Neural Computing and Applications (2024) 36:7609–7624
https://doi.org/10.1007/s00521-024-09503-6

ORIGINAL ARTICLE

Real-time sign language recognition based on YOLO algorithm

Melek Alaftekin[1] · Ishak Pacal[2] · Kenan Cicek[1]

Received: 7 March 2023 / Accepted: 14 January 2024 / Published online: 15 February 2024
© The Author(s) 2024

Abstract

This study focuses on real-time hand gesture recognition in the Turkish sign language detection system. YOLOv4-CSP based on convolutional neural network (CNN), a state-of-the-art object detection algorithm, is used to provide real-time and high-performance detection. The YOLOv4-CSP algorithm is created by adding CSPNet to the neck of the original YOLOv4 to improve network performance. A new object detection model has been proposed by optimizing the YOLOv4-CSP algorithm in order to provide more efficient detection in Turkish sign language. The model uses CSPNet throughout the network to increase the learning ability of the network. However, Proposed YOLOv4-CSP has a learning model with Mish activation function, complete intersection of union (CIoU) loss function and transformer block added. The Proposed YOLOv4-CSP algorithm has faster learning with transfer learning than previous versions. This allows the proposed YOLOv4-CSP algorithm to perform a faster restriction and recognition of static hand signals simultaneously. To evaluate the speed and detection performance of the proposed YOLOv4-CSP model, it is compared with previous YOLO series, which offers real-time detection, as well. YOLOv3, YOLOv3-SPP, YOLOv4-CSP and proposed YOLOv4-CSP models are trained with a labeled dataset consisting of numbers in Turkish Sign language, and their performances on the hand signals recognitions are compared. With the proposed method, 98.95% precision, 98.15% recall, 98.55 F1 score and 99.49% mAP results are obtained in 9.8 ms. The proposed method for detecting numbers in Turkish sign language outperforms other algorithms with both real-time performance and accurate hand sign prediction, regardless of background.

Keywords Turkish sign language · YOLO · CSPNet · Deep learning · CNN · Sign language recognition

1 Introduction

Sign language is a communication tool that deaf and dumb people use to convey their feelings, thoughts, desires etc. to the outside of their world. It has a nonverbal form of communication that uses hands, arm, head, facial gestures and body posture. According to the data of the world health organization in 2021, there are around 466 million people suffering from hearing loss around the world including 34 million children. It is estimated that this number will exceed 700 million by 2050 Word (Health Organization, 2021). Sign language has a key role in communication for hearing-impaired people and the rest of society. Although sign language is not prevalent between normal people, there are very few people who can interpret it. This creates a communication barrier between the deaf-mute and other people, causing them to be unable to express themselves and to experience difficulties in daily life such as communicating with other people, learning at school, shopping etc. In order to create a common communication base between these disabled people and the rest of the world, sign language recognition (SLR) platforms that track and identify signs performed by signers and convert them into meaningful letters, numbers, words and expressions has been developed.

Sign language recognition is based on two fundamental sensing techniques; vision-based and glove-based [1]. In glove-based technique, a number of sensors attached to signers are used to capture the movement, position, direction and speed of the hand and fingers. Researchers have used sensors such as data gloves [2], Microsoft Kinect [3], leap motion controller [4], depth camera [5], surface

✉ Kenan Cicek
eexkc@my.bristol.ac.uk

1 Department of Electrical Electronics Engineering, Engineering Faculty, Igdir University, 76000 Igdir, Turkey

2 Department of Computer Science, Engineering Faculty, Igdir University, 76000 Igdir, Turkey

electromyography (sEMG) signal arm rings [6, 7] and inertial measurement unit (IMU) [6] in their studies. Although these studies achieved excellent results, it is generally expensive and not practical for daily life usage. On the other hand, in case of vision-based techniques, sign recognition is performed employing images of hand movements taken with the help of a single camera. Not only the vision-based technique is easier but also it has relatively lower computational cost. For the vision-based technique, image processing or/and signal processing is performed to identify the relevant clue with segmentation, contour and boundary modeling, color and motion and feature vector of the image. Afterward, obtained data are compared with the stored labeled dataset for final interpretation. During these operations, certain probabilities are calculated and depending on the probability the output is classified in the corresponding class.

In recent years, deep learning that is a subset of machine learning has become popular in vision-based SLR for a robust and real-time operation [8]. With its high performance in image classification, object detection, image acquisition, semantic segmentation and human pose prediction [9], deep learning achieved great importance and reliability in sign language studies. In addition, the processability of large datasets, which is highly required in image processing, is another advantage of deep learning techniques [10]. The main point of deep learning in vision-based SLR is to make an accurate classification by comparing the relevant hand model with the whole image captured, and process it to find the relevancy in between [11].

From the literature, the comparison of deep learning and traditional computer vision models in sign language shows a tendency for deep learning models to achieve more effective and accurate results. Traditional computer vision models typically require long training processes to manually design and learn acquired features, whereas deep learning models can operate more quickly and efficiently due to their automatic feature extraction and learning capabilities. Recently, the number of studies that utilizing you only look once (YOLO) CNN method in sign language has been rapidly increasing [12–17]. It is shown that YOLO has high impact on accuracy and performance of computer vision detection [13, 17].

In this study, optimizing YOLOv4-CSP [18], a new method is proposed to recognize the relevant gesture without any pre-image processing or the need for a uniform background. In the proposed method, the numbers of Turkish sign language, which is the first study using Turkish sign language in such a method in the literature according to our knowledge, is used for the validation of the proposed method. In this method, real-time object detection is performed by detecting the hand movements of the users from a video without the need for an external mechanism.

2 Materials and methods

2.1 Overall approach

In this section, the detailed structure of the proposed method is explained. In the first step, a dataset is created using a professional camera, outlining the approach followed by the suggested model presented visually in Fig. 1. Prior to commencing the model's training, a manual classification process is undertaken. As a result of this process, the emerged dataset consists of 1500 labeled images and is divided into random training, testing and validation stages at specified ratios. Images for each class are allocated as 70% for training, 15% for validation and 15% for testing. Data augmentation involves various techniques to alter the size and quality of training datasets. During training, augmentation techniques such as contrast adjustment, rotation, mirroring and brightness are utilized to increase data diversity. Due to the limited size of the Turkish sign language dataset for standalone training, different data augmentation techniques are applied to enhance the models' generalization ability on test data. With these two methods, both data size and diversity are increased.

Yolov4-CSP is an object detection model based on the scaling technique [18]. It can be scaled as both small and large models, with the smallest size and the least number of parameters. Typically employed for swiftly detecting objects at lower resolutions, it excels in speed. On the other end of the spectrum, the largest model boasts the highest number of parameters and is utilized for detecting more complex and higher-resolution objects. While this model often delivers higher accuracy, it demands more computational power.

2.2 YOLO series

Current object detection algorithms generally have two methods: one-stage and two-stage. The general architecture of object detection algorithms is shown in Fig. 2. In two-stage object detection algorithms, a region recommendation network (RPN) is used to extract target object information from an image and object detection is performed by making a classification for each proposed region, while in single-stage object detection algorithms, a full regression problem is used, instead of suggesting the relevant regions separately. Thus, an object detection operation is performed which obtains the bounding box and object class directly from the image.

