Decision Science Letters 13 (2024) 977–990

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Hand gesture recognition based on CNN and YOLO techniques

Maha Helal[a*], Wesam Shishah[a], Mohammed Zakariah[b] and Tariq Kashmeery[c]

[a]*College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia*
[b]*College of Computers and Information Systems, King Saud University, Riyadh, Saudi Arabia*
[c]*College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia*

C H R O N I C L E

Article history:
Received: February 28, 2024
Received in the revised format: May 25, 2024
Accepted: July 9, 2024
Available online: July 9, 2024

Keywords:
Deep learning
Computer vision
Hand gesture
American Sign Language (ASL)
alphabet recognition
YOLOv5

A B S T R A C T

Communication is essential for humanity today and in the past. However, some individuals lack verbal communication due to their innate disability and physical losses from accidents. There are sign-language communication methods developed for such people to communicate. Artificial intelligence solutions are offered to remove the disadvantaged situations of people with disabilities due to communication in daily life. Nowadays, rapidly developing image processing and artificial intelligence methods are proper solutions for the problem focused on in this study. Convolution neural network techniques, which have become very popular recently, offer solutions to many problems. On the other hand, the YOLO algorithm shows very high performance in real-time object detection. In this study, we proposed a method for identifying the alphabets which each gesture delivers. This work studied hand detection on images and classification according to hand movements. The American Sign Language (ASL) standard was used as the sign language. The most recent version of YOLO, known as YOLOv5x, is used for gesture detection. Concentrating on the Static Sign-language problem, a study was conducted on the definition of hand movements. The letters "J" and "Z" are not included in the data set because movable hand signals are required. Apart from these two letters, a total number of 24 letters are classified. The proposed model achieved a training performance of 99.45% mAP@.5. Moreover, the proposed model has a performance of 97.9% mAP@.5 on the test dataset. The results demonstrate that the model's object detection performance is excellent. A statistical analysis of the training time shows that the training time has been drastically decreased, 4.5 hours with the current model as compared to the existing models in the literature.

1. Introduction

As digital technology and network infrastructure have matured, human-computer interaction (HCI) has become a regular part of our lives. People's interest in hand gestures in HCI has grown due to how well-suited they are for computer engagement (Wu et al., 2016). Using your fingers and palm, you can express yourself through hand gestures (Ying Wu & Huang, n.d.). The requirement for any extra input device can be removed by using a hand gesture to communicate with HCI devices. A person's signature must be recognized by a real interaction between a human and a system. In response, hand gesture recognition (HGR) has emerged as a hot area in recent times in various capacities (Sagayam & Hemanth, 2017a), robotic commands (Tao et al., 2018), virtual games (Kulshreshth et al., 2017), and organic touch screens for small applications (Ng et al., 2011). Hand gesture recognition is used daily in the human communication network, especially sign language identification (Lichtenauer et al., 2008). Sign language is a visual language that uses an ordered set of expressive hand movements to describe concepts (Sharma & Singh, 2020). This is the only mode of communication for people who are deaf. As per the World Health Organization (WHO), five percent of the total population of the world (about 360 million) suffers from mild to severe hearing problems and can only interact via their geographical gestures' language (WHO, 2015).

* Corresponding author.
E-mail address: mhelal@seu.edu.sa (M. Helal)

doi: 10.5267/dsl.2024.7.002

There is still a language barrier between both the normal and speech-hearing challenged individuals since this interaction is hard for the typical person to comprehend. As a result, motion detection with computer assistance may be used to interpret across sign languages. Since it would serve as a bridge between groups, this would be advantageous.

A significant worry is the communication obstacle that occurs when deaf and hard-to-hear and voiceless individuals seek to engage with normal people who do not understand sign language. This apparent communication gap is typically bridged with the assistance of translators who convert sign language to spoken language and conversely. Unfortunately, such a device is highly costly and may not be accessible to them throughout their lives. As a result, advancements in automated detection of sign language motions would be highly advantageous to the deaf and hard-of-hearing people, as this will help to tear down the present communication gap (Sahoo et al., 2014).

Dynamic and static hand movements are possible (Mitra & Acharya, 2007; Rautaray & Agrawal, 2015a). Hand positions, also called stable hand signals, are made up of diverse forms and orientations of hands that do not communicate moving objects. A series of hand positions with related gesture information make up dynamic hand movements. Hand positions primarily comprise the sign language lexicon's fingerspelling, used for word-by-word signing of names, site names, ages, numerals, dates, and years that do not have established signals in the glossary. Optical interaction utilizing hand gestures has also gained widespread acceptability in a variety of application sectors, such as in human-computer interaction (HCI) (Liu & Wang, 2018; Pavlovic et al., 1997), human-robot involvement (HRI) (Jacob et al., 2013; Sagayam & Hemanth, 2017b), and medical interventions because it eliminates body contact using conventional interacting tools. Therefore, automated hand position identification has been a hot research subject, with several studies employing sight and electronic signal-based techniques (Cheok et al., 2019; Liu & Wang, 2018). When considering the complexity of the data-gathering procedure, vision-based systems appear to be more consumer pleasant and easy than others.
The application for real-time hand gesture detection primarily focuses on classifying and identifying gestures. In order to comprehend how a hand moves, we may employ a variety of methods and concepts from diverse disciplines, including image processing and neural networks. Hand gesture recognition has a huge variety of uses in general. For instance, we can use sign language to converse with deaf people who are unable to hear.

Despite encouraging results, traditional approaches cannot extract constant attribute descriptors for hand position identification in actual applications because of the variety of troublesome circumstances (Pisharady & Saerbeck, 2015; Rautaray & Agrawal, 2015b). The difficulties stem mostly from the inability of traditional machine learning approaches to effectively extract distinguishing data of shapes from regular raw input information. The hand position detection technique addresses the identification and breakdown of hands from photos acquired with composite background circumstances (Stergiopoulou et al., 2014). Another challenge is determining the strong elements that distinguish the geometrical differences in the appearance of the unchanged hand position displayed by various people [(Rautaray & Agrawal, 2015b). Another difficult issue, particularly in automatic sign language identification, is the high number of motion modules with relatively minimal interclass variance (Rautaray & Agrawal, 2015b). To turn the underdone pictures into the most discriminative demonstration by which the classifier can recognize and discriminate the patterns properly, statistically sophisticated image/video processing processes with extensive domain expertise are required. Another stumbling barrier in learning sign linguistic acknowledgment is an absence of widely accessible information with an adequate amount of example pictures.

In the literature of today, two basic sorts of methods are prevalent for hand gesture detection. To record and identify the gesture, one is utilizing a particular piece of equipment. The second strategy involves using deep learning to identify hand gestures. The drawback of this strategy is the high cost of deep learning model creation and processing, as well as the lack of readily accessible, reasonably priced, and widely utilizing specialized device tools on the market today. Deep Learning techniques are applied in various applications (Kataria et al., 2021; Pillai et al., 2021; S. Srivastava et al., 2021; Arora et al., 2021). The deep learning methodology is involved in multiple applications. Deep learning methodologies and advances in convolutional neural networks (CNN) outperform the traditional methodology to hand action identification because they prevent the necessity to derive composite handcrafted attitude descriptors from pictures, which is required in the conventional initialization and categorization phases (Li et al., 2019; Neto et al., 2018; Xing et al., 2018). CNN accelerates the feature extraction method by acquiring high-level picture assumptions and capturing the utmost discriminative attribute value in classified style (Affonso et al., 2017; Traore et al., 2018). As a result, it eliminates the issue of receiving irregular characteristic descriptions while functioning with a huge quantity of movement classes with very modest interclass variances. To generate quick and accurate object recognition, YOLO was built using Convolutional Neural Networks (CNN). According to the state-of-the-art, it is a very quick end-to-end object identification technique. YOLO is frequently used to forecast object detection tasks such as real-time pedestrian detection, traffic sign recognition, mask detection, etc.

In this work, we suggested a method for determining the alphabet which each gesture delivers. This work investigated hand detection on images and classification based on hand gestures. Concentrating on the Static Sign-language issue, a study was undertaken on the definition of hand movements. This work presents the following list of contributions:

- Instead of a regular classification task, here object detection, and recognition method is applied, which is a quite rare method in hand gesture tasks in the literature.
- The most recent version of YOLO, known as YOLOv5x, is used to enable future researchers to comprehend the model's utility for the task by comparing it to earlier efforts in ASL. In addition, the images were augmented before model training, which considerably improved the model's performance.

- For object detection tasks, intersection over union parameters can be chosen lower to get better results. In this work, the 0.45 IoU threshold is used, which is the default value for YOLOv5 to understand better whether the model is trained well or not.

The paper is organized as follows: Section 2 gives a brief description of various techniques applied for Hand Gesture Recognition in the Literature review, then section 3 gives details about the dataset used in this study and preprocessing methods applied to images, section 4 gives the details about the proposed methodology, section 5 gives the results, and section 6 gives discussion and comparison and section 7 with conclusions followed by the list of references.

2. Literature Review

ASL uses PCA-based elements, a Gabor filter, and an orientation-based hash code to represent the various ASL alphabets. An artificial neural network (ANN) is subsequently used to categorize the obtained traits. The efficacy of their database of 24 static motions was examined in this study. The authors of this work (Kang et al., n.d.) developed a CNN-based prototype for recognizing human gestures. The framework has been tested and trained on 31 different ASL alphabet and number classes. Similar to how the authors here (Ameen & Vadera, 2017) used another method of ASL alphabet identification, a CNN model fed both color and depth images of motions. To extract characteristics from each input in this model, two convolutional layers are used. The data from these layers are then merged and sent to an entirely associated layer for categorization. Numerous researchers have investigated employing depth sensors, such as the Microsoft Kinect, in addition to RGB photographs. A method for sign language recognition using CNN with multi-view growth and implication synthesis was demonstrated in another work.

In another study, Ansari and Harit suggested CNN model training using augmented data. The Microsoft Kinect camera was used to take depth photos of the actions. This method has significant computing requirements but achieves outstanding detection performance. Another hand gesture recognition method for ISL identification using the Kinect sensor has been seen in the research (ANSARI & HARIT, 2016). In this study, hand motions were correctly identified using a unique combination of feature extraction and machine learning techniques. Many scientists in the field used contact-based techniques for gesture recognition. Here, Chong and Kim described a method for identifying ASL using a wireless gadget (Chong & Kim, 2020).

28 ASL words were produced using six inertial measurement units (IMUs), and they were then categorized using the LSTM algorithm. Xiao et al. suggest a gesture detection technique based on recurrent neural networks (RNN) for a system that translates Chinese sign language. The signer's skeleton pattern is used for two-way communication in this piece. The effectiveness of this strategy is evaluated using common RGB-depth images of various stationary movements (Xiao et al., 2020).

Abraham et al. (2019) showed a sensor-based real-time hand gesture detection mechanism for ISL translation. Hand posture and finger motions were retrieved from sensor data and electronically transferred to the processing equipment in this work. Finally, for classification, an LSTM model is used. This model has been evaluated on 26 regularly used ISL motions (Abraham et al., 2019). For detecting ISL, Gupta and Kumar proposed a unique sensor-based system. Electromyograms and IMUs were placed on both signers' forearms to collect sign detail. This approach was classified with a 2.73 % error rate utilizing a multi-label model that focuses on the linguistic features of signals (Gupta & Kumar, 2021). Finally, a contact-based process for identifying ISL and ASL alphabets and numbers was presented by (Kakoty & Sharma, 2018). Finger and wrist joint angles are obtained and preprocessed with a rolling normal filtration mechanism after sign data is collected utilizing data gloves. For segmentation, SVM with 10-fold cross-validation is used, and a precision of 96.7 % is achieved. Ameen and Vadera suggested a CNN-based acknowledgment system for ASL alphabet symbols. They used two concurrent CNNs to extract features from both color and depth photos of motions and obtained a cognitive efficiency of 80.34 on the ASL finger-writing benchmark dataset. In another approach, (Rastgoo et al., 2018) used RBMs (Restricted Boltzmann Machines) to detect ASL fingerspelling with RGB and distance pictures in a deep learning technique. This approach used CNNs to identify hands, and the discovered hand pictures were passed into the RBM store to identify the sign tags. Their prototype was evaluated on four openly accessible databases (Massey University Gesture Dataset, ASL, and Fingerspelling Dataset from the Center for Sight, Language, and Signal Processing at the University of Surrey, NYU, and ASL Fingerspelling datasets) and outperformed the competition.

Mohanty et al. (2017) introduced another deep learning technique based on CNN with the availability of a complicated environment and variable lighting conditions to detect static hand motions (Mohanty et al., 2017). Their developed framework, which consists of two obscurity operations covered with a ReLu initiation role, was assessed with three publicly accessible standard databases, namely the NUS hand position database with a complex background (Daniels et al., 2021; Dima & Ahmed, 2021). Trish hand position database with a consistent dark framework and the Marcel hand position database produced decent identification performance on all three. In an another study, A rapid, precise fine-grain object identification framework based on YOLOv4 deep neural network was developed by (Roy et al., 2022). With DenseNet in the backbone to maximize feature transfer and reusable, two novel remaining blocks in the backbone and neck enhance feature extraction and lower computing costs, the SPP helps improve receptive field, a modified Path Aggregation Network (PANet) preserves fine-grain localized information, and a modified PANet improves feature fusion, the modified network architecture maximizes both detection accuracy and speed. This study offers a practical and efficient way for identifying

numerous plant illnesses in challenging situations, which may be expanded to identifying various fruits and crops, identifying general diseases, and utilizing numerous technological agricultural detection methods. A deep learning (DL)-based automated detection accuracy model for real-time endangered wildlife identification is presented by the author in a related paper called WilDect-YOLO (Roy et al., 2023). In the model, we include DenseNet blocks to enhance the preservation of crucial feature information and introduce a residual block in the CSPDarknet53 backbone for powerful and discriminating deep spatial feature extraction. A modified PANet and SPP have been used to increase feature fusion, maintain fine-grain localized information, and improve receptive field representation, leading to better identification in a variety of difficult settings.

According to the literature study, YOLO is an object detection model, therefore it would be effective not only for detecting and localizing the hand in an image/video but also for identifying the gesture. As a result, we present a hand gesture recognition model based on YOLOv5, which is detailed in detail in the next section.

3. Materials and Methods

3.1 Dataset

American Sign Language (ASL) is a sign language dataset widely used by deaf communities in the United States. It is a form of expression created by hand gestures and is commonly used in many countries. ASL has a set of 26 signs, as shown in Fig. 1, known as the manual alphabet, for spelling words from the English language. In this work, fingerspelling has been studied using object detection and classification methods based on Deep Learning Models. The American Sign Language Letters Dataset ("American Sign Language Letters Dataset," 2021), provided by the Roboflow platform as a public dataset is used for our task. This dataset consists of images with 720 or 1080p resolution and 26 English letters of the alphabet. Since "J" and "Z" letters require dynamic movements, most research and datasets do not include these letters, especially if the task is based on a classification. These 720 images are in RGB format and have a size of 416 x 416 to process on the YOLOv5 model. The dataset is labeled with corresponding bounding boxes on the Roboflow website, a quite popular website for labeling datasets. This website makes it easy to annotate and create data tags in the desired format.

3.2 Pre-processing

The selected ASL dataset contains 720 images collected by an ordinary camera; however, the number of images is quite scarce for training. To train the model with more images, the images were reproduced by applying the augmentation method to the images. Horizontal flip, ±10% hue change, ±15% saturation application, ±10% brightness change, ±10% exposure, and 1.5px blur application were performed to differentiate the images from their original state. Seven hundred twenty images were reproduced up to 1638 images. In Fig. 2, a few examples of the images reproduced for the letter 'A' are given. The letters 'Z' and 'J', which are emotional in ASL, were removed from the data set.

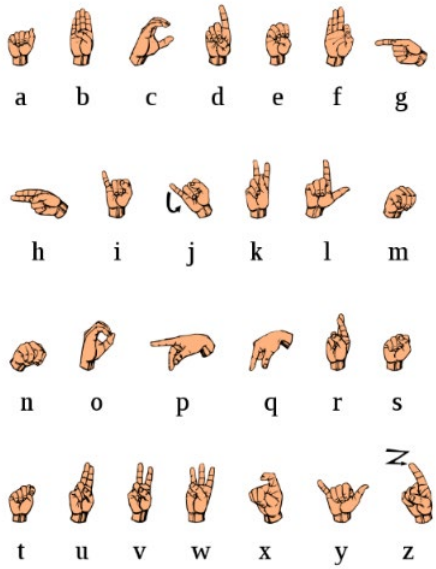
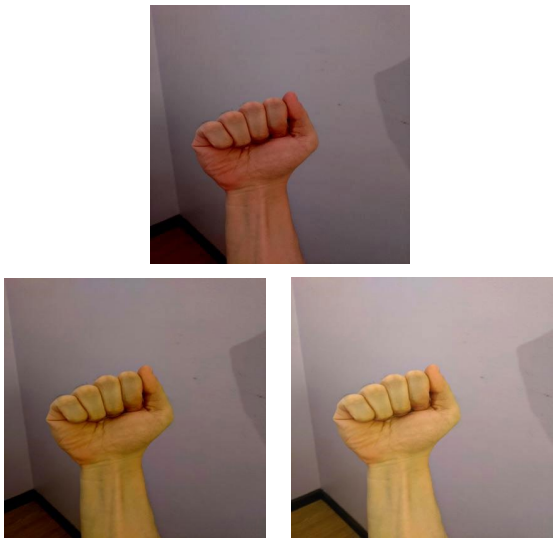


Fig. 1. Fingerspelling for letters (***Wikipedia Web Page, American Manuel Alphabet, Https://En.Wikipedia.Org/Wiki/American_manual_alphabet, 11/12/2021.***, **n.d.).**

The data was labelled using a very well-known website called Roboflow online. Images were labeled with bounding boxes for use in the YOLOv5 model. 24 classes were labeled to represent 24 letters without 'Z' and 'J' letters. The input size of the YOLOv5 model is 416x416. Therefore, images of different sizes in the dataset were converted to 416x416 by reshaping.

The new data set created by data augmentation was divided into train, validation, and testing. The dataset was set as 1392 (85%) training, 180 (11%) validation, and 66 (4%) testing.



Fig/ 2. Examples of Augmentation Images.

4. Research Methodology

In this part, the model architecture and the dataset are explained in detail.

4.1 Model Architecture

Since our main objective is to detect hands and estimate the corresponding letter, YOLO (You Only Look Once) algorithm is preferred because the effectiveness of this algorithm is relatively high, and it is a widely used algorithm for object detection. This algorithm is a one-stage object detection algorithm that considers the problem as a regression problem. Rather than subtracting the Region of Interest (RoI), it directly generates the bounding box coordinates of each class and their probability by using the regression method as shown in Fig. 3. YOLO is especially used for real-time object detection tasks as an object detection algorithm. It can predict the class and coordinates of all objects in a frame by passing them through the neural network at once. This feature is a high-speed algorithm compared to other real-time object detection methods such as Fast R-CNN, Single-Shot MultiBox Detector (SSD), etc. YOLO splits the input into S×S grids for object detection, and each grid is responsible for finding out whether there is an object in the area. The algorithm checks the following arguments; if its midpoint is in it, its length, height, and what class it is in. The anchor box, first used in the Faster R-CNN model, is used even though there is more than one object in each grid. Its logic is that the prediction of the box is performed around the object with the help of specific hand-picked patterns. Additionally, predictions are made for each grid's predetermined number of anchor boxes.

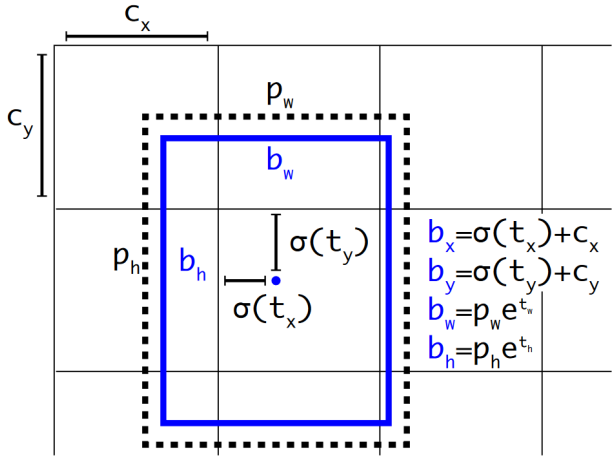


Fig. 3. Bounding Boxes (Redmon & Farhadi, 2016)

The algorithm generates so many prediction boxes for one object. Non-max Suppression algorithm is used to select the correct box among these prediction boxes. This algorithm consists of 3 steps. Firstly, all boxes with a confidence score below a certain level are removed from the prediction. Print the box with the highest confidence score if there are still boxes. Finally, all other boxes are excluded from the prediction, except the box with the highest confidence score. When this operation is performed for each object, we are left with one box for each object due to the process.

The YOLOv5 network architecture is based on a single convolutional neural network (CNN) that is trained to predict bounding boxes and class probabilities directly from full images in one evaluation. The architecture of YOLOv5 is divided into three parts: the backbone network, the neck network, and the head network. The backbone network is responsible for extracting features from the input image. YOLOv5 uses a variety of backbone architectures, such as ResNet, Darknet, and EfficientNet, which are pre-trained on the ImageNet dataset to improve the accuracy of the model. The backbone network is followed by the neck network, which is responsible for fusing the features from the backbone network and creating a feature pyramid. The neck network is implemented using the SPADE (SPatially Adaptive Normalization) module and a lightweight convolutional network. The head network is responsible for predicting the bounding boxes and class probabilities. YOLOv5 uses anchor boxes to predict the bounding boxes. Anchor boxes are pre-defined boxes of different aspect ratios and scales that are used to detect objects of different sizes and shapes. The head network is implemented using a lightweight convolutional network with multiple layers of 3x3 convolutions and a 1×1 convolutional layer that predicts the bounding boxes and class probabilities. In terms of computational complexity, YOLOv5 requires more computational resources than its predecessor YOLOv4. However, it is still relatively fast and can run in real-time on a standard GPU. The model size of YOLOv5 is also smaller than YOLOv4, which makes it more suitable for deployment on edge devices. The exact computational complexity of the YOLOv5 model depends on the specific architecture and the size of the input image. Overall, YOLOv5 is a powerful object detection system that is able to detect and classify objects in real-time with high accuracy. Its architecture is composed of a backbone, neck and head network, which work together to extract features, fuse them and make predictions. The computational complexity and model size are relatively high but still manageable for most use-cases.

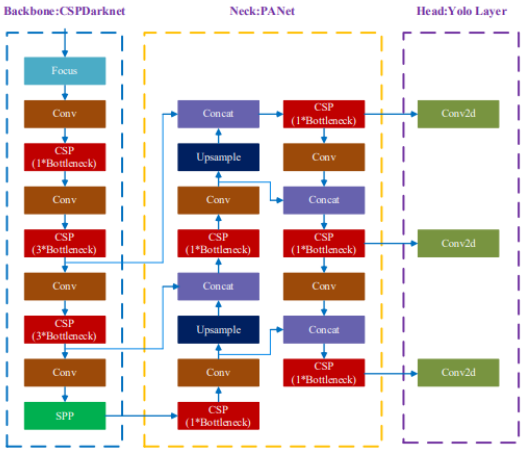


Fig. 4. The pipeline of YOLOv5 Architecture (Fang et al., 2021).

Analytics presented YOLOv5 in 2020, giving much better results than previous versions of the YOLO algorithm in terms of speed and performance. Fig. 4 displays the pipeline of YOLOv5 Architecture. The structure is like the YOLOv4 algorithm. It consists of three main parts: head, spine, and neck. The backbone section was created with CSPDarknet for feature extraction. CSPDarknet was created by incorporating CSPNets (Cross Stage Partial Networks) into the Darknet.

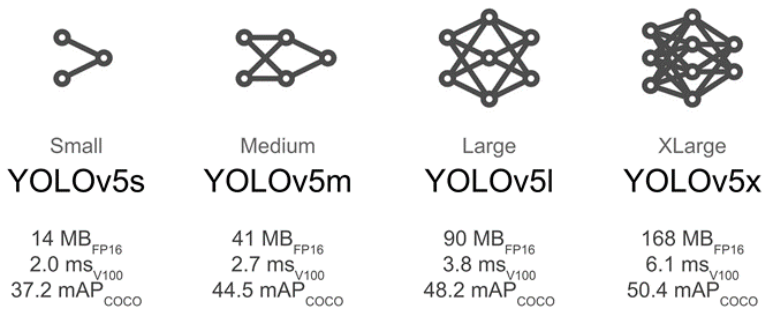


Fig. 5. Different sizes of YOLOv5 with Pretrained Models ("YOLOv5," 2021).

The neck section is used to build feature pyramids to help YOLO-v5 generalize object scaling to describe the same item in various sizes and scales. In the neck area, PANet is utilized. For feature maps, the head section generates junction boxes.

The final output vectors are produced together with class probabilities and bounding boxes for nodes that have been discovered. With just around one-fourth of the computational complexity, YOLOv5 s achieves the very same accuracy as YOLOv3-416. YOLOv5 has different versions depending on the width and depth of the backbone network. Some of the versions are shown in Fig. 5. These are YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. In this study, a deeper version, the YOLOv5x version, was chosen.

5. Results

This part considers and explains the dataset preprocessing, metrics evaluation for the model training process, and results for the train, validate, and test procedures. To match our model architectures with the chosen dataset, preprocessing is required. In consideration of our model architecture, the evaluation measures are chosen.

5.1 Evaluation Metrics

5.1.1 Precision and Recall Metrics

As a model evaluation, accuracy is not enough to check the correctness of this kind of task. The modeled ratio calculates the predicted areas' accuracy value to the entire data set. Model accuracy alone is sufficient, especially in unbalanced distributed data sets. However, just looking at the accuracy metric can be misleading in object detection and recognition cases. The Confusion Matrix table in Fig. 6 is often used in model evaluation. The Confusion matrix table shows the actual and predicted values in a classification problem. True Positives represent a situation where the model correctly predicts the true classes. True Negative describes a situation where the model predicts false, and it is false. False Positive refers to the situation where the model says the prediction is true but false. False Negative, on the other hand, refers to the case where the model says it is wrong as a prediction, but it is correct.

		Prediction	
		Positive	Negative
Actual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Fig. 6. Predicted Training Data Set.

When the accuracy metric is insufficient in model evaluation, the concepts of recall and precision emerge. The precision metric displays the proportion of positive values that match our estimates. When a False Positive estimation costs high, precision value is crucial. Contrarily, recall is a statistic that demonstrates the proportion of operations that we must estimate as positive; we estimate as positive. The recall value is the statistic that aids us in situations where the cost of estimating as a False Negative is high. The precision and recall formulas are listed below.

$$Precision = TP/(TP+FP) \tag{1}$$

$$Recall = TP/(TP+FN) \tag{2}$$

where *TP* = True Positive, *FP*= False Positive and *FN*= False Negative.

5.1.2 Mean Average Precision

Mean Average Precision (mAP) is a widely used metric in object detection problems. It compares object detection models such as YOLO, R-CNN, etc. It has also been used to evaluate applications in competitions such as the COCO and PASCAL VOC competitions. The Intersection Over Union (IoU) ratio is a popular metric for assessing an input's correctness in object detection algorithms. Many object detection algorithms employ a measure known as Intersection-Over-Union or Jaccard Index. IoU is a relatively straightforward and highly effective metric. The connected region between the predicted bounding box and the real area divides the overlap area between the anticipated bounding box area and the underlying real area by the IoU. IoU metrics have a scale from 0 to 1. Between 0 and 1, there is a complete overlap. With a preset threshold, intersection over union rate predicts the outcome. A graph showing precision as a recall function is known as the precision-recall (PR) curve. The graph shows the balance between the two measurements for various model detection confidence levels. High sensitivity results from low FP. However, more occurrences of items may be overlooked, leading to high FN and low recall. Conversely, the recall will increase if more positives are taken into account by lowering the IoU threshold, but false positives may also rise, lowering the precision score. A good model should still have excellent sensitivity and

recall even if the confidence threshold shifts. Ideally, AP @ is the Area Under the PR Curve (AUC-PR). The definition of AP in mathematics

$$AP@\alpha_{11} = \frac{1}{11}\sum_{r\epsilon R} p_{interp}(r) \tag{3}$$

$$p_{interp}(r) = maxp(r') \tag{4}$$

Each class's AP score is determined separately. This indicates that there is roughly the same number of (loose) AP values as there are classes. Next, the average of these AP values yields the measurement: Mean Average Precision (mAP). Finally, the AP values across all classes are averaged to create the Mean Average Precision.

$$mAP@_{\alpha} = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{5}$$

$for\ n\ classes$, $AP = Average\ Precission$, I ϵ R

5.2 Train, Validate, and Test Results

After preprocessing, the data set is ready for training. The Google Collaboratory (Colab) platform was used for training. Google Colab is a cloud service to assist machine learning research and studies. The Google Brain team developed it. It allows you to work on the cloud in the Python programming language. Google Colab offers you GPU and TPU usage services on the cloud. The Graphics Process Unit (GPU) helps them perform quickly in training and inference. Its parallel processing power and wide bandwidth are very effective in matrix and vector operations. Tesla P100-PCIEE series 16 GB model is used as Colab Pro GPU. Python 3.9 version was used as the coding language. Stochastic Gradient Descent (SGD) was chosen as the optimizer of the YOLOv5 model. The model is customized for 24 classes, each representing a letter. mAP@.5, precision and recall were used as evaluation metrics. The model was trained with 16 batch sizes and 300 epochs. An increasing success rate was observed throughout the training. The algorithm records two different training weights. One of them has the highest success rate. The other one records the latest training parameters. In Fig. 7, the metric values recorded in each epoch are graphically visualized.

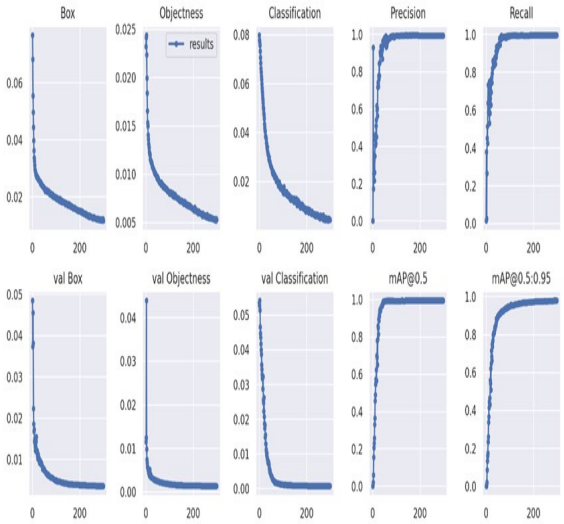


Fig. 7. Train History

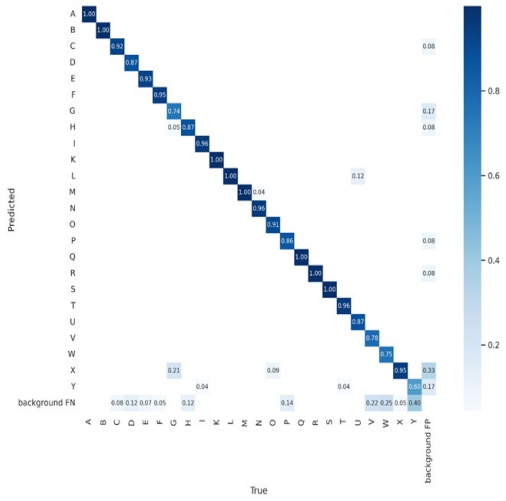


Fig. 8. Confusion Matrix of Test Dataset.

When the graphs are examined, it is observed that very successful learning is obtained. It is seen that the mAP@.5 evaluation metric performed on the validation data set increases up to 99.45%. At the same time, a 97.9% mAP@ 0.5:0.95 evaluation metric is seen following this. Box and Objectless evaluation parameters represent the loss concerning the labeled and predicted bounding boxes. These evaluation metrics are performed on both training and validation datasets, therefore val Box and val Objectless correspond to the evaluation on the validation dataset. The classification part considers the class of the object detected in the predicted bounding box concerning the actual label in the actual predicted box. Since the training dataset contains augmented data, the classification metric is not reaching zero quickly. Instead, it takes a long time to increase the number of epochs. However, it can be easily said that the model can classify the validation dataset after a short time, around 50 epochs. It can be said that quite high performance is achieved by using this model for classification tasks with high precision and recall values. Figure 8 shows the model output in the random images selected from the training dataset. It is observed that the prediction is performed correctly in the images containing hand movements representing the letters Q, R, and S. There are also versions of an image with the augmentation method applied in the images. However, the training dataset should not be used for evaluating the model. Fig. 9 displays some of Predicted Training Data Set.

The test data set consisting of 509 images is not evaluated only with the confusion matrix table. Precision, recall, and mAP@.5 metrics were also calculated. In addition, the values of these metrics were calculated for each letter and as the mean of all classes. These values are given in Table 1. It is seen that the metric values of precision, recall, and mAP@.5 are pretty high. The visualization of the test dataset evaluation results on each Letter for Precision-Recall and Map@.5 are displayed in figure 10. It is observed that the prediction success of the model is relatively high. To evaluate the prediction success of the model, visuals that the model did not see during the training are used. These images were previously reserved as the test dataset. By making estimations on the model test data set, evaluation metrics are created based on the accuracy of these estimations. The confusion matrix table is the most widely used evaluation tool. Table 1 shows the confusion matrix table on the test data set of the model. When the table is examined, it is observed by deviation for the letters “G” and “Y”. The model either makes these letters look like other letters or does not make any predictions. However, it is seen as a fair margin of error. Looking at the table generally, the model can also make successful predictions on visuals it has not seen before. It shows that this model is very successful in education and also shows that the model is generalizable.

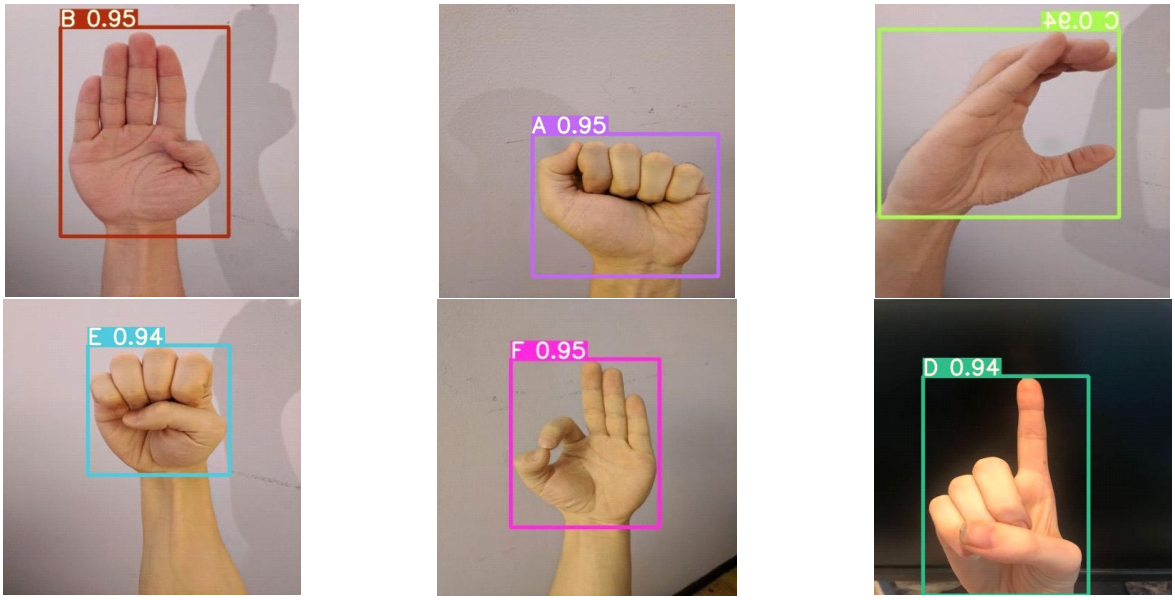


Fig. 9. Predicted Training Data Set.

Table 1
Evaluation Metrics for Test Dataset.

Class	Precision	recall	map@.5
a	0.968	0.969	0.992
b	0.997	0.996	0.994
c	0.962	0.984	0.974
d	0.908	0.954	0.994
e	0.993	0.969	0.996
f	0.987	1.00	0.995
g	0.996	0.976	0.996
h	0.924	1.00	0.995
ı	0.940	1.00	0.996
k	0.999	1.00	0.996
l	1.00	0.947	0.994
m	0.997	1.00	0.996
n	0.998	1.00	0.996
o	0.997	1.00	0.996
p	0.948	0.960	0.996
q	0.979	1.00	0.995
r	0.815	0.965	0.985
s	0.997	1.00	0.996
t	0.992	1.00	0.995
u	0.940	0.939	0.989
v	1.00	0.884	0.995
w	0.988	0.789	0.989
x	0.995	1.00	0.995
y	0.878	0.897	0.968

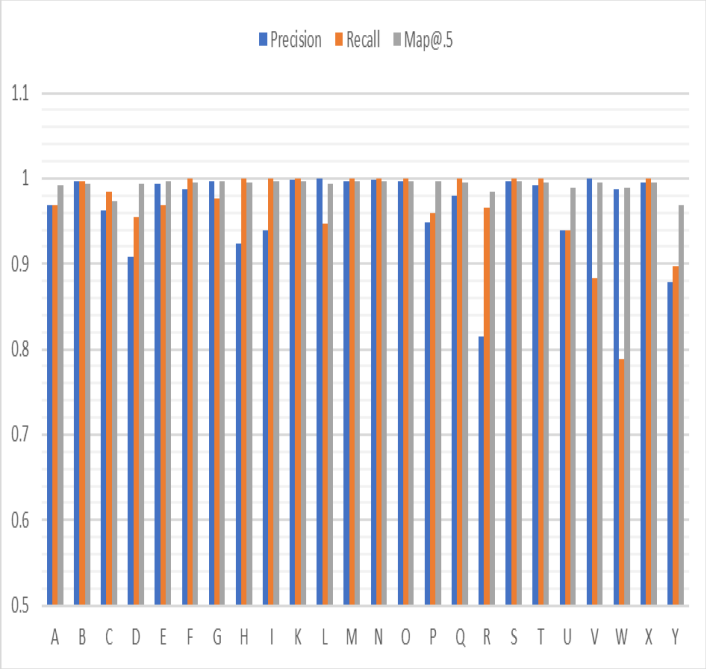


Fig. 10. Test Dataset Evaluation Results on each Letter for Precision-Recall and Map@.5

6. Discussion and Comparison

Detection and recognition tasks are prevalent, with much work on American Sign Language. Many different methods have been tried to provide a solution to this problem. Many of them offer artificial intelligence solutions. Especially nowadays, studies on such issues have increased with the popularization of deep learning methods. On the other hand, the model architecture used here, YOLOv5, is also a pretty popular model for detection and recognition tasks. This model has several applications on different types of datasets, both test datasets and real-time applications. The performance is critically better than other proposed state-of-art model structures on these object detection and recognition tasks. So far, the number of works done using YOLO on sign language detection problems is relatively low. Therefore, in this study, the performance is measured by using several evaluation metrics. The successful results are obtained using different datasets, data-augmentation methods, and preprocessing methodologies based on the dataset and YOLOv5 model architecture. Here, a comparison is made with similar works that offer deep-learning solutions to the problem.

6.1 Comparative Analysis

In 2021, Tasnim Ferdous Dima and his team described their ASL work in their article “Using YOLOv5 Algorithm to Detect and Recognize American Sign Language” (“American Sign Language Letters Dataset,” 2021). The YOLOv5m model was used to detect and describe the ASL language. Finger-spelled vocabulary, a subset of the ASL language, was chosen as the data set. The data consists of a total of 2515 RGB images collected manually. These data were replicated to 6033 images using different augmentation methods. The background of the images consists of a black pattern, as can be seen in Fig. 11.

Fig. 11. Example of Image's Dataset.

Model training was carried out on Google Colab, a cloud service. Model Python version 3.8 was used, and the Tesla K80 model with 12GB memory was used as the graphics card model. The learning rate value was selected as 0.01 and the model was trained for 300 epochs with a 16-batch size. Precision, recall, and mAP@.5 metrics were used to evaluate model training.

Table 2

Hardware and Software Comparison.

	Referred Study (Roy et al., 2022, 2023)	**This Study**
Model Architecture	YOLOv5m	YOLOv5x
GPU	Tesla K80	Tesla P100
Memory	12 GB	16 GB
Python version	3.8	3.9
Training Process	9 hours	4.5 hours

The critical part here, in our study the size of the model, is increased by selecting YOLOv5x instead of YOLOv5m and the required training time is reduced to 4.5 hours by using Google Colab Pro version which provides better hardware facilities as it can be seen in table 2. Result comparison of the proposed model with the referred study of ("American Sign Language Letters Dataset," 2021) in terms of batch size, epoch, and mAP are shown in Table 3.

Table 3

Result comparison with respect to the referred study.

	Referred Study (Dima & Ahmed, 2021)	**This Study**
Batch Size	16	16
Epoch	300	300
mAP	0.4	0.5
mAP Results	0.987	0.992

However, only the value of the mAP@.5 metric is shared in the article. The mAP@.5 value, on the other hand, has a very high value of 0.987. When compared with the YOLOv5x model used in this study, it is seen that the results are close to each other. Higher performance is shown with the YOLOv5x model. At the same time, different environmental environments were selected as the background of the data set used in this study so that the model could be generalized. Another study was carried out by Daniels et al. (2021) in 2021. Unlike this study, it was carried out on the Indonesian Sign Language (ISL) standard. The study was carried out on a data set consisting of 160-200 images per class and 4,547 images in 640x480 dimensions and RGB format. The letters "J" and "R" have been omitted because they contain movable hand signs. YOLOv3 was chosen as the model, the older version of the YOLO architecture used in this study. Additionally, they have applied the Transfer Learning method by using pre-trained weights for ImageNet.

Mainly changing the intersection threshold over the union evaluation metric is tested in this study since they have tested the results for different threshold values of IoU. As they decreased the threshold, they reached higher evaluation metrics results. Therefore, our study here selects the IoU threshold as 0.45. Since there is not much information about the evaluation process of the study, only a few metrics were able to compare. Also, recall, precision, and F1 score were used as evaluation metrics on video data. The maximum recall value of the model trained with different parameters is 93.1%, and the precision value is 80.56%. A very high-performance difference is observed when YOLOv5 and YOLOv3 are compared.

7. Conclusions

Sign language is crucial for bridging the communication gap between hearing and deaf individuals. This study focused on American Sign Language (ASL) and was conducted on the perception and identification of hand-written letters using artificial intelligence. This study utilized an open-source American Sign Language (ASL) dataset. Compared to earlier studies, the size of the data set was expanded using various augmentation techniques. The procedure is applied at the preprocessing stage of the YOLOv5 model training, which has a 416x416 input shape. At the evaluation stage, it is seen that this dataset has reached high mAP@.5 values. The mAP@.5 measure is a crucial criterion for comparing the trained model's performance to that of other models. With a high mAP@. With five scores, this study demonstrates that the model's object detection performance is excellent. Similarly, high recall and precision levels are also seen. It is possible to say that the evaluation metrics of the test and training data sets are comparable. The proposed model achieved a training performance of 99.45% mAP@.5. Moreover, the proposed model has a performance of 97.9% mAP@.5 on the test dataset. The results demonstrate that the model's object detection performance is excellent. A statistical analysis of the training time shows that the training time has drastically decreased, 4.5 hours with the current model as compared to the existing models in the literature.

Based on this parallelism, it can be concluded that the model does not overfit the training and validation datasets. As a result of this study, individuals with disabilities can successfully utilize deep learning approaches. Overall, it is anticipated that this study will promote the collection of knowledge and the development of intelligent-based SLR, as well as provide readers, researchers, and practitioners with a road map for future direction. In the future, we will concentrate on hybrid solutions including smart mobile applications or robotics in many circumstances. The dataset can be expanded to enable the

system to recognize additional gestures. Also, comparisons can be conducted using various YOLOv5 architectures, including small, medium, and big versions. The system can be constructed for multiple sign languages by modifying the dataset.

Limitations

The important limitation of the model is that, unlike previous YOLO versions, YOLOv5 has not been the subject of an official formal publication. Additionally, YOLO v5 is still under development, and as we frequently receive updates from Ultralytics, developers may later tweak various parameters

Funding Statement

The author(s) received no specific funding for this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

Abraham, E., Nayak, A., & Iqbal, A. (2019). Real-Time Translation of Indian Sign Language using LSTM. *2019 Global Conference for Advancement in Technology (GCAT)*, 1–5. https://doi.org/10.1109/GCAT47503.2019.8978343

Affonso, C., Rossi, A. L. D., Vieira, F. H. A., & de Carvalho, A. C. P. de L. F. (2017). Deep learning for biological image classification. *Expert Systems with Applications*, *85*, 114–122. https://doi.org/10.1016/j.eswa.2017.05.039

Ameen, S., & Vadera, S. (2017). A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. *Expert Systems*, *34*(3). https://doi.org/10.1111/exsy.12197

American Sign Language Letters Dataset. (2021). *Roboflow Web Page*.

ANSARI, Z. A., & HARIT, G. (2016). Nearest neighbour classification of Indian sign language gestures using kinect camera. *Sadhana*, *41*(2), 161–182. https://doi.org/10.1007/s12046-015-0405-3

Arora, P., Chaudhary, G., Crespo, R. G., Khari, M., & Srivastava, S. (2021). *Concepts and real-time applications of Deep Learning*. Springer.

Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, *10*(1), 131–153. https://doi.org/10.1007/s13042-017-0705-5

Chong, T.-W., & Kim, B.-J. (2020). American Sign Language Recognition System Using Wearable Sensors with Deep Learning Approach. *The Journal of the Korea Institute of Electronic Communication Sciences*, *15*(2), 291–298.

Daniels, S., Suciati, N., & Fathichah, C. (2021). Indonesian Sign Language Recognition using YOLO Method. *IOP Conference Series: Materials Science and Engineering*, *1077*(1), 012029. https://doi.org/10.1088/1757-899X/1077/1/012029

Dima, T. F., & Ahmed, Md. E. (2021). Using YOLOv5 Algorithm to Detect and Recognize American Sign Language. *2021 International Conference on Information Technology (ICIT)*, 603–607. https://doi.org/10.1109/ICIT52682.2021.9491672

Fang, Y., Guo, X., Chen, K., Zhou, Z., & Ye, Q. (2021). Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model. *BioResources*, *16*(3), 5390–5406. https://doi.org/10.15376/biores.16.3.5390-5406

Gupta, R., & Kumar, A. (2021). Indian sign language recognition using wearable sensors and multi-label classification. *Computers & Electrical Engineering*, *90*, 106898. https://doi.org/10.1016/j.compeleceng.2020.106898

Jacob, M. G., Wachs, J. P., & Packer, R. A. (2013). Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images. *Journal of the American Medical Informatics Association*, *20*(e1), e183–e186. https://doi.org/10.1136/amiajnl-2012-001212

Kakoty, N. M., & Sharma, M. D. (2018). Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove. *Procedia Computer Science*, *133*, 55–62. https://doi.org/10.1016/j.procs.2018.07.008

Kang, B., Tripathi, S., & Nguyen, T. Q. (n.d.). Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth map. *ArXiv (Cornell University)*.

Kataria, G., Gupta, A., Kaushik, V. S., & Chaudhary, G. (2021). *Emotion Recognition from Speech Signals Using Machine Learning and Deep Learning Techniques* (pp. 63–73). https://doi.org/10.1007/978-3-030-76167-7_4

Kulshreshth, A., Pfeil, K., & LaViola, J. J. (2017). Enhancing the Gaming Experience Using 3D Spatial User Interface Technologies. *IEEE Computer Graphics and Applications*, *37*(3), 16–23. https://doi.org/10.1109/MCG.2017.42

Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., Tao, B., Xu, S., & Liu, H. (2019). Hand gesture recognition based on convolution neural network. *Cluster Computing*, *22*(S2), 2719–2729. https://doi.org/10.1007/s10586-017-1435-x

Lichtenauer, J. F., Hendriks, E. A., & Reinders, M. J. (2008). Sign Language Recognition by Combining Statistical DTW and Independent Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(11), 2040–2046. https://doi.org/10.1109/TPAMI.2008.123

Liu, H., & Wang, L. (2018). Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics*, *68*, 355–367. https://doi.org/10.1016/j.ergon.2017.02.004

Mitra, S., & Acharya, T. (2007). Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *37*(3), 311–324. https://doi.org/10.1109/TSMCC.2007.893280

Mohanty, A., Rambhatla, S. S., & Sahay, R. R. (2017). *Deep Gesture: Static Hand Gesture Recognition Using CNN* (pp. 449–461). https://doi.org/10.1007/978-981-10-2107-7_41

Neto, G. M. R., Junior, G. B., de Almeida, J. D. S., & de Paiva, A. C. (2018). *Sign Language Recognition Based on 3D Convolutional Neural Networks* (pp. 399–407). https://doi.org/10.1007/978-3-319-93000-8_45

Ng, W. L., Ng, C. K., Noordin, N. K., & Mohd. Ali, B. (2011). *Gesture Based Automating Household Appliances* (pp. 285–293). https://doi.org/10.1007/978-3-642-21605-3_32

Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 677–695. https://doi.org/10.1109/34.598226

Pillai, M. S., Chaudhary, G., Khari, M., & Crespo, R. G. (2021). Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning. *Soft Computing*, *25*(18), 11929–11940. https://doi.org/10.1007/s00500-021-05576-w

Pisharady, P. K., & Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, *141*, 152–165. https://doi.org/10.1016/j.cviu.2015.08.004

Rastgoo, R., Kiani, K., & Escalera, S. (2018). Multi-Modal Deep Hand Sign Language Recognition in Still Images Using Restricted Boltzmann Machine. *Entropy*, *20*(11), 809. https://doi.org/10.3390/e20110809

Rautaray, S. S., & Agrawal, A. (2015a). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, *43*(1), 1–54. https://doi.org/10.1007/s10462-012-9356-9

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. *Computer Vision and Pattern Recognition*.

Roy, A. M., Bhaduri, J., Kumar, T., & Raj, K. (2023). WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection. *Ecological Informatics*, *75*, 101919. https://doi.org/10.1016/j.ecoinf.2022.101919

Roy, A. M., Bose, R., & Bhaduri, J. (2022). A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Computing and Applications*, *34*(5), 3895–3921. https://doi.org/10.1007/s00521-021-06651-x

Sahoo, A. K., Mishra, G. S., & Ravulakollu, K. K. (2014). Sign language recognition: State of the art. *ARPN Journal of Engineering and Applied Sciences*, *9*(2), 116-134.

Sagayam, K. M., & Hemanth, D. J. (2017a). Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, *21*(2), 91–107. https://doi.org/10.1007/s10055-016-0301-0

Sagayam, K. M., & Hemanth, D. J. (2017b). Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, *21*(2), 91–107. https://doi.org/10.1007/s10055-016-0301-0

Sharma, S., & Singh, S. (2020). Vision-based sign language recognition system: A Comprehensive Review. *2020 International Conference on Inventive Computation Technologies (ICICT)*, 140–144. https://doi.org/10.1109/ICICT48043.2020.9112409

Srivastava, S., Chaudhary, G., & Shukla, C. (2021). Text-Independent Speaker Recognition Using Deep Learning. *EAI/Springer Innovations in Communication and Computing*, 41–51.

Stergiopoulou, E., Sgouropoulos, K., Nikolaou, N., Papamarkos, N., & Mitianoudis, N. (2014). Real time hand detection in a complex background. *Engineering Applications of Artificial Intelligence*, *35*, 54–70. https://doi.org/10.1016/j.engappai.2014.06.006

Tao, W., Leu, M. C., & Yin, Z. (2018). American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence*, *76*, 202–213. https://doi.org/10.1016/j.engappai.2018.09.006

Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. *Ecological Informatics*, *48*, 257–268. https://doi.org/10.1016/j.ecoinf.2018.10.002

Wikipedia Web Page, American Manuel Alphabet, https://en.wikipedia.org/wiki/American_manual_alphabet, 11/12/2021. (n.d.).

Wu, C.-H., Chen, W.-L., & Lin, C. H. (2016). Depth-based hand gesture recognition. *Multimedia Tools and Applications*, *75*(12), 7065–7086. https://doi.org/10.1007/s11042-015-2632-3

Xiao, Q., Qin, M., & Yin, Y. (2020). Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks*, *125*, 41–55. https://doi.org/10.1016/j.neunet.2020.01.030

Xing, K., Ding, Z., Jiang, S., Ma, X., Yang, K., Yang, C., Li, X., & Jiang, F. (2018). Hand Gesture Recognition Based on Deep Learning Method. *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, 542–546. https://doi.org/10.1109/DSC.2018.00087

Ying Wu, & Huang, T. S. (n.d.). Human hand modeling, analysis and animation in the context of HCI. *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, 6–10. https://doi.org/10.1109/ICIP.1999.817058

YOLOv5. (2021). *GitHub Web Page - Https://Github.Com/Ultralytics/Yolov5/Wiki/Train-Custom-Data*.