International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2022, 6 (1), 61-65
DOI: DOI: 10.36287/ijmsit.6.1.61
Received: May 21, 2022; Accepted: June 20, 2022

American Sign Language Recognition using YOLOv4 Method

Ali Mahmood AL-Shaheen[1] *, Mesut Çevik (Adviser)[2], Alzubair Alqaraghuli[3]

[1*]*Electrical and Computer Engineering, Altinbas Uni, Istanbul, Turkey (ali4realx@gamil.com) (ORCID: 0000-0002-9668-9556)*
[2] *Electrical and Computer Engineering, Altinbas Uni, Istanbul, Turkey (Mesut.cevik@altinbas.edu.tr) (ORCID: 0000-0000-0299-9076)*
3 Electrical and Computer Engineering, Altinbas Uni, Istanbul, Turkey Country (zubairsk53@gmail.com) (ORCID: 0000-0002-6117-8051)

Abstract – Sign language is one of the ways of communication that is used by people who are unable to speak or hear (deaf and mute), so not all people are able to understand this language. Therefore, to facilitate communication between normal people and deaf and mute people, many systems have been invented that translate gestures and signs within sign language into words to be understandable. The aim behind this research is to train a model to be able to detect and recognize hand gestures and signs and then translate them into letters, numbers and words using You Only Look One (YOLO) method through pictures or videos, even in real time. YOLO is one of the methods used in detecting and recognizing things that depend in their work on convolutional neural networks (CNN), which are characterized by accuracy and speed in work. In this research, we have created a data set consisting of 8000 images divided into 40 classes, for each class, 200 images were taken with different backgrounds and under lighting conditions, which allows the model to be able to differentiate the signal regardless of the intensity of the lighting or the clarity of the image. And after training the model on the dataset many times, in the experiment using image data we got a very good results in terms of MAP = 98.01% as an accuracy and current average loss=1.3 and recall=0.96 and F1=0.96, and for video results it has the same accuracy and 28.9 frame per second (fps).

Keywords *– American sign language, Real-Time Detection, You Only Look Once, YOLO, CNN, Recognition, Hand Gestures, Computer Vision, Machine Learning, Deep Learning.*

Citation: Al-Shaheen, A. M., Cevik, M., Alqaraghuli, A. (2022). American Sign Language Recognition using YOLOv4 Method. International Journal of Multidisciplinary Studies and Innovative Technologies, 6(1): 61-65.

I. INTRODUCTION

Sign language is one of the forms of visual communication between (deaf, mute people) and normal people through signs, gestures, and hand movements. Because of many normal people don't understand sign language because it's not wanted, they will not be able to communicate with deaf and mute people. With the great development in this technology and artificial intelligence, computer vision technology has got a great development, as it's used in building projects and systems for detecting and classifying objects according to certain specifications, including detection and recognition of hand signs and movements. Previous researches has been conducted in detection and recognition sign language, we can divide it into two parts. The first one is the use of special apparatuses like the accelerometer [1], the sensory glove [2], or the Kinect [3]. All of these types depend in their work on capturing sensory signals from equipment with motion sensors to identify the captured signals. This type of equipment provides high accuracy, but there will be a need to use a variety of equipment that is expensive and unwieldy for daily use. And for the second type, it's based on vision, which isn't expensive and available, and requires only a camera to take pictures or videos and is suitable for daily use, as we can use phone cameras or web cameras. For example, there is a research in which the k-Nearest Neighbors (KNN) classifier was used to classify, discover and identify signs [4]. Convolutional neural networks were once used to detect and identify American Sign Language Signs, and Google Net was used in the CNN architecture, which was trained on the ILSVRC dataset in 2012. This training gained an accuracy of 72%, and this shortage of accuracy is because of the similarity between the signs of some letters such as g, h, m, n, s, t. [5] R-CNN It is a method that combines the proposals of region and convolutional neural networks where you localize the sores in a convolutional neural network and then train a high capacity model with little annotated detection data. This algorithm achieves good accuracy in object detection by using the deep network ConvNet to classify object proposals. Also, it has the ability to detect thousands of objects without the need for approximate techniques. One of its drawbacks is the weakness in detecting small things, and also that it predicts only a sign and not a hash square. [12] [20] [21] [22] Fast R-CNN It's an algorithm written in Python programming language and is a training algorithm for object detection, developed to solve R-CNN problems regarding to accuracy and speed. One of its advantages is that it has a high accuracy than R-CNN in objects detecting and it also trains in one stage and the training can update all layers of the network. But one of its drawbacks is that it's slow when detecting objects, and this is due to the slow establishment of the selective search area. [13] [17] [18] [19] Faster R-CNN It's an object detection method similar to R-CNN that uses a Region Proposal Network (RPN), which is