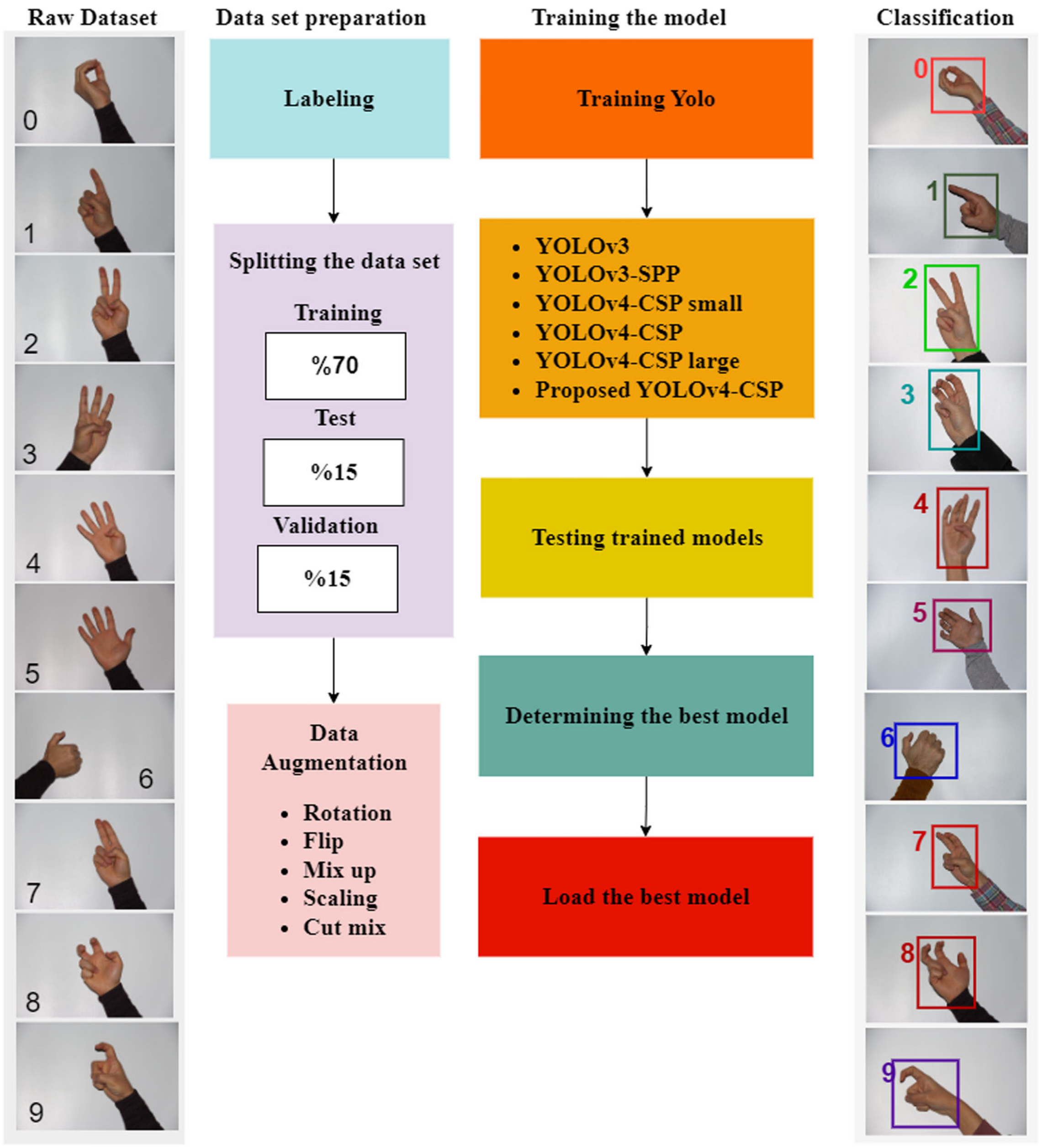


Fig. 1 System flow diagram

Although two-stage object detection algorithms such as SPP-NET [19], Fast-RCNN [20], Faster-RCNN [21] perform better in terms of detection and localization sensitivity, these algorithms could not allow real-time object detection. As the single-stage object detection algorithm, YOLO series algorithm is proposed by a group in 2016 [22] which is also the first regression-based object detection algorithm.

YOLO offers end-to-end real-time detection by processing at 45 fps with a simple regression analysis model [22]. Although YOLO has opened the door to achieve real-time object detection, it has the error of detecting and localizing small-sized objects. A threshold value with a confidence score is defined for object prediction in the YOLO algorithm. Bounding boxes that have a value below the threshold value on the image are eliminated with the non-maximal suppression technique. Thus, the most appropriate bounding box for detection is found and a loss function value is calculated. The detection diagram of YOLO is shown in Fig. 3.

In 2017, YOLOv2 [23], an improved version of YOLO, is released. The main purpose of YOLOv2 is to make improvements in recall and localization while maintaining classification accuracy. The YOLOv2 architecture has a

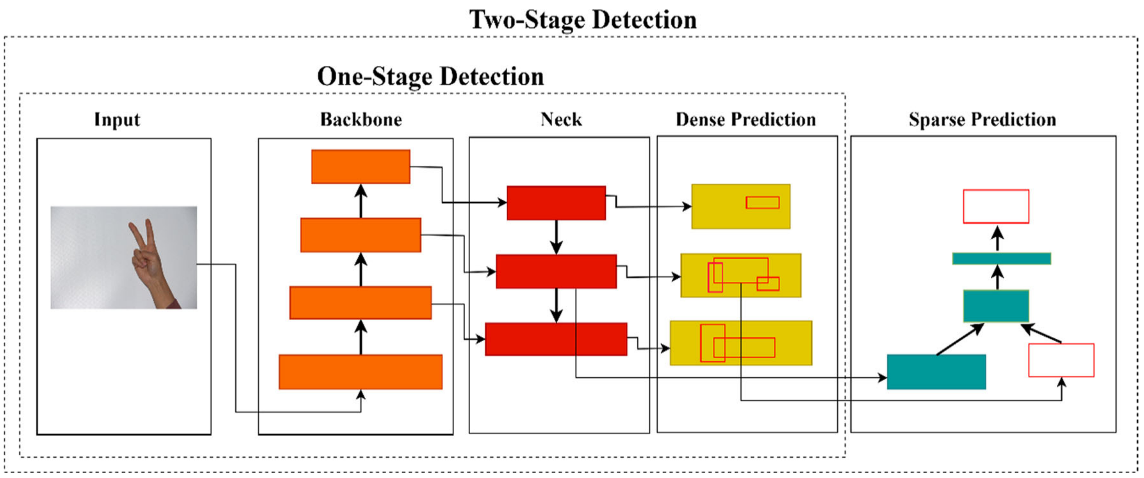
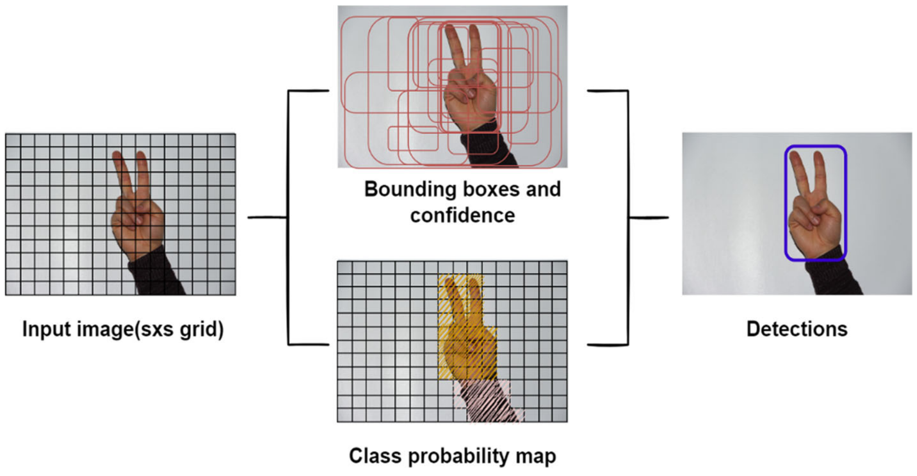


Fig. 2 General architecture of object detection algorithm. Inspired by He et al. [19]

Fig. 3 The detection process diagram in YOLO algorithms



Darknet19 backbone structure consisting of 19 convolutional layers. It has been observed that this backbone structure caused limitations in object detection accuracy in the studies by researchers. To overcome this limitation, YOLOv3 [24] with the backbone structure Darknet53, which is a version of the Darknet architecture in the backbone structure, is released in 2018. One of the key features that distinguishes the YOLOv3 architecture from previous versions is the Darknet53 backbone with residual modules and short-path connections. It also possesses a multi-scale prediction FPN to detect objects of different sizes, especially small objects, within an image. These two structures have made Yolov3 superior to previous versions in terms of both speed and performance. In addition, the YOLOv3 architecture has three object detection heads to capture information of objects at both low and high level, which output rich feature maps. However, using logistic regression in YOLOv3, object confidence score and class estimation are made for bounding boxes that calculate different aspect ratios. YOLOv4 [25] is proposed in 2020, with continuous improvement of YOLO versions to further improve the object detection ability. The YOLOv4 algorithm is released with increased object detection speed, parallel computing power and multiple label classification capability compared to previous versions.

2.3 YOLOv4 algorithm

The YOLOv4 algorithm is proposed by Bochkovskiy et al. [25] with further optimization and improvement of YOLOv3. The structure of YOLOv4, unlike other YOLO series, consists of a backbone used for feature extraction, a head that performs object classification and bounding box estimation, and neck structures that contain additional path connections that provide rich semantic features by collecting feature maps. The YOLOv4 architecture is given in Fig. 4.

In the YOLOv4 architecture, the authors have used the CSPDarkNet53 backbone by adding CSPNet [26] to the backbone structure to make the DarkNet53 [24] neural

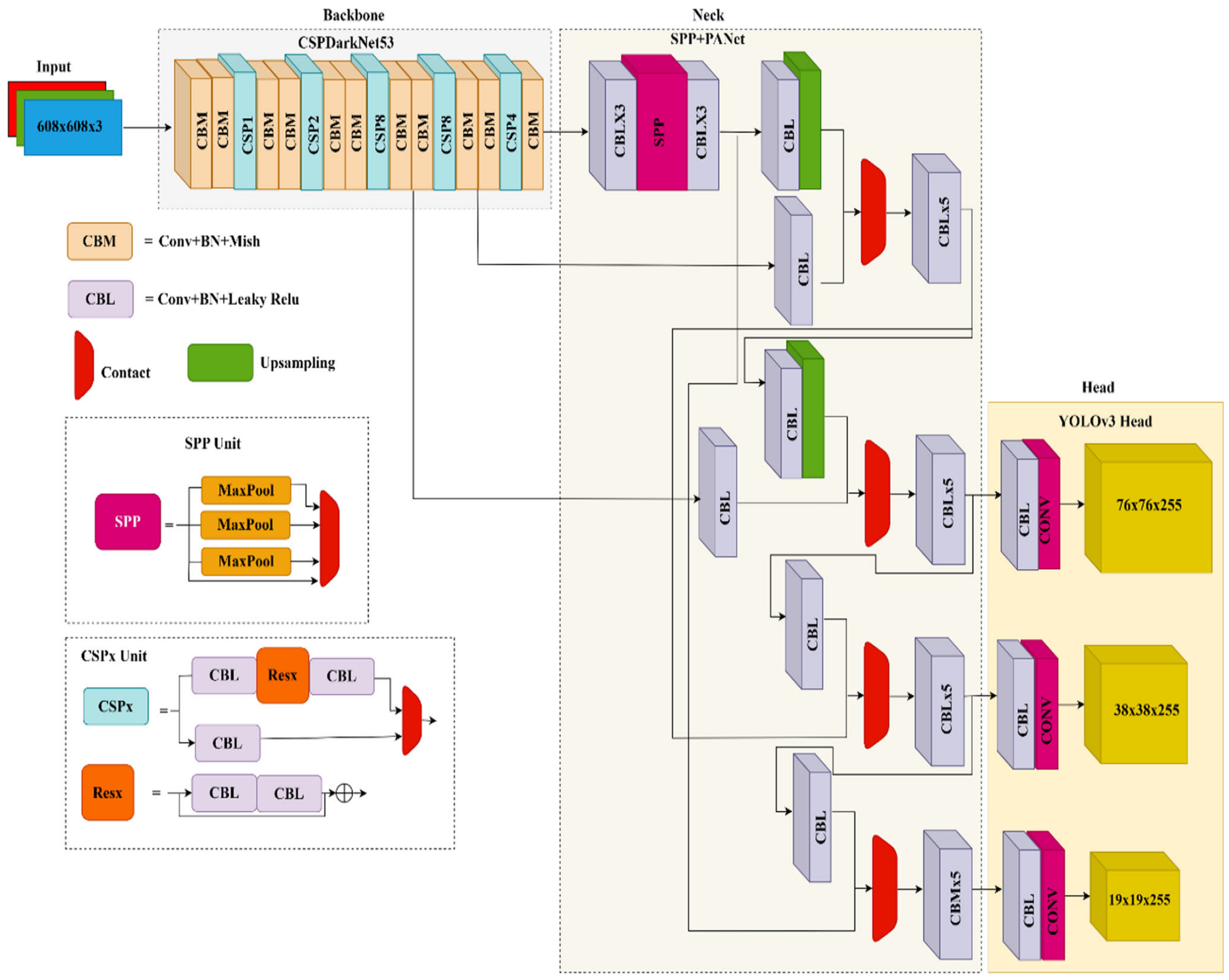


Fig. 4 General architecture of object detection algorithm. Inspired by Pacal and Karaboga [50]

network more efficient. The main purpose of adding CSPNet to the backbone is to increase the learning capability of the network by reducing gradient problems. CSPDarkNet53 convolutional structure consists of 5 blocks of CSP and 11 CBM modules. Each CSP block consists of convolution layer, batch normalization (BN) [27] and Leakly ReLU [28]. CSP blocks, an innovation in the object detection network, contain residual (Res) units that allow the network to become deeper and extract more features. CSP blocks in YOLOv4 architecture include CSP1, CSP2, CSP8, CSP8 and CSP4 blocks named according to the number of Res units they contain.

In order to strengthen the generalization ability of the YOLOv4 algorithm and to avoid the overfitting issue, the Mish activation function [29] is used in the neck part, while the Leakly ReLU activation function is used in the rest of the network. The spatial pyramid pooling (SPP) module used in the neck of the network improves the receptive field size of the network by maintaining the spatial dimension of the network. In the SPP module, a maximum pooling is implemented with sliding windows of sizes 1 × 1, 5 × 5, 9 × 9, 13 × 13. The SPP module connected to the last convolution layer increases the accuracy of the network without any noticeable change in network operating speed. In the YOLOv4 algorithm, modified path aggregation network (PAN) is used to ensure localization. Modified PAN brings features from lower layers to upper layers with the help of combined shortcuts. Thus, it made it easier to gather the information from different layers of the network. Finally, a modified spatial attention module (SAM) is used in the neck of the network. Normally, maximum and average pooling is applied to the input feature maps separately to create two feature maps in a SAM module. In YOLOv4, however, these pooling layers are removed and replaced with a modified SAM module with a convolution layer.

The YOLOv4 architecture has a more complex structure than the previous series. In order to improve the accuracy of the neural network, many object detection techniques have been tested and the most appropriate optimization has been tried to be applied. Bochkovskiy et al. have proposed two methods that include a set of specific modules. These are the Bag of Freebies (BoF) and Bag of Specials (BoS) methods. Bag of Freebies (BoF) offers different training strategies to the model to increase the detection accuracy of the model or simply increase the training cost and Bag of Specials (BoS) also includes a number of methods that significantly increase object detection accuracy with less computational cost. The contents of these methods may differ depending on the detectors designed for different purposes. This allows researchers to create many new object detection algorithms that can be used for different purposes, providing a wide range of studies. BoF and BoS are used both the backbone and detector part (head + neck) to significantly increase the efficiency of YOLOv4. The contents of the BoF and BoS modules used in YOLOv4 are given in Table 1.

Offline training of object detection detectors and obtaining more accurate results from the detector without affecting the cost of inference during the training is a preferred training methodology by researchers. Therefore, the BoF module offers a suitable training strategy in the backbone with techniques such as DropBlock regularization and class smoothing, CutMix and Mosaic data augmentation. In addition, CIoU loss for the sensitivity of the neural network, CmBN for a more accurate prediction by batch normalizing the training images, DropBlock regularization, Mosaic data augmentation and SAT to prevent overfitting, Cosine annealing scheduler, Optimal hyperparameters and random training shapes used in the head and neck. The BoS module includes many add-on modules that significantly increase the sensitivity of the neural network. These add-on modules include SPP to expand the receptive field of the network, SAM for the attention mechanism, PAN to collect feature maps at different layers, Mish activation function to increase the sensitivity of the network and DIoU-nms to make the true bounding box most dominant in the final processing stage. The YOLOv4 algorithm is an object detection network that is easy to optimize due to its structure. As shown in the original YOLOv4 article, different combinations of the most suitable algorithms from BoF and BoS methods have been tried to be created to obtain better results than YOLOv4.

The head part of the YOLOv4 algorithm is the part that uses the extracted features to give the final result. Three YOLO heads are adopted in the detector part of the YOLOv3. The inputs of the YOLOv4 detector heads consist of rich spatial information from the neck. The main function here, is to find and classify the best bounding box. Feature maps of 19×19, 38×38 and 76×76 dimensions are created separately in the detector. These heads, used to find objects of different sizes, include the target object's coordinates [x, y, width (w) and height (h)], a confidence score and a class prediction score.

2.4 Proposed method

CNN-based object detection algorithms are mostly designed for different suggestion systems. For researchers, efficiency is a major part of this. In the EfficientDet [30] article, it is proposed to increase the efficiency of object detection algorithms and systematically scale the network which is one of the key points of CNN. Scaling is a common technique in CNN architectures such as ResNet [31], ResNeXt [32] and DarkNet [33]. Scaling in neural networks is usually done by changing the number of convolution layers (depth), the number of convolution filters (width) or the sizes of the input images. The scaling of the object detection network not only increases the accuracy of the network, but also increases the number of network

Table 1 Summary of BoF and BoS used in YOLOv4

Architecture of YOLOv4	BoF used in training	BoS used for inference
Backbone	● CutMix and Mosaic data augmentation	● Mish activation
	● DropBlock regularization	● Cross-stage partial connections (CSP)
	● Class label smoothing	● Multi-input weighted residual connections (MiWRC)
Detector (Neck + Head)	● CIoU loss	● Mish activation
	● CmBN	● SPP-block
	● DropBlock regularization	● SAM-block
	● Mosaic data augmentation	● PAN path aggregation block
	● Self-Adversarial Training (SAT)	● DIoU-nms
	● Cosine annealing scheduler	
	● Optimal hyperparameters	

parameters, leading to an increase in the high computational cost. Although the increase in computational cost makes the object detection network a better detector, it requires more powerful hardware devices for real-time applications. In the literature, scaled YOLOv4 has been proposed by improving the backbone of YOLOv4 to overcome this problem and provide deepening of the neural network [18]. The scaled YOLOv4 algorithm is based on the up and down scaling of the YOLOv4 algorithm with CSPNet. The use of CSPNet architecture both reduces the cost of calculation and increases the accuracy and speed of estimation [34]. In addition, the use of CSPNet for scaling in CNN-based networks effectively reduces computation. As it can be seen in Figs. 4 and 5, CSPNet is used in the neck part of YOLOv4, as it is used throughout the network of YOLOv4-CSP.

We use a system based on scaled YOLOv4 to offer a real-time sign language detection model. Our main goal is to provide real-time detection by optimally adjusting the scaling parameters in the backbone and neck of the model. We add transformers [35] to the scaled YOLOv4 architecture to improve the performance of our proposed detection model. An important study on transformer architecture is recommended by Vaswani et al. Transformer, which consists of an encoder-decoder architecture, is based on a system that works entirely with an attention mechanism instead of a traditional recurrent neural network to increase the training speed of the network. Transformers are preferred in object detection algorithms to distinguish foreground and background objects in the encoder part of the model and to make position and class estimation for these objects [36]. While the transformer is first created to solve natural language processing problems, it has recently been used effectively in the field of computer vision. The most widely used transformer model in the field of computer vision is the vision transformer. The vision transformer (ViT) has achieved remarkable success in the literature in computer vision compared to cutting-edge CNNs, requiring fewer computational resources for preliminary training [37]. ViT processes the image by putting it in patches, while CNN processes it in pixels. In addition, ViT offers a hybrid model use for CNNs to improve the performance of the object detection model.

The YOLOv4 architecture we propose exhibits significant alterations, incorporating the CSPNet network into the YOLOv4 structure and thereby reoptimizing the recommended architecture in the form of YOLOV4-CSP. By integrating the CSPNet, we have addressed the gradient vanishing issue that arises with the deepening of the network, resulting in a configuration that enables more effective gradient propagation. As the network's model deepens, CSPNet allows for the management of complexity to reduce computational costs, rendering it applicable to any neural network with simplicity. As stated in the original Scaled-YOLOv4 paper [17], CSPNet [25], with its additional residual layers at each stage, significantly diminishes the computational expense in the backbone, thereby enhancing interactions across different layers of the network and facilitating the extraction of distinct feature maps. The neck of a network aggregates features at various scales and levels to obtain enriched feature maps. In the proposed model's bottleneck, we employ the existing PANet [38] architecture alongside CSPNet. This effectively reduces the computational cost of generating rich feature maps. Finally, by integrating CSP into the SPP module, CSPPAN's initial calculation set has been augmented at the central position. Consequently, this enables a more effective incorporation of the CSPNet architecture in both the backbone and bottleneck sections of the network.

In this study, we designed to require less FLOPs by adding a hybrid structure to the neural network that is similar to the ViT structure. This made a positive contribution to reducing both inference and training times. In order to increase the detection capability of the object

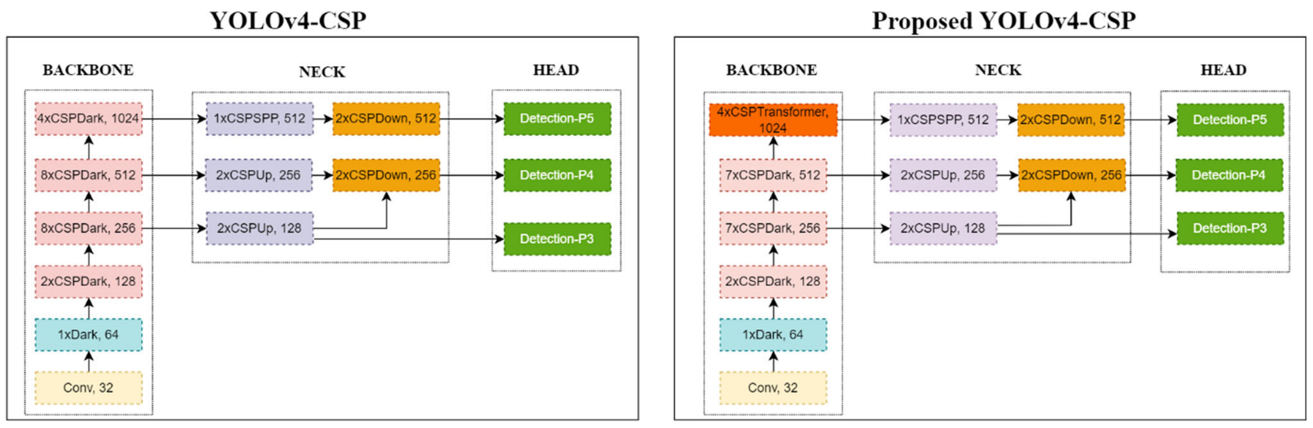


Fig. 5 Architecture of YOLOv4-CSP and the proposed YOLOv4-CSP

detection model, especially the detection of large objects, the transformer block has been added to the last block, P5 (1024 channels), located in the backbone of the neural network. It has also been observed that by adding the transformer block to other parts of the backbone, more power is required for calculation and not efficient in object detection. In our experiments, the last CSPNet in the YOLOv4-CSP backbone is combined with the transformer. Next, we used all blocks, including the first Residual block, with CSPNet. However, the CSPNet that we added to the first block increases the learning ability of the network, but also causes a minor increase in computational cost. The detection model that is proposed for the Turkish sign language detection system is shown in Fig. 5.

In order to make the proposed method more specific, the number of filters and blocks of YOLOv4-CSP are reduced. By reducing the number of 1xDark, 2xCSPDark, 8xCSPDark, 8xCSPDark, 4xCSPDark, CSPNet blocks and filter numbers, which are the backbone of the existing structure, it has been transformed into 1xDark, 2xCSPDark, 7xCSPDark, 7xCSPDark, 4xCSPTransformer structure and presented a better detection model. In addition, reducing the number of layers and filters in the neck of the network makes the proposed learning model shallower. Thus, we have obtained a sign language recognition model that provides shorter training time and more accurate results. We also compared the performance of YOLOv3, YOLOv3-SPP and YOLOv4-CSP object detection models on sign language.

2.4.1 Activation function

The activation function has an important role in neural networks. The activation functions are used to determine the output of neural networks by applying a mathematical transformation to an input for layers. Activation functions are an important parameter for neural networks to perform better training and have higher classification precision. In addition, the differentiability of the activation functions is an important factor for back propagation (BP).

While the YOLOv4 algorithm is being designed, Mish activation function and Leaky ReLU [28] are utilized in the spine and neck-head parts, respectively. The Leaky ReLU activation function is a popular activation function in deep learning. Leaky ReLU provides a solution for the negative values that cause the dead gradient problem which occurs by taking zero values in ReLU. It multiplies the inputs with negative values by a very small α coefficient (in a range of 0–1) and returns very small negative numbers, and allows a constant slope. Thus, it makes back propagation possible even for negative input. Mathematical equation of the Leaky ReLU activation function [28]:

$$f(x,a) = \{\alpha x \;\text{ for }\; x < 0 x \;\text{ for }\; x \geq 0 \quad (1)$$

Mish activation function is a self-regulating, smooth, continuous and non-monotonous activation function. The mathematical expression of the Mish activation function is as follows [29]:

$$f(x) = x * \tanh(\ln(1 + e^x)) \quad (2)$$

SiLU, which stands for Sigmoid-weighted Linear Unit, is another popular activation function in the field of deep learning. SiLU has been suggested as an activation function for neural network function approximation in reinforcement learning. SiLU activation function is computed by taking the product of the input and the output of the sigmoid function, which can be expressed mathematically as follows [39]:

$$\sigma(x) = \frac{1}{1 - e^{-x}} \quad (3)$$

$$a_k(s) = z_k \sigma(z_k) \quad (4)$$

Here, $a_k(s)$ is activation of SiLU, s input vector of SiLU, z_k is input to hidden units k.

The graphs of Mish, Leaky ReLU and SiLU activation functions are given in Fig. 6. As shown in figure, while all three activation functions tend to approach infinity for positive values, the Mish and SiLU activation functions tend to approach zero for negative values. In addition, the graph of the Leaky ReLU activation function exhibits a segmented linear pattern. While it remains unaltered for positive input values, a gentle slope is introduced for negative inputs to avert the occurrence of dormant neurons. The Mish function smoothly transitions from a linear behavior for extreme inputs to a more saturated behavior for inputs close to zero, whereas SiLU provides a gradual and non-monotonic activation response with its smooth

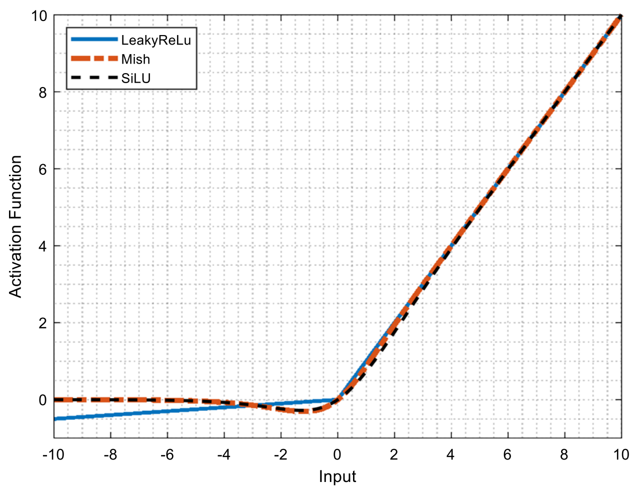


Fig. 6 Response comparison of Mish, Leaky ReLU and SiLU activation function

sigmoid-like curve, effectively addressing the vanishing gradient problem. In Fig. 6, similar behaviors of the Mish and SiLU activation functions are demonstrated. When examining the derivatives of these two activation functions, the derivative of Mish starts close to 1 for negative inputs, gradually approaches zero for inputs near zero, and then steadily increases as the input becomes positive. In contrast, the derivative of SiLU maintains a sigmoid-like curve with values slightly below 1 for both negative and positive inputs, indicating a consistent and controlled gradient throughout. Both derivatives contribute to reducing vanishing gradients, yet the Mish derivative showcases a more nuanced modulation, potentially allowing better adaptability to various input distributions and aiding convergence during neural network training. This makes it more popular for object detection algorithms [40].

In Turkish sign language detection, we proposed a model that uses Mish activation function instead of Leaky ReLU in the entire model in order to make more accurate predictions as well as efficient model. Therefore, Conv + BN + Mish (CBM) combination is employed instead of Conv + BN + Leaky ReLU (CBL) in the entire structure.

2.4.2 Loss function

Loss functions are another important factor affecting the sensitivity of object detection algorithms to detect the target object. In YOLO algorithms, the loss functions are found by summing the total square errors of the classification loss, localization loss and confidence loss values [22]. Many object detection algorithms use the intersection of union (IoU) function for the bounding box prediction. The IoU function is a metric that is used to describe the quantity of overlap between the predicted bounding box and the target bounding box. The IoU is formulated as follows:

$$\mathrm{IoU} = \left|\frac{B \cap B^{\mathrm{gt}}}{B \cup B^{\mathrm{gt}}}\right| \tag{5}$$

$$L_{\mathrm{IoU}} = 1 - \mathrm{IoU} \tag{6}$$

Here, B is the predicted bounding box, B^{gt} is the target bounding box and L_{IoU} is the IoU loss of function.

The IoU function works when the predicted bounding box and the target bounding box overlap, otherwise it cannot make any predictions. Also, the CIoU function found in the BoF module is also recommended for training in YOLOv4 [41]. CIoU loss of function considers the distance and aspect ratios between the center point of the target bounding box and the center point of the predicted bounding box for the bounding box regression. Unlike the IoU loss of function, it introduces an important geometric factor that maintains the consistency of the bounding boxes' aspect ratio. The CIoU loss is formulated as:

$$L_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(b, b^{\mathrm{gt}})}{c^2} + \alpha v \tag{7}$$

$$v = \frac{4}{\pi^2} \tag{8}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{9}$$

Here, b, b^{gt} are center of B and B^{gt}, respectively, ρ (.) is Euclidean distance, c is the diagonal length of the smallest closed box covering the two boxes, α is a positive displacement parameter and v is the parameter that measures the consistency of the aspect ratio. We used the CIoU loss function as the loss function in the sign language recognition system. The CIoU is outperformed IoU by providing geometric factors such as aspect ratio for bounding box regression.

2.5 Proposed dataset

In order to examine the training and evaluation processes of our proposed approach, we used a dataset of images of hand gestures representing figures from 0 to 9 specific to the Turkish sign language. Sign language has its own vocabulary and meaningful movements, just like spoken language. Sign languages are not standard or universal. Every spoken language has its own sign language. While creating a dataset in Turkish sign language, which has a unique grammar and hand sign, we aimed to create a unique dataset under the provision of an expert. All images are taken from 6 different volunteers with the CANON 700d camera capable of taking 18 megapixel images. Each image in the dataset is used with its original dimensions of 5184 × 3456. Figure 7 shows RGB images of all digit movements in Turkish sign language, which is part of the dataset.

In order to make the experimental dataset more diverse, the hand position and angle are constantly changed and 25 different images are taken from each volunteer for each number, and a Turkish sign language dataset is created. Examples of different hand images of the same number are given in Fig. 8. Thus, a dataset consisting of a total of 1500 images is obtained. While creating the dataset, a background consisting of a uniform color (white) is preferred for each hand sign. It is preferred to have better quality of the input images and to avoid background complexity during training. In addition, using a uniform background also facilitates the manual labeling process for classification.

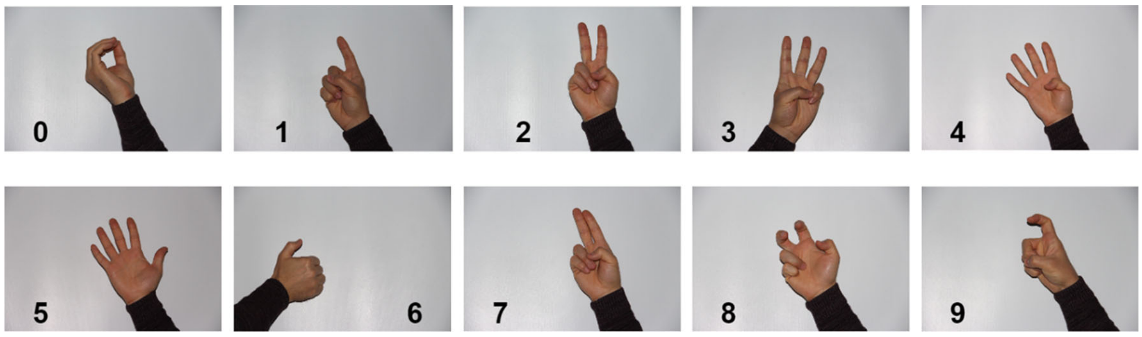


Fig. 7 Turkish sign language numbers

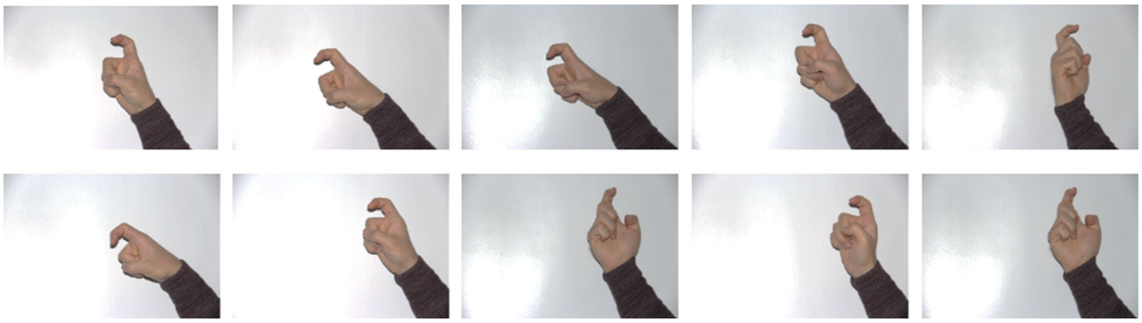


Fig. 8 Hand signs of the number 9

Prior to commencing the training of the recommended model, a manual classification process was conducted. In this study, LabelImg software (Windows version), a graphical image annotation tool, available at https://github.com/tzutalin/labelImg, was employed for the labeling process in the Turkish sign language dataset. LabelImg facilitates the bounding box annotation of labeled objects in images and saves label data for each image in a.txt file. Visually, the data labeling process is presented in Fig. 9.

3 Results and discussion

3.1 Experimental design

All experiments in this study are conducted with a high-power computer running Ubuntu 20.04 and a Linux operating system. This computer, used to train and test deep learning models, features the following: Intel Core i9 9900X (10 cores 3.50 GHz, 19.25 MB Intel® Smart Cache) processor, 32 GB DDR4 RAM and single RTX 2080TI ((11 GB GDDR6) 4352 cuda core) graphics card, NVIDIA CUDA Toolkit 11.1 and NVIDIA GPU-Accelerated Library (cuDNN) 8.1 with the latest stable version of PyTorch and DarkNet frameworks.

Adam optimizer [42] is used for iteration parameter updates during the training process while designing the model. Adam optimization is an algorithm used for variable gradient descents to train deep learning models. Adam is chosen because of its fast computation time, less parameter requirement for proper optimization and less memory requirement. In the study, momentum is 0.937, weight reduction is 0.0005 and other parameters are used in the same way as the values specified in the original YOLOv4 reference article [25].

3.2 Transfer learning and data augmentation

3.2.1 Transfer learning

Training a neural network from scratch puts at a disadvantage the neural network in terms of both speed and performance. Transfer learning is a machine learning method that is used to apply the acquired knowledge to a different field or related task using pre-trained weights for a task. With transfer learning, a detection model that has a higher performance and learns faster can be obtained with less labeled data by using previously acquired knowledge. This provides a greater advantage than retraining a network. In addition, transfer learning has a fine-tuning operation that allows very fine-tuning of network parameters. Transfer learning plays an important role in deep learning models, helping to learn with less data and knowledge. Because the dataset is small, transfer learning is utilized for a more efficient learning process of deep learning-based Turkish sign language detection system.

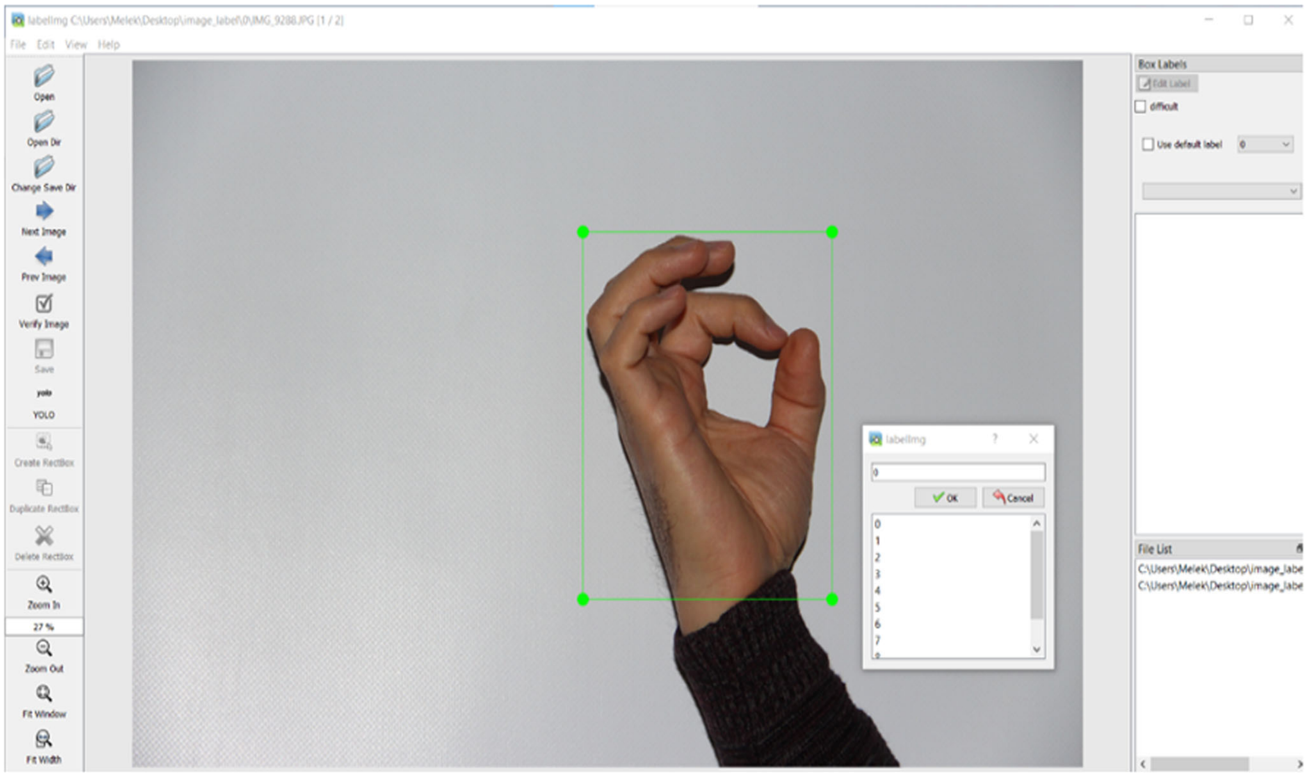


Fig. 9 Labeling data in LabelImg

3.2.2 Data augmentation

In deep neural networks, the size of the dataset is an important factor for the network to perform properly by preventing overfitting. Therefore, data augmentation techniques are crucial in object detection algorithms. Data augmentation includes a series of techniques that change both the size and quality of training datasets. Since the Turkish sign language dataset alone is insufficient for training, different data augmentation techniques are used to increase the generalization ability of the model on the test data. Mosaic4 data augmentation techniques, which is one of the important data augmentation techniques and also included in the original YOLOv4 algorithm, is used during the training. Mozaik4 data augmentation technique combines four original input images in certain proportions or one by one into a single image. In this technique, the four corners of the image are selected separately and then the data augmentation process is applied to these corners. The desired hand sign may not be found exactly in the resulting image.

This allows the searching hand signal to be perceived outside the normal contexts by localizing different images in different parts of the resulting frame. In addition, the learning ability of the network is examined by using data augmentation techniques such as CutMix, contrast adjustment, histogram synchronization, rotation, mirror and brightness in order to increase variety in the dataset during the training. Some data augmentation techniques used in our proposed model are given in Fig. 10.

Data augmentation includes a set of techniques that change the size and quality of training datasets for an effective deep learning model. All these data augmentation techniques have been chosen by considering factors such as different positions, angles and light of hand movements during a possible conversation. The most important feature of data augmentation technique is that each image is used once during the training which reduces the training time of the network. In addition, the images in the dataset are randomly selected, which improves the predictive ability of the object detection network.

3.3 Performance metrics

In YOLO algorithms, object detection is done by dividing the input image into $S \times S$ grids and generating a confidence score that states the probability of finding the target object in these produced grids. In sign language recognition, the hand signals of the users are in the target object position. The confidence score indicates how accurately hand signals are perceived in the bounding box. In the real-time Turkish sign language detection system, the successful detection of the hand signal is determined by a value of IoU. The value of the IoU function is taken a threshold

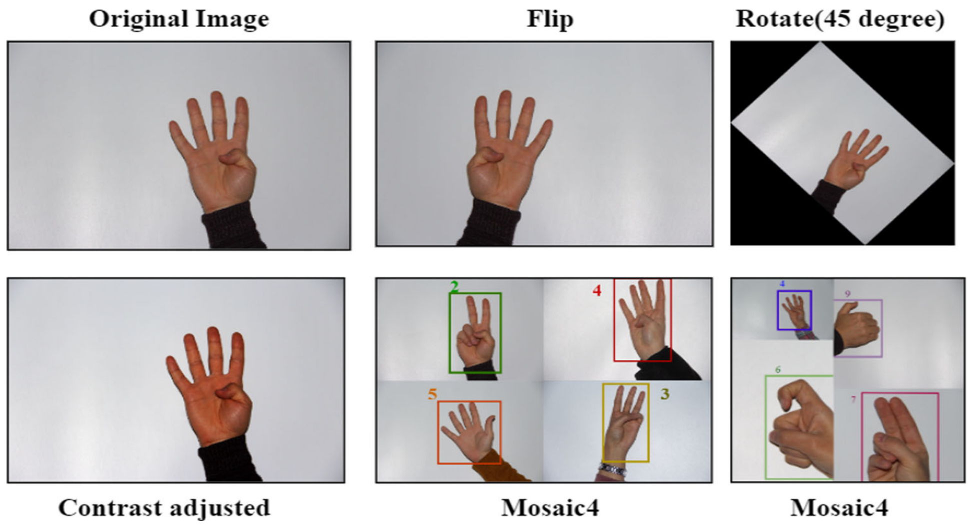


Fig. 10 Data augmentation techniques applied on Turkish sign language

value determined by looking at the overlap between the predicted and the base true bounding box. For this study, the IoU threshold is set to 0.5. If the overlap between the predicted box and the base true bounding box is higher than 0.5, the hand signal predicted by the trained model is considered correct.

Many metrics can be used to measure the performance of deep learning algorithms. In addition to the IoU function, true positive (TP), false positive (FP), false negative (FN) and true negative (TN) metric values are also used in this study. The true positive (TP) indicates that the hand signal of the predicted bounding box is inside the actual bounding box and the model has correctly detected the hand signal. The false positive (FP) indicates that the predicted bounding box is outside the actual bounding box of the hand signal. False negative (FN) is the opposite of false positive (FP) indicating that there is no predicted bounding box even though there is a hand sign in the actual bounding box. The true negative (TN) indicates that the marking on images without hand signals has not been detected. This metric is not considered in our study since, there are no images without hand signs in the dataset. In order to evaluate the performance of the model trained with all these metric values, the system's precision, recall, F1 score value and mAP values are taken into account.

Precision: The ratio of correctly predicted positive results (TP) to all predicted positive results (TP + FP). Precision value formula:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

Recall: The proportion of correctly predicted positive results. Recall also measures the precision of the model. Recall formula:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$F1$ score value: Calculates the harmonic weight of the precision and recall values. $F1$ score formula:

$$F1 = 2\frac{\mathrm{Precision} \times \mathrm{Recall}}{Precision + Recall} \tag{12}$$

mAP value: The average precision (AP) value is a popular metric for object detection detectors. The AP value calculates an average value for the recall value from 0 to 1. The mAP value is a value that gives the average of all AP values as a single number. mAP value formula:

$$\mathrm{mAP} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP}_i \tag{13}$$

The inference rate milliseconds (ms) or fps is the most commonly used measurement in real-time applications. fps is the number of images that can be processed per second. Low extraction rate or high fps is required for real-time applications.

3.4 Results

It is known that the activation process is a crucial part of the neural network that affects the performance. Therefore, we begin with the evaluation of the Mish function by comparing it with the other two common activation functions of SiLU and Leaky ReLU. The proposed model is trained by all these three functions and results are presented in Table 2. As can be seen from the table, the performance parameters (Precision, Recall, F1 Score and mAP) of the proposed model, in which the Mish activation function is used, are higher than the other two functions.

Here, we try to find a real-time Turkish sign language detection system by making structural changes to the

Table 2 Experimental results of various activation functions for the Proposed Model

Activation Function	Precision	Recall	F1 score	mAP
Mish	98.95	98.15	98.55	99.49
Leaky ReLU	97.24	94.96	96.08	98.22
SiLU	98.51	97.03	97.76	99.09

Yolov3 and Yolov4-CSP algorithms. In our proposed model, we use a structure of; CSPDarkNet53 as the backbone, Mish activation as the activation function, CIoU and Transformer as the loss function. The original YOLOv3 and YOLOv4CSP algorithms and their modified structures are trained with a dataset of 1500 images. The dataset is trained in two ways. Firstly, the dataset is originally trained from scratch for YOLOv3, YOLOv3-spp and YOLOv4-CSP. Secondly, the effects on the original dataset are examined by focusing on transfer learning, which is used as a deep learning technique in the training of the proposed YOLOv4-CSP model. With this study, we have optimized the most appropriate parameters in the YOLOV4-CSP algorithm. For the evaluation, training and validation of the proposed models; 70%, 10% and 20% of the dataset of Turkish sign language is utilized, respectively. A threshold value of 0.5 for the IOU, a confidence threshold value of 0.25, and a single graphics card RTX 2080TI is used to evaluate the models.

A detailed review of this entire study is shown in Table 3. The results discussed in the table compares YOLOv3, YOLOv3-SPP and YOLOv4-CSP with proposed YOLOv4-CSP object detection models on Turkish sign language. As it can be seen in Table 3, even though the Proposed YOLOv4-CSP is slower than YOLOv4-CSP and YOLOv4-CSP small, it shows higher results in precession, recall, F1 score and Map which are crucial measures for accuracy of models' prediction. In order to show the predictive performance of the proposed model on the Turkish sign language, a dataset with random samples is shown in Fig. 11. On the left side of the figure, the labels that are assigned for corresponding images are presented and on the right the Model Predictions with confidence scores of these images are shown. The YOLOv4-CSP (transformer) model appears to predict numbers at different positions in the labeled dataset quite successfully. In addition, YOLOv4-CSP (large) presents similar results as our model which includes the transformer block. The YOLOv3 and YOLOv4-CSP (small) models have the lowest performance. While these two models give similar results in terms of accuracy, YOLOv4-CSP is better in terms of speed. Likewise, while YOLOv3-spp and YOLOv4-CSP have similar accuracy results, YOLOv4-CSP also performs faster. The reason why the YOLOv4-CSP and proposed YOLOv4-CSP models are faster than other models is that they both have CSPNet on their necks and heads. Considering the all results in the table, the proposed YOLOv4-CSP model exhibits a higher performance than the others in terms of both accuracy and speed.

3.5 Comparison with other methods

To discuss the novelty of this study, we compare our results with recent studies on sign language recognition systems of both Turkish and some other sign languages. It can be seen from Table 4; many studies in the literature use accuracy as an evaluation criterion. However, the accuracy metric cannot fully demonstrate performance of system compared to other metrics. In object detection, the metric values that show the actual performance of the system are F1 score, giving the harmonic average of precision and recall, and mAP value. Therefore, we evaluate the performance of our system using the F1 score, mAP value and other metrics. Among the studies given in Table 4, [43, 44] perform a preprocessing step for the input images for their proposed sign language recognition system, which has a 94.7% recognition rate and a 98.07% mAP value, respectively. In the model we present here, RGB images are studied without any preprocessing on the input images. This makes the system simple by preventing the process confusion in

Table 3 Models trained from scratch and their results for the Turkish sign language dataset

Model	Precision (%)	Recall (%)	F1 score (%)	mAP (mAP@.5) (%)	Speed (ms)
YOLOv3	96.56	94.45	95.49	97.13	10.3
YOLOv3-SPP	97.23	95.36	96.28	98.27	10.4
YOLOv4-CSP small	96.71	94.70	95.69	97.55	8.5
YOLOv4-CSP	97.53	95.76	96.63	98.67	8.6
YOLOv4-CSP large	98.84	97.27	98.04	99.32	9.8
Proposed YOLOv4-CSP (Transformer)	98.95	98.15	98.55	99.49	9.5

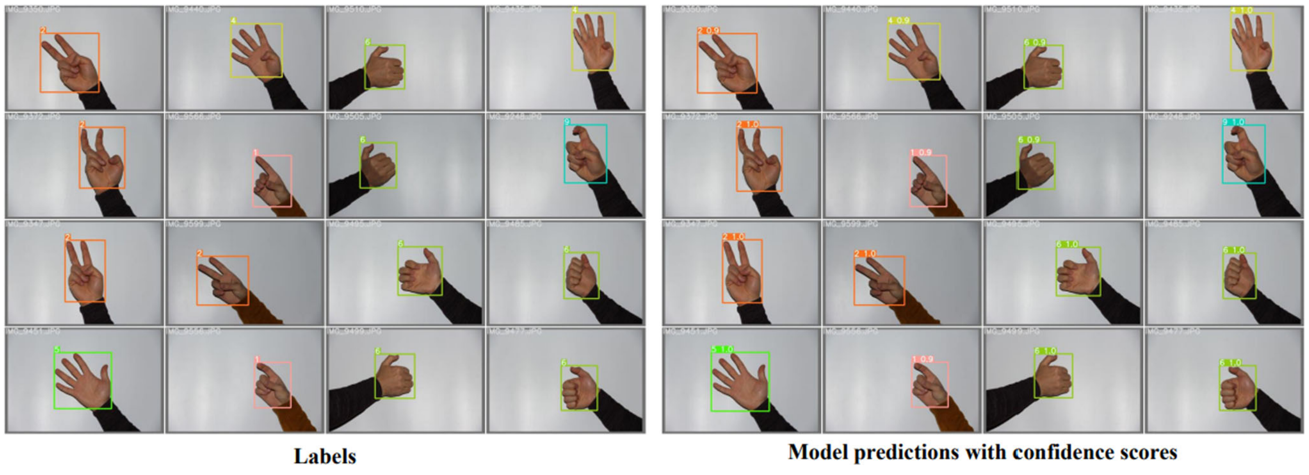


Fig. 11 Figure on the left shows the labeled images of the Turkish sign language dataset, figure on right shows the prediction of the model and the confidence score of the predictions

Table 4 Recent studies on Turkish sign language and other sign languages

References	Sign language	Metrics	Method
Khari et al. [43]	ASL	Recognition rate: 94.8%	Fine-tuned VGG19
Ganggrade and Bharti [45]	ISL	Accuracy: 99.3%	CNN, Microsoft Kinect sensor
Sincan and Keleş [46]	TSL	Accuracy: 96.11%	CNN, LSTM
Rivera-Acosta et al. [44]	ASL	mAP: 81.74% (for letters) Accuracy: 98.07% (for words)	YOLO, LSTM
Özcan et al. [47]	TSL	Success rate: 93.93%	GoogLeNet-based CNN
Khan et al. [48]	MSL	Recognition rates: over 90%	CBAM (Convolution Block Attention Module)
Huang et al. [49]		Recognition rate: 89.90%	CNN-Bi-LSTM
Our method	TSL	Precision: 98.95% Recall: 98.15% F1: 98.55% mAP: 99.49%	Proposed YOLOv4-CSP (with Transformer Block)

order to recognize the interested sign. [45, 46] use a Kinect sensor to capture the interested hand sign from the user in their proposed method.

As a result of their models, they have presented a sign language recognition system with 99.3% and 96.11% accuracy, respectively. Although these models give good results, the need for an external reference during the conversation limits the user. In Turkish sign language [47] and in Malaysian sign language [48] suggested CNN-based methods. In addition, [49] presents a CNN-based recognition system using sensors in their study. The common disadvantage of these studies is that they require two stages: sensor or preprocessing stage and classification for detection and processing of images. However, the proposed YOLOv4-CSP model (with Transformer Block), is performed in a single step and clearly outperforms the other models. It has over 98% values in all metric values. Although we cannot make a real comparison with the studies in the literature, our study is both real-time and has achieved higher results in terms of performance than other studies.

4 Conclusion

In this article, we propose a real-time Turkish sign language recognition system by optimizing the YOLOv4-CSP model that has higher accuracy than the previous YOLO series. To improve the performance of YOLOv4-CSP, we optimize the method by applying the CSPNet network to the entire architecture. We improve the learning of the model by adding the Mish activation function, CIoU loss function and transformer block to the architecture. To evaluate the speed and detection performance of the proposed YOLOv4-CSP model, we compare it with the previous YOLO series, which also offers real-time detection as well. We also evaluate the performance of scratch training and transfer training in object detection. With the proposed method, over 98% values are obtained in all metric values in 9.8 ms. In object detection algorithms, mAP and F1 score values, which show the real performance of the system, are 99.49% and 98.55%, respectively. The YOLOv4-CSP model, which we propose for the real-time Turkish sign language recognition system, detects the static hand sign as an object and performs the process of identifying and classifying the interested region simultaneously. Therefore, YOLOv4-CSP does not have a background complexity problem because it only detects the relevant hand signal. This eliminates the need for preprocessing in the learning process by eliminating the background dependence of selectivity in perception. In addition, the proposed model provides the opportunity to quickly and accurately estimate the hand shape without being tied to the background by making video detection. Therefore, we believe that this study will be a pioneering reference for future work.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). No funding resources available for this research to declare.

Data availability Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest We have no conflicts of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Elakkiya R (2021) Machine learning based sign language recognition: a review and its research frontier. J Ambient Intell Humaniz Comput 12:7205–7224
2. Shukor AZ, Miskon MF, Jamaluddin MH, Bin Ali F, Asyraf MF, Bin Bahar MB (2015) A new data glove approach for Malaysian sign language detection. Proc Comput Sci 76:60–67
3. Ren Z, Yuan J, Meng J, Zhang Z (2013) Robust part-based hand gesture recognition using Kinect. IEEE Trans Multimed 15(5):1110–1120
4. Naglot D, Kulkarni M (2016) Real time sign language recognition using the leap motion controller. In: International conference on inventive computation technologies (ICICT)
5. Almeida SGM, Guimarães FG, Ramírez JA (2014) Feature extraction in Brazilian sign language recognition based on phonological structure and using RGB-D sensors. Expert Syst Appl Int J 14(6):7259–7271
6. Khomami SA, Shamekhi S (2021) Persian sign language recognition using IMU and surface EMG sensors. Measurement 168:108471
7. Tateno S, Liu H, Ou J (2020) Development of sign language motion recognition system for hearing-impaired people using electromyography signal. Sensors 20(20):5807
8. Rastgoo R, Kiania K, Escalