as well as the width and height, which are represented by p_w, p_h. The estimated four coordinates aret_x, t_y,t_w, t_h, a bounding box for each. The numbers c_x and c_y correspond to the grid cell's upper left coordinates. These variables, which reflect the box prediction components as defined by Equations, are predicted in relation to the entire image (Eq. (1) to (5)).

$$b_x = \sigma(t_x)+C_x \quad (1)$$

$$b_w = \sigma(t_y) + C_y \quad (2)$$

$$b_y = p_w e^{tw} \quad (3)$$

$$b_h = p_h e^{th} \quad (4)$$

$$\sigma(x)=1/(1+e^{-x}) \quad (5)$$

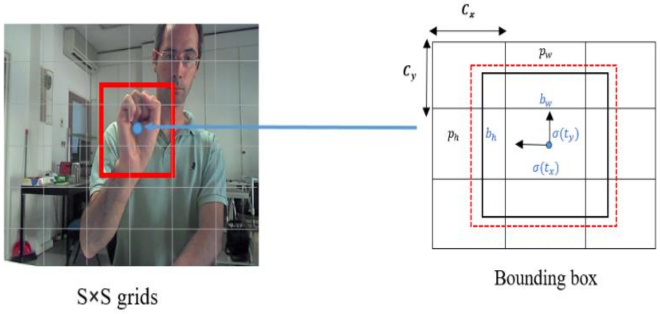

Fig. 4. The bounding boxes with dimension priors and location prediction.

Fig. 4 demonstrates how each value of the algorithm's bounding box gives us the x and y coordinates for the center.

3) YOLO v4 model: The one-stage object identification technique known as YOLOv4 represents the YOLOv3 model's evolution and significant advancement. The number of FPS (Frames per Second) increased by 12% and the mAP (mean Average Precision) by 10% as a result of the introduction of a new architecture in the Backbone and changes in the Neck [47]. The architecture of YOLOv4 is made up of the YOLOv3 head, PANet path aggregation neck, spatial pyramid pooling extra module, and Darknet53 as the backbone.

4) YOLOv4-Tiny model: The YOLOv4-tiny model is based on the YOLOv4 approach and is aimed to increase object detection speed. The prediction process is the same as YOLOv4 and it has a faster target detection speed. YOLOv4-tiny [48] [49] is proposed to reduce parameters and make the network structure simpler and it significantly improves the viability of implementing object detection methods on embedded systems or mobile devices. The Yolov4-tiny method utilized Darknet53-tiny network as backbone network to instead of the CSPDarknet53 network that is used in Yolov4 method. The ResBlock module in the residual network is substituted by the Block module in the CSPDarknet53-tiny network. Fig. 5 depicts the YOLOv4-tiny network structure.

5) YOLO v5 model: The Backbone, Neck, and Head architectural components of the YOLOv5 network are shown in Fig. 6. YOLOv5 Backbone: In order to extract features from images, including cross-stage partial networks, YOLOv5 uses CSPDarknet as its backbone. YOLOv5 Neck: It makes use of PANet to create a feature pyramid network that is then passed to the Head for prediction after the features have been aggregated. YOLOv5 Head: Its layers produce predictions for object detection from the anchor boxes [50].

YOLOv5 is quick and lightweight, and it uses less computing power than other current state-of-the-art architecture models while maintaining accuracy levels that are comparable to those of current state-of-the-art detection models. Compared to the other YOLO versions, it is substantially faster. CSPNET serves as the foundation for YOLOv5's feature map extraction from the image. In order to improve information flow, it also makes use of the Path Aggregation Network (PANet) [51]. For the following reasons, we are utilizing YOLOv5 as it includes helpful elements like a cutting-edge activation function, a convenient manual, a hyperparameter, and a data augmentation technique. It can be trained computationally quickly with minimal resources, thanks to its lightweight architecture. The size model can be utilized with mobile devices because it is relatively tiny and light. Yolov5 differs from the Yolo series in several lighting areas: (1) Multiscale: utilize FPN to improve the feature extraction network rather than PAN [46], which will make the model easier to use and more quickly.

Yolov5 differs from the Yolo series in several lighting areas: (1) Multiscale: utilize FPN to improve the feature extraction network rather than PAN [46], which will make the model easier to use and more quickly. (2) Target overlap: identify nearby positions using the rounding method such that the target is mapped to several central grid points all around it. Yolov5 is a continuation of the YOLO series' most recent iterations [52]. It is more manageable and, in general, more cozy to utilize throughout training. Its architecture may be modified with equal ease, and it can be exported to numerous deployment environments [53].

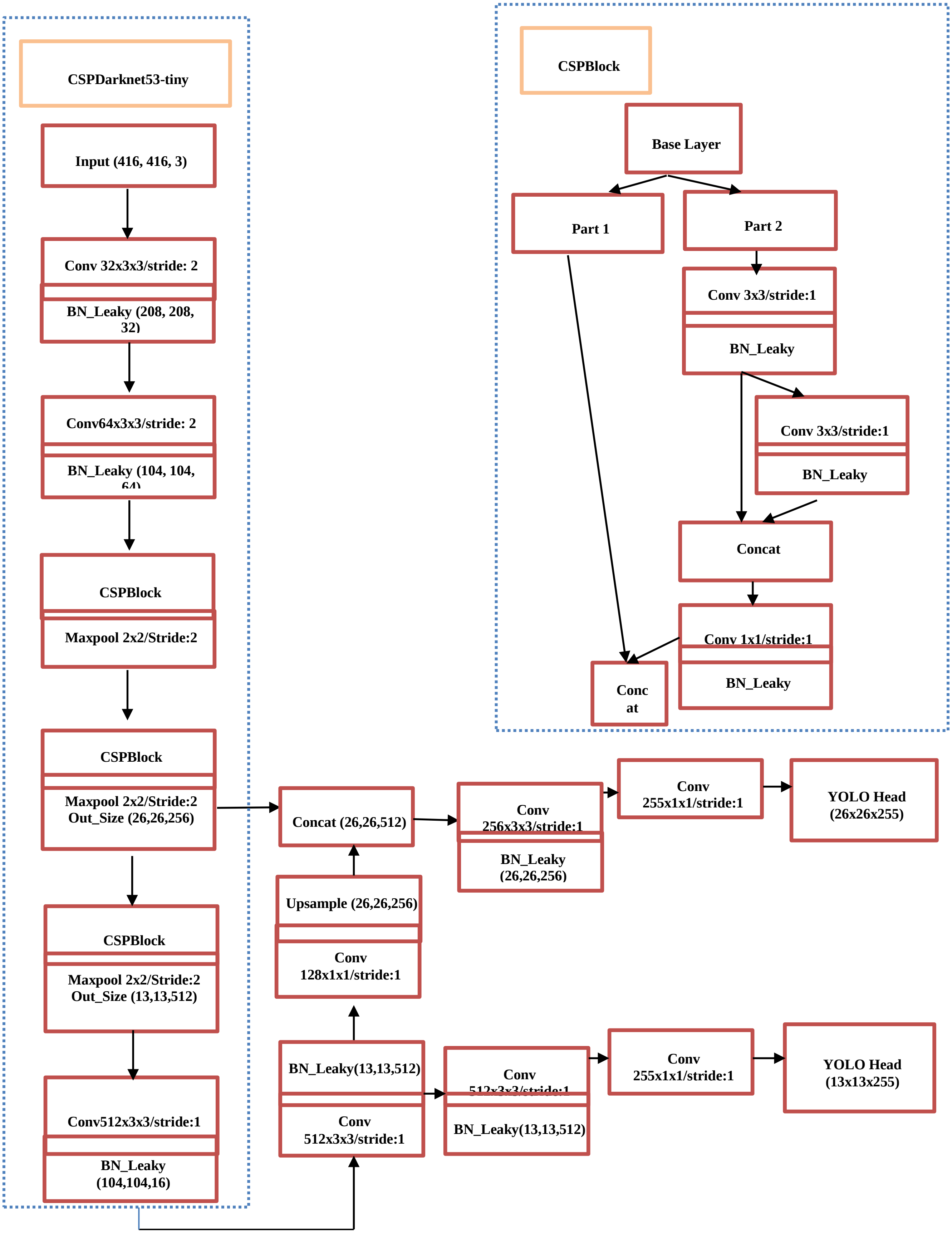

Fig. 5. The network structure of YOLOv4-tiny.

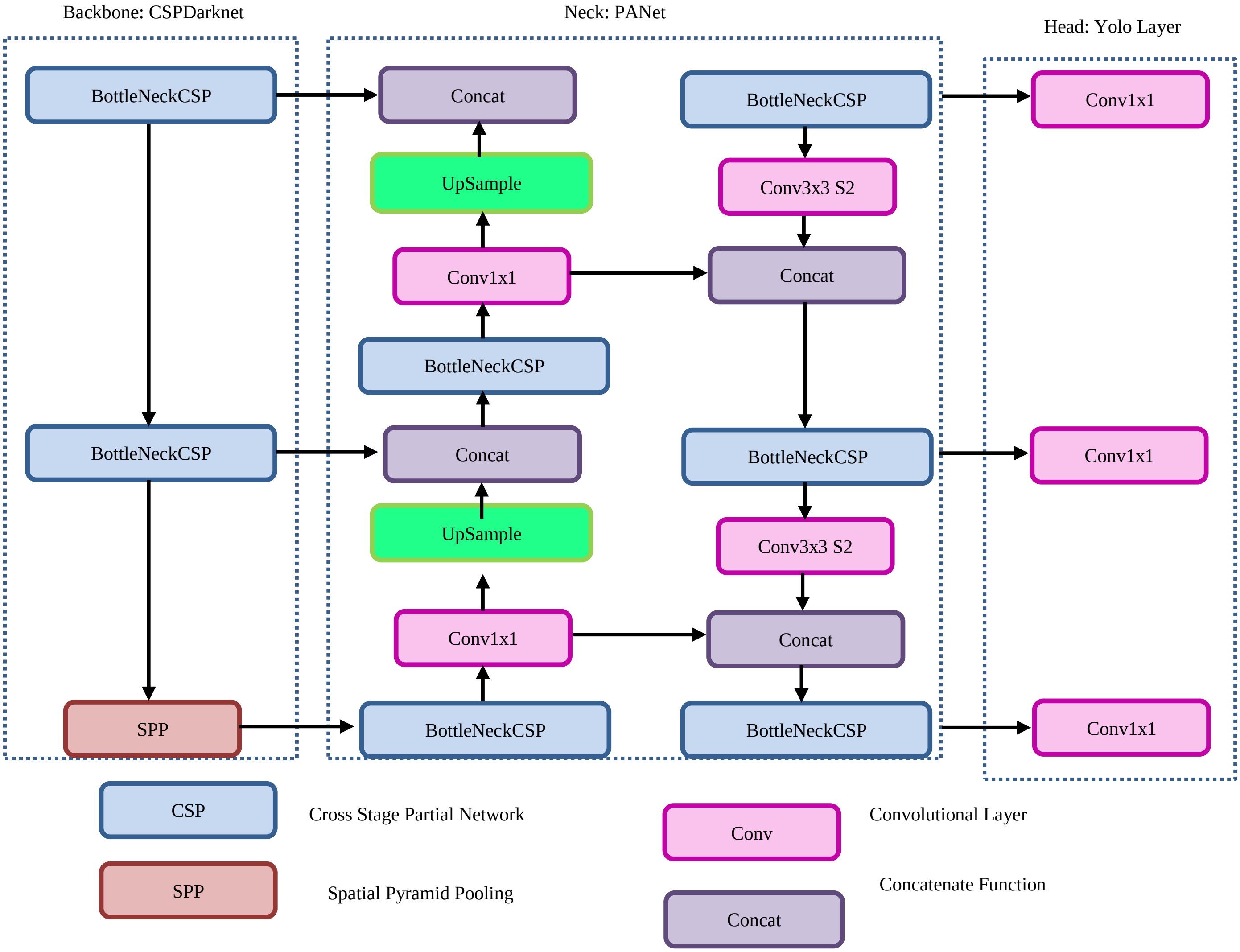

Fig. 6. The general architecture of the YOLOv5 network.

There are many algorithm parameters in the YOLO models, and understanding the influence of these parameters is essential for optimizing the performance of the model for a specific task. Here are some of the most important parameters in YOLO models and their influence:

- Input size: The input size of the YOLO model refers to the resolution of the input image. Larger input sizes cannot only improve the accuracy of the model but also increase the computational cost.
- Anchor boxes: Anchor boxes are predefined boxes of various shapes and sizes that are used to predict object locations and sizes. The number and aspect ratio of anchor boxes can significantly affect the accuracy of the model.
- Batch size: The batch size refers to the number of images processed in a single iteration during training. Larger batch sizes can speed up the training process, but they also require more memory.
- Confidence threshold: The confidence threshold is used to filter out low-confidence predictions. Increasing the confidence threshold can reduce the number of false positives but may also increase the number of false negatives.
- increasing the confidence threshold can reduce the number of false positives but may also increase the number of false negatives.
- NMS threshold: Non-Maximum Suppression (NMS) is used to remove overlapping bounding boxes. The NMS threshold controls the amount of overlap allowed between boxes. A higher threshold can remove more overlapping boxes but may also remove some true positives.
- Backbone architecture: The backbone architecture refers to the architecture used to extract features from the input image. Different architectures have different complexities and can affect the accuracy and speed of the model.

- Training parameters: Training parameters such as learning rate, weight decay, and optimizer can significantly affect the training process and the performance of the model.

The parameters in YOLO models can significantly affect the accuracy, speed, and memory usage of the model. Choosing the right parameters for a specific task requires experimentation and fine-tuning to optimize the performance of the model.

IV. Experiments and Results

A. Evaluation Metrics

In this section, we discuss the four experiments performed using Yolo algorithms YOLO version (v3, v4, v4 tiny, and v5). We implemented and tested the four models during our experiments to train them for our different publicly available datasets. The configuration of these models differs from one to the other. Extensive testing was carried out during our research to confirm the dependability of the suggested YOLO model for hand gesture recognition. The experimental settings are presented in the first stage. The evaluation metrics are then described. The comparative experimental findings are then thoroughly examined and analyzed. To gauge the effectiveness of the proposed hand gesture recognition model in terms of recognition, detection, and computational performance, a number of indicators or metrics were used. The average precision (AP), which is shown as the area under the precision and recall curve at various detection thresholds, was used in this experiment. Eq. (6) contains a definition of the AP equation.

$$AP = \int_1^0 P_r(R_c)\, dR_c \qquad (6)$$

Precision and recall are represented by Pr and Rc. The following parameters of precision, recall, and F1-score are calculated to estimate the model accuracy and the efficiency. When the predicted bounding boxes match the ground truth boxes, the accuracy of the prediction is measured. In addition to these measures, we used Eq. (7), (8), and (9) to derive precision, recall, F1-score, and accuracy using the True positive (TP), False positive (FP), and False negative (FN) metrics. The Precision (Pr), presented in Eq. (7), shows the ratio of true positives (TP) to all expected positives (TP+FP). As a result, it is a crucial measure for deducting the cost of the FP number.

$$P_r = \frac{Tp}{Tp+F_p} \qquad (7)$$

If the predicted bounding box falls beyond the ground truth of the hand, it is indicated by the letters FP, whereas TP signifies that it does so. The likelihood of correctly detecting the ground truth objects is then calculated using recall. Accordingly, the Recall (Rc) shows the proportion of estimated true positives to all actual positives (TP+FN). It is created by Eq. (8) and is occasionally referred to as sensitivity. Instead of the projected bounding box, FN displays the hand of the frame.

$$R_C = \frac{T_p}{T_p * F_N} \qquad (8)$$

The F_1-score measures the overall accuracy, this as shown in Eq. (9), includes the recall values and a statistical precision measure. The F_1-score should be chosen in particular when a balance between precision and recall is necessary, with an F_1-score of 1 defining the optimal value.

$$R_C = \frac{2 * P_r * R_c}{P_r * R_c} \qquad (9)$$

Mean Average Precision (mAP), a well-liked object identification statistic created by Eq. (10), averages the AP values for all classes. As a result, the performance of the model may be quantified using a single metric.

$$\text{mAP}= \frac{\sum_{q=1}^{Q} \text{AveP(q)}}{Q} \qquad (10)$$

Where Q is the number of queries in the set, q is the query for average precision. The mAP is the mean value of average precision for the detection of all classes and is an indicator generally utilized to estimate how good a model is. The FPS identifies how many images can be correctly identified in a single second. GPU utilization refers to the use of GPU RAM when evaluating various detection strategies.

B. Detection Results of YOLO Model

The output of the various classes of hand gesture recognition is shown in Fig. 7. The bounding box covered the maximum part of the hand. It will cleverly determine which gesture is being represented in the zoom situation when the object is too huge, and it then delivers the class ID with the best match. To determine which algorithm was the most effective for hand gesture detection, we used a variety of different ones.

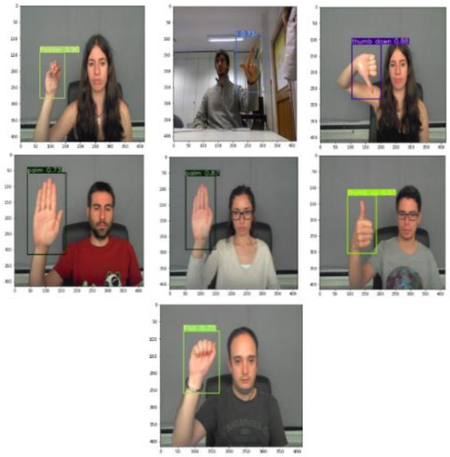

Fig. 7. Detection results of YOLO v3 model.

416 x 416 pixels were chosen as the size of the input images for the training process. The outcomes of the hand detection test utilizing our suggested YOLO v3 model are listed in Table I. We calculated the mean average precision (mAP), then Precision (Pr), average recall (Rc), and F_1-score for each test. Using the suggested deep learning model, we assessed the performance of deep learning models, and we were able to get an accuracy of 98.20% for YOLOv3.

TABLE I. PERFORMANCE ACCURACY FOR YOLOV3

Class	Precision %	Recall%	F1-Score	mAP(mAp@.5)%
Fist	62.40	96.80	73.60	93.70
I	88.40	97.10	91.40	96.40
Pointer	95.70	98.40	96.90	97.60
Ok	96.80	98.60	97.60	98.10
Palm	96.70	98.60	97.60	98.10
Thumb down	97.30	98.60	97.90	98.20
Thumb up	96.40	98.40	97.40	98.00

The suggested model also worked well in various lighting situations, as seen in Fig. 8. The experimental findings show that the proposed model can accurately and efficiently identify different classes of hands in a variety of situations.

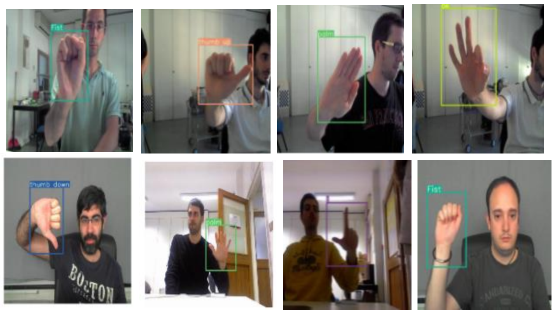

Fig. 8. Detection results of YOLO v4 model.

Table II displays the experimental outcomes for hand gesture recognition using our dataset. This table compared the speed and accuracy of the various classes. Experiments show that 98.40% of the results are accurate.

TABLE II. PERFORMANCE ACCURACY FOR YOLOV4

Class	Precision %	Recall%	F1-Score	mAP(mAp@.5)%
Fist	97.10	98.50	97.79	98.20
I	97.90	98.60	98.40	98.40
Pointer	97.30	98.60	97.94	98.30
Ok	96.90	98.60	97.74	98.40
Palm	96.70	98.60	97.60	98.10
Thumb down	97.30	98.60	97.90	98.20
Thumb up	96.40	98.40	97.40	98.00

Fig. 9 depicts the process for detecting hand gestures. As these results exhibit, our proposed model can treat various shapes of hands, scales, and under various lighting circumstances, as well as comprehend motions in many difficult situations.

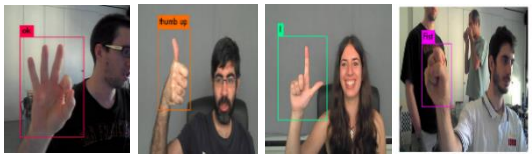

Fig. 9. Detection results of YOLO v4-tiny model.

We conclude that effective hand detection improves the performance of the gesture recognition system with quick processing, which in turn facilitates accurate human-machine interaction, based on the experimental findings provided in Tables II and III.

TABLE III. PERFORMANCE ACCURACY FOR YOLOV4-TINY

Class	Precision %	Recall%	F1-Score
Fist	1.0	92.30	95.00
I	97.0	92.20	94.82
Pointer	1.0	88.50	93.89
Ok	98.870	91.56	95.01
Palm	1.0	92.63	96.17
Thumb down	1.0	88.30	93.80
Thumb up	1.0	90.27	94.88

Fig. 10 introduces the results of test images that contain different people. The results of the experiments show that the suggested model can meet object detection in various complicated backgrounds where the majority of movements were successfully detected.

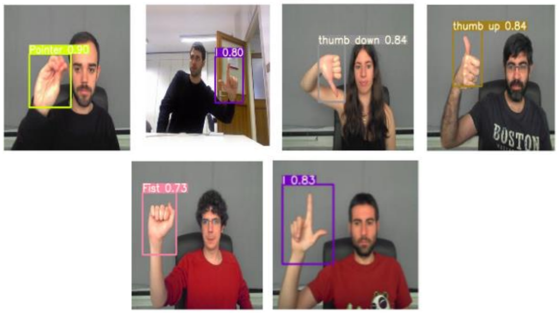

Fig. 10. Detection results of YOLO v5 model.

In this experiment, YOLOv5 performed better overall than YOLOv4, YOLOv4-tiny, and YOLOv3. In comparison to the other models, the YOLOv5 model produced the best results in terms of precision and error. Compared to YOLOv4, YOLOv5 is quicker and more accurate. The results showed that the mAP was much higher when YOLOv5 was compared to YOLOv3, YOLOv4, and YOLOV4-tiny for hand motion recognition. The most effective object detection method at the moment is YOLOv5 (refer Table IV).

TABLE IV. PERFORMANCE ACCURACY FOR YOLOV5

Class	Precision %	Recall%	F1-Score	mAP(mAp@.5)%
Fist	1.00	97.70	98.83	97.80
I	98.80	97.60	98.19	98.10
Pointer	99.9	96.40	98.11	96.70
Ok	98.60	1.0	99.29	99.10
Palm	99.60	1.0	99.79	99.50
Thumb down	1.0	98.10	99.04	98.60
Thumb up	99.80	1.0	99.89	99.50

V. Discussion

YOLO (You Only Look Once) is an object detection algorithm that predicts the bounding boxes and class probabilities of objects in an input image. YOLOv3 and YOLOv4 are earlier versions of the algorithm, while YOLOv5 is a more recent version. Here are some of the differences between these versions:

- Architecture: YOLOv5 uses a different architecture than its predecessors. It has a smaller and more efficient model that makes use of Scaled-YOLOv4 architecture and advanced training techniques such as Mosaic data augmentation.
- Speed: YOLOv5 is faster than its predecessors, particularly YOLOv3, due to its smaller model size and improved architecture. YOLOv5 can process up to 155 frames per second on a Tesla V100 GPU, compared to 82 and 65 frames per second for YOLOv4 and YOLOv3, respectively.
- Accuracy: YOLOv4 is generally more accurate than YOLOv3, with improvements in object detection accuracy and speed. YOLOv5, on the other hand, achieves comparable accuracy to YOLOv4 but with a smaller model size and faster processing speed.
- Training: YOLOv5 uses a different training approach called self-supervised pre-training, which allows it to learn from large amounts of unlabeled data. This leads to better generalization and improved performance on smaller datasets.

The discussion is about the performance evaluation of a proposed deep learning model for hand gesture recognition. The model achieved an accuracy of 98.20% for YOLOv3, which was able to identify different classes of hands in various lighting situations. The Table I provided shows the precision, recall, F1-Score, and mAP scores for various hand gesture classes. The "Fist" gesture had the lowest precision score of 62.40%, while the "Thumb down" and "Thumb up" gestures had the highest precision scores. The recall scores were high for all classes, indicating that the model correctly identified a large proportion of actual positive instances.

The Table II shows the precision, recall, F1-Score, and mAP scores for various hand gesture classes. The precision scores for all classes were high, ranging from 96.4% to 97.9%. The recall scores were also high, ranging from 98.4% to 98.6%, indicating that the model correctly identified a large proportion of actual positive instances. The F1-Scores were all above 97%, indicating a high level of accuracy in detecting and recognizing hand gestures. The mAP scores were also high, ranging from 98.0% to 98.4%, indicating that the model was able to detect the objects with high precision across all the classes.

The Table III displays the precision, recall, and F1-Score for various hand gesture classes. The precision score for most classes is high, ranging from 1.0% to 98.87%. However, the precision score for the "Fist," "Pointer," and "Palm" classes is only 1.0%, which indicates that the model produced a large number of false positives for these classes. The recall score for all classes was above 88%, indicating that the model correctly identified a large proportion of actual positive instances. The F1-Score for all classes was above 93%, which indicates a high level of accuracy in detecting and recognizing hand gestures.

Compared to the previous tables, this table shows higher precision, recall, F1-score, and mAP values for most of the hand gesture classes. The precision, recall, and F1-score for the "Fist" and "Pointer" classes have significantly improved from the previous table, reaching perfect precision for the "Fist" class and near-perfect precision for the "Pointer" class. The "Ok" and "Palm" classes also showed improvement in precision and F1-score, although their recall values were 1.0, indicating that there were no false negatives. The "Thumb down" and "Thumb up" classes also demonstrated high precision and F1-score values.

YOLOv5 is a more efficient and faster version of the YOLO algorithm with comparable accuracy to YOLOv4. While YOLOv3 is still a popular choice for object detection, YOLOv5 offers improved performance and training techniques.

Despite recent enormous advancements in object detection, it is still challenging to detect and classify objects rapidly and accurately.

The YOLOv5 method was cited by Yan et al. (2021) as the most potent object-detecting algorithm available today.

In the current study, YOLOv5 outperformed YOLOv4 and YOLOv3 in terms of overall performance.

As we discovered multiple studies comparing YOLOv5 to earlier iterations of YOLO, such as YOLOv4 or YOLOv3, this conclusion is consistent with some earlier studies. Thuan (2021) claims that YOLOv5 is more precise and quick than YOLOv4.

VI. Conclusion

The proposed technique using the YOLO model can significantly enhance communication for deaf or hard-of-hearing individuals, regardless of their disability.

The recent advancements in computer vision and deep learning have improved the accuracy of object detection, and this study utilizes this progress to develop a hand gesture recognition system. The model for hand gesture recognition is based on the deep learning models YOLO (YOLOv3, YOLOv4, YOLOv4-tiny, and YOLOv5) to recognize motions and classes in sign language. The experiments conducted show that the suggested YOLO model has exceptional detection and performance, with a 99.50% accuracy rate when identifying objects and gestures from various datasets. The proposed method clusters the dataset based on the suggested algorithm, which necessitates manual annotation of a number of classes and analysis for patterns that aid in target prediction. The results demonstrate that the suggested YOLOv5 method outperformed the YOLOv3, YOLOv4, and YOLOv4-tiny algorithms in all datasets and improved the hand detection performance. By leveraging messaging or video calling, this technique can help overcome the obstacles of communication faced by deaf or hard-of-hearing individuals and enable them to interact with others more effectively. There are several directions this research can take. Another approach could be to use a combination of YOLO models for different stages of the hand gesture recognition pipeline. The YOLOv3 model could be used to detect the hand region in an image, and a YOLOv4 model could be used to classify the hand gesture. This approach could improve the accuracy of the system while also reducing the computational cost.

References

[1] R. P. Sharma, G. K. Verma, “Human computer interaction using hand gesture”. Procedia Computer Science, vol. 54, pp.721-727, 2015.

[2] D. Phutela, “The importance of non-verbal communication”. IUP Journal of Soft Skills, vol. 9, no. 4, pp. 43.

[3] A. A. Q. Mohammed, J. Jiancheng and M. S. Islam, “A deep learning-based end-to-end composite system for hand detection and gesture recognition”. Sensors, vol. 19, no. 23, pp. 5282, 2019.

[4] P. Nakjai, T. Katanyukul, “Hand Sign Recognition for Thai Finger Spelling: An Application of Convolution Neural Network”. Journal of Signal Processing Systems, vol 91, pp. 131–146, 2019.

[5] M. M. Alam, M. T. Islam, and S. M. Rahman, “A unified learning approach for hand gesture recognition and fingertip detection”. UMBC Student Collection, 2021.

[6] S. Pramada, D. Saylee, N. Pranita, N. Samiksha, N., and M. S. Vaidya, “Intelligent sign language recognition using image processing”. IOSR Journal of Engineering (IOSRJEN), vol. 3, no. 2, pp. 45-51, 2013.

[7] N. Cihan Camgoz, S. Hadfield, O. Koller, and R. Bowden, “Subunets: End-to-end hand shape and continuous sign language recognition”. In Proceedings of the IEEE international conference on computer vision, pp. 3056-3065, 2017.

[8] A. Kusters, “International Sign and American Sign Language as different types of global deaf lingua francas”. Sign Language Studies, vol. 21, no. 4, pp. 391-426, 2021.

[9] A. I. Khan, and S. Al-Habsi, “Machine learning in computer vision”. Procedia Computer Science, vol.167, pp. 1444-1451, 2020.

[10] N. O’Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, and J. Walsh, “Deep learning vs. traditional computer vision”. In Science and information conference, pp. 128-144, 2020.

[11] C. Wang, and Z. Peng, “Design and implementation of an object detection system using faster R-CNN”. In 2019 International Conference on Robots & Intelligent System (ICRIS), pp. 204-206, 2019 IEEE.

[12] M. Oudah, A. Al-Naji, and J. Chahl, “Hand gesture recognition based on computer vision: a review of techniques”. Journal of Imaging, vol. 6 , no. 8, pp. 73, 2020.

[13] H. Huang, Y. Chong, C. Nie, and S. Pan, “Hand gesture recognition with skin detection and deep learning method”. In Journal of Physics: Conference Series, vol. 1213, no. 2, pp. 022001,. 2019, IOP Publishing.

[14] Z.Q Zhao, P. Zheng, S.T Xu, and X. Wu, “Object detection with deep learning: A review”. IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212-3232, 2019.

[15] F. Sandelin, “Semantic and instance segmentation of room features in floor plans using Mask R-CNN”. 2019.

[16] K. Roy, A. Mohanty, R. R Sahay, “Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation”. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 640–649, 2017.

[17] T.H.N. Le, C. Zhu, Y. Zheng, K. Luu, and M. Savvides, “M. Robust hand detection in Vehicles”. In Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 573–578, 2016.

[18] A. Orbay, and L. Akarun, “Neural sign language translation by learning tokenization”. In 15th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 222-228, 2020.

[19] O. Koller, N.C Camgoz, H. Ney, H.; and R. Bowden, “Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos”. IEEE transactions on pattern analysis and machine intelligence, vol. 42, no. 9, pp. 2306-2320, 2019.

[20] I. Gruber, D. Ryumin, M. Hrúz, and A. Karpov, “Sign language numeral gestures recognition using convolutional neural network”. In International Conference on Interactive Collaborative Robotics, pp. 70-77, 2018.

[21] D. Ryumin, I. Kagirov, A. Axyonov, et al, “A multimodal user interface for an assistive robotic shopping cart”. Electronics, vol. 9, no. 12, pp. 2093, 2019.

[22] S. F. Chevtchenko, R. F. Vale, V. Macario, and F. R. Cordeiro, “A convolutional neural network with feature fusion for real-time hand posture recognition”. Appl. Soft Comput. J, vol. 73, pp. 748-766, 2018.

[23] C. Liang, Y. Song, and Y. Zhang, “Hand gesture recognition using view projection from point cloud”. In Proceedings of the International Conference on Image Processing, (ICIP), Phoenix, AZ, USA, 25–28, pp. 4413–4417, Septmber 2016.

[24] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, “We don’t need no bounding-boxes: Training object class detectors using only human verification”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 854-863, 2016.

[25] K. Konyushkova, J. R. R. Uijlings, C. H. Lampert, and V. Ferrari, “Learning intelligent dialogs for bounding box annotation”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9175–9184, 2018.

[26] N. D. Vo, K. Nguyen, T.V. Nguyen, and K. Nguyen, “Ensemble of deep object detectors for page object detection”. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, pp. 1-6, January 2018.

[27] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In Advances in Neural Information Processing Systems (NIPS), vol 28, 2015.

[28] H. M. Soe, and T. M. Naing, “Real-time hand pose recognition using faster region-based convolutional neural network”. In Big Data Analysis and Deep Learning Applications: Proceedings of the First International Conference on Big Data Analysis and Deep Learning 1st, pp. 104-112, Springer Singapore, 2019.

[29] P. K. Pisharady, P. Vadakkepat, and A. P. Loh, “Attention based detection and recognition of hand postures against complex backgrounds”. International Journal of Computer Vision, vol. 101, pp. 403–419, 2013.

[30] M. M. William, P. S. Zaki, B. K. Soliman, K. G. Alexsan, M. Mansour, M. El-Moursy, and K. Khalil, “Traffic signs detection and recognition system using deep learning”. In 2019 Ninth international conference on intelligent computing and information systems (ICICIS), pp. 160-166, 2019, IEEE.

[31] G. Jocher, et al, “Ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models AWS Supervise. ly and YouTube integrations”. Zenodo, vol. 11, 2021.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection”. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.

[33] X. Deng, Y. Zhang, S. Yang, P. Tan, L. Chang, Y. Yuan, and H. Wang, “Joint hand detection and rotation estimation using cnn”. IEEE transactions on image processing, vol. 27, no 4, pp. 1888-1900, 2017.

[34] Y. Li, X. Wang, W. Liu, and B. Feng, “Deep attention network for joint hand gesture localization and recognition using static rgb-d images”. Information Sciences, vol. 441, pp. 66–78, 2018.

[35] S. Shinde, A. Kothari, V. Gupta, “YOLO based human action recognition and localization”. Procedia Comput. Sci. 2018, vol. 133, pp. 831–838, 2018.

[36] J. M. Yu, W. Zhang, “Face mask wearing detection algorithm based on improved YOLO-v4”. Sensors, 2021, vol. 21, no 9, p. 3263, 2021.

[37] R. Shukla, A. K. Mahapatra, J. S. P Peter, “Social distancing tracker using yolo v5”. Turkish Journal of Physiotherapy and Rehabilitation, vol. 32 (2), pp. 1785–1793, 2021.

[38] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Gui, and W. Wang, “Face mask recognition system with YOLOV5 based on image recognition”.

In 2020 IEEE 6th International Conference on Computer and Communications. ICCC), pp. 1398-1404, 2020.

[39] L. Wang, W.Q. Yan, “Tree leaves detection based on deep learning”. In : Geometry and Vision: First International Symposium, ISGV 2021, Auckland, New Zealand, January 28-29, 2021, Revised Selected Papers 1. Springer International Publishing, pp. 26-38, 2021.

[40] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object detection with deep learning: A review”. IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212-3232, 2019.

[41] G. Marin, F. Dominio, and P. Zanuttigh, “Hand gesture recognition with jointly calibrated leap motion and depth sensor”. Multimedia Tools and Applications, vol. 75, no. 22, pp. 14991-15015, 2016.

[42] A. Memo, and P. Zanuttigh, “Head-mounted gesture controlled interface for human-computer interaction”. Multimedia Tools and Applications, vol 77, pp, 27-53, 2018.

[43] P. Bao, A. I. Maqueda, C. R. Del-Blanco, and N. Garciá, “Tiny hand gesture recognition without localization via a deep convolutional network”. IEEE Transactions on Consumer Electronics, vol. 63, no. 3, pp. 251-257, 2017.

[44] S. Biasotti, M. Tarini, and A. Giachetti, “Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition”.

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

[46] J. Redmon, A. Farhadi, “YOLO9000: Better, Faster, Stronger”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017, pp. 7263-7271. 41.

[47] A. Bochkovskiy, C. Y. Wang, and H. Y. M Liao, “Yolov4: Optimal speed and accuracy of object detection”. arXiv preprint arXiv:2004.10934,2020.

[48] Z. Jiang, L. Zhao, S. Li, and Y. Jia, “Real-time object detection method for embedded devices”. In computer vision and pattern recognition. 2020.

[49] Bochkovskiy, A. Darknet: Open source neural networks in python. 2020. Available online: https://github.com/AlexeyAB/darknet.

[50] C. -Y. Wang, H. -Y. M. Liao, Y.-H. Wu, et al, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN”. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 390-391.

[51] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, “PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment”. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9197–9206.

[52] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A Forest Fire Detection System Based on Ensemble Learning”. Forests, vol. 12, no. 2, pp. 217, 2021.

[53] B. Jabir, N. Falih, and K. Rahmani, “Accuracy and Efficiency Comparison of Object Detection Open-Source Models”. International Journal of Online & Biomedical Engineering, vol. 17, no. 5, 2021.