effectively shares the convolutional features of the image with the detection network. Faster R-CNN is much faster than Fast R-CNN and R-CNN because it's used RPN (Region Proposal Network) for generating anchor boxes (region proposals). And It can be used in real-time object detection. There is one drawback in this method, which is after the RPN is trained, all the anchors are extracted in the mini-batch, because of the great similarity in the features of all the objects in the same image, which causes slow detection and the network may take a long time to reach a point of similarity between the objects. [14] [15] [16] YOLO is a newly recognition methods in deep learning, which has been developed on the basis of CNN, it's more effective and accurate in objects detection [6]. Deep learning models have achieved great success and tremendous developments in the field of object detection, including YOLO, which is a highly efficient model capable of differentiating and detecting objects through images, videos, or in real-time. It's an algorithm that depends in its work on the principle of regression, which means that instead of detecting and specifying the clear and interesting part, it analyzes and predicts the categories and boxes that surround the entire image, and that is done through one run of the algorithm. YOLO used in researches that require the detection of certain objects, or in determining the time, detection of traffic lights, pedestrians, cars plates and parking spaces, people, animals, etc. [10].

II. THEORY

2.1 Object Detection

Object detection It is a computer technology directly relies on computer vision and image processing that deals with recognition the states of objects within certain categories (such as people, cars, animals, hand movements, etc.) through images or videos. This technology is used in several areas, including the discovery of traffic lights and car plates for security reasons, as well as the facial recognition system in the protection systems of phones. It is also used to track certain things such as the movement of pedestrians, cars, etc. [23] [24] [25]

2.2 Convolutional Neural Networks (CNN)

CNN are a type of neural networks used to process data with a network structure such as images. It consists of neurons with weights, biases and activation functions. These neural networks consist of two layers, the feature extraction layer and the fully connected layer. The feature extraction layer consists of two layers, the convolutional layer and the pooling layer. The idea of convolutional neural networks was inspired by the shape of the cortex of the human brain. The artificial neurons in a CNN connect to a local area of the visual field, called the receptive field. This is achieved by performing discrete convolutions on the image using filter values as trainable weights. Multiple filters are applied to each channel, and with neuronal activation functions, they form feature maps. This is followed by the aggregation scheme, where only the interesting information of the feature maps is grouped together. [7] [8]

These techniques are implemented in various layers as shown in the figure (1).

.

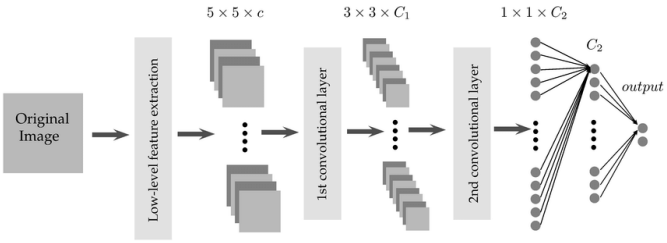

Fig. 1. Deep learning Model Architecture [8]

2.3 You Only Look Once

The YOLO algorithm used in the field of computer vision. It's able to classify the objects inside a particular image (human, fruit, car) in addition to determining the location of these objects inside the image (Object detection). YOLO algorithm is an acronym for "You Only Look Once", meaning that it requires only one pass (forward propagation) through the convolutional neural network to discover multiple objects within an image, so that the image is divided into regions and the bounding box is predicted and the probabilities for each region. [9]

2.4 YOLOv4

YOLOv4 is a real-time object discovery model developed and published in 2020 by three developers, Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, and it has achieved impressive performance. YOLOv4 depends in its work on convolutional neural network (CNN), this neural network divides the image into a group of regions and predicts the boxes of limits and probability for each region. This version of YOLO has been developed in terms of speed and accuracy, now YOLOv4 has high accuracy and speed compared to the rest of the versions. The average accuracy (AP) and the fps (frames per second) increased by 10% and 12% compared to YOLOv3. The YOLOv4 object recognition model used in this research is mainly divided into three sections:

1) Backbone Model: Extracts features from the input image.

2) Neck Model: Generates a feature pyramid.

The Neck Model helps the object detector to detect the same object at different scales, thus helping with the generalization of the convolution neural network.

3) Head Model: Obtains features from the previous section and predicts the bounding-box area and the associated class, as shown in the figure (2) below. [10] [11]

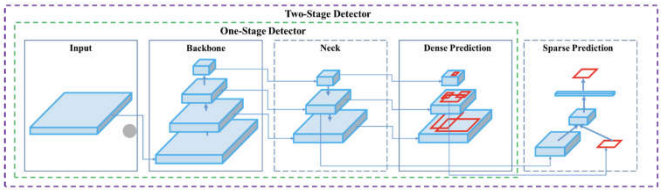

Fig. 2.: YOLOv4 Architecture [10]

III. DESIGN

3.1. Dataset

3.1.1. Image Collecting

The dataset used as input data for the recognition system was taken by the author in the form of a set of 8000 images divided into 40 classes, each class representing a number and a letter

from the alphabet and some words, with a total of 200 images for each class as shown in the table (1) below.

Table 1. Dataset Specification

Specification	Value
Resolution	1920*1080
Extension	.JPG
Number of images	8000
Number of class	40
Number of images per class	200
Image size	900-1000 Kb

The data set was divided into two parts, the first for training and the second for testing, it was divided by 80% for training to 20% for testing under different lighting conditions and divided into four groups as shown in the table (2) below.

Table 2. Dataset Conditions

Conditions	Description
Condition1	Brightness, taken from 20-30 cm away
Condition2	Brightness, taken from 50-60cm away
Condition3	Low Brightness, taken from 20-30 cm away
Condition4	Low Brightness, taken from 50-60cm away

(a) Condition1

(b) condition2

(c) condition3

(d) condition4

Fig. 3. dataset conditions

3.1.2. Image labelling

Each class was given a name using "labelimg" as a Labelling program which is draws a box around the object to be detected and specifies five values for the object, which are a specific value or number for the class, two values represented by X and Y for the object location, and other values for the object size, and Then save these values into a XML file as shown in the figure (4) and figure (5) below. This process is done for each picture in the class for one and also for the rest of the classes.

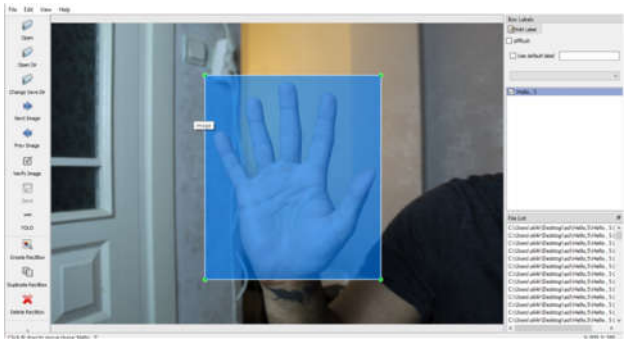

Fig. 4. Image labelling using labelimg

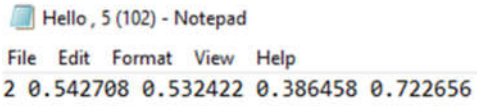

Fig. 5. Object Values

3.2. Model training and results

3.2.1. Platform

In the training we used windows 10 as an operating system with NVIDIA RTX3070 GPU (graphics card) and Ryzen 7 5800H (processer) and python 3.7 as programming language. The YOLOv4 model has been run under the Darknet framework.

3.2.2. Training

The model was trained on a data set collected by the author, but it did not give good results because of the large size of the images, which led to training problems, so we modified the size of the images and retrained the model on the new data set more than once.

In the first training, the model was trained for 3000 iterations, and we got good results as shown in figure (6) below.

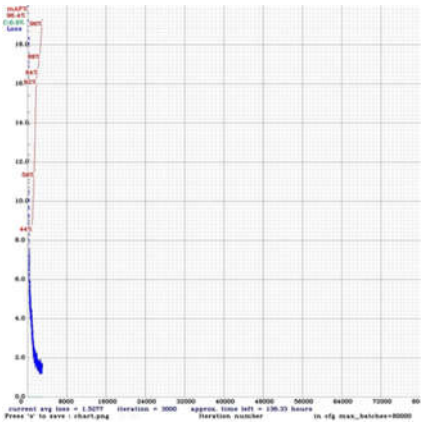

Fig. 6. First training results

As we can see in the figure (6) that the accuracy started to rise, so at iteration 1000 the accuracy was 44%, and rise to 96% at iteration 3000 and the rate of loss started decreasing at iteration 1000 also down to iteration 3000 where it was 1.52 and in this iteration stopped decreasing and the first training ended here.

After that, we re-trained for 4000 iterations, and we got an increase in accuracy and a lower average loss, as shown in figure 7 below.

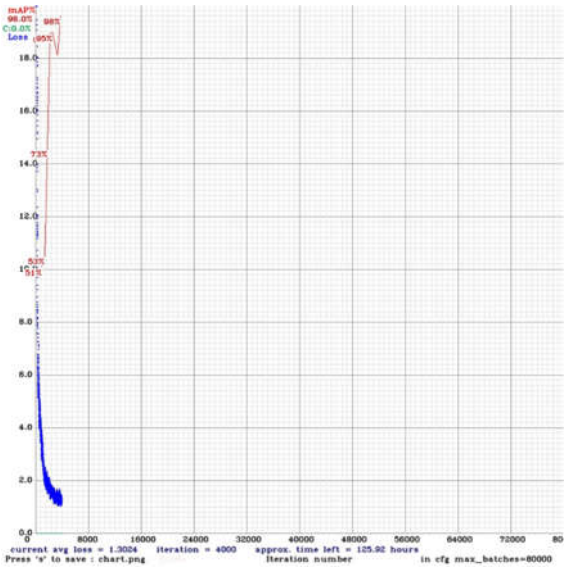

Fig. 7. Second training results

And as we can see in the figure (7) that the accuracy started to rise, so at iteration 1000 the accuracy was 51%, and rise up to 98% at iteration 4000 and the rate of loss started decreasing at iteration 1000 also down to iteration 4000 where it was 1.3 and in this iteration stopped decreasing and the second training ended here.

The increase in iterations does not mean getting better results. The number of iterations has been increased, but we did not get good results and Training has been stopped when the current average loss doesn't decrease anymore.

3.3 Results

The latest results for each class of the data set can be summarized according to the accuracy and the percentage of true and false in precision in the table (3) below.

Table 3. Classes Results

Class name	Average Precision (ap)	True positive (tp)	False positive(fp)
1	100.00%	40	0
2,V	99.10%	39	2
3	100.00%	40	0
4	83.88%	18	4
5,Hello	99.62%	39	3
6,W	97.78%	39	5
7	100.00%	40	0
8	100.00%	40	0
9	100.00%	40	0
Zero ,O	100.00%	40	0
A	99.88%	40	1
B	100.00%	40	0
C	100.00%	40	0
D	100.00%	40	0
E	100.00%	39	1
F	100.00%	40	4
G	100.00%	40	0
H	100.00%	40	0
I	100.00%	40	0
J	100.00%	40	1
K	100.00%	40	0
L	100.00%	40	0
M	86.87%	35	7
N	100.00%	40	0
P	100.00%	40	1
Q	96.00%	39	1
R	100.00%	40	0
S	100.00%	40	0
T	99.40%	40	6
U	100.00%	40	5
X	92.86%	20	1
Y	100.00%	40	4
Z	100.00%	40	0
Yes	100.00%	40	0
No	100.00%	40	2
Please	76.03%	25	8
Thanks	100.00%	40	0
Ok	100.00%	40	0
Sorry	89.10%	40	11
I Love You	100.00%	40	0

3.4 Examples of images testing results:

Hello

9

G

I Love You

V,2

8

And we can summarize the results in both trainings in the table (4) below

Table 4. Last Trainings Results

Results	First Training	Second Training
Precision	0.88	0.96
Recall	0.94	0.96
F1-Score	0.91	0.96
TP	1510	1533
FP	215	67
FN	90	67
Average IoU	69.86%	76.67%
mAp	96.44%	98.01%

Detection Time	41 seconds	42 seconds

And for video testing results in first training we got 28.3 frame per second (fps) and in second training we got 28.9 frame per second (fps).

IV. CONCLUSION

In this research, a recognition system for The American Sign Language (ASL) using YOLOv4 method has been done. Also we introduced a new dataset of American Sign Language. The experiment on images got 98% MAP as an accuracy which is very good and the current average loss is 1.53 and for video testing results we got 28.9 fps. In the future, we will work to increase the dataset by adding new words and signs. We will also improve the accuracy of the images in the data set to get better results.

REFERENCES

[1] Z. Zafrulla, H. Brashear, P. Yin, P. Presti, T. Starner, and H. Hamilton, "American sign language phrase verification in an educational game for deaf children," in 2010 20th International Conference on Pattern Recognition, 2010, pp. 3846–3849.

[2] C. Oz and M. C. Leu, "American sign language word recognition with a sensory glove using artificial neural networks," Eng. Appl. Artif. Intell., vol. 24, no. 7, pp. 1204–1213, 2011.

[3] S. Lang, M. Block, and R. Rojas, "Sign language recognition using kinect," in International Conference on Artificial Intelligence and Soft Computing, 2012, pp. 394–402.

[4] D. Aryanie and Y. Heryadi, "American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier," in 2015 3rd International Conference on Information and Communication Technology (ICoICT), 2015, pp. 533–536.

[5] R. A. Kadhim and M. Khamees, "A real-time american sign language recognition system using convolutional neural network for real datasets," Tem J., vol. 9, no. 3, p. 937, 2020.

[6] 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, 2016, pp. 779–788.

[7] J. Du, "Understanding of object detection based on CNN family and YOLO," in Journal of Physics: Conference Series, 2018, vol. 1004, no. 1, p. 12029.

[8] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian Sign Language Recognition using YOLO Method," in IOP Conference Series: Materials Science and Engineering, 2021, vol. 1077, no. 1, p. 12029.

[9] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video," arXiv Prepr. arXiv1709.05943, 2017.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv Prepr. arXiv2004.10934, 2020.

[11] J. Yu and W. Zhang, "Face mask wearing detection algorithm based on improved YOLO-v4," Sensors, vol. 21, no. 9, p. 3263, 2021.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[13] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Adv. Neural Inf. Process. Syst., vol. 28, 2015.

[15] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3339–3348.

[16] Z. He and L. Zhang, "Multi-adversarial faster-rcnn for unrestricted object detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6668–6677.

[17] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," IEEE Trans. Multimed., vol. 20, no. 4, pp. 985–996, 2017.

[18] X. Wang, H. Ma, and X. Chen, "Salient object detection via fast R-CNN and low-level cues," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 1042–1046.

[19] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2606–2615.

[20] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented r-cnn for object detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3520–3529.

[21] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.

[22] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-CNN for small object detection," in Asian conference on computer vision, 2016, pp. 214–230.

[23] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V.-K. Papastathis, and M. G. Strintzis, "Knowledge-assisted semantic video object detection," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 10, pp. 1210–1224, 2005.

[24] L. Guan, Multimedia image and video processing. CRC press, 2017.

[25] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," in 2007 IEEE 11th international conference on computer vision, 2007, pp. 1–8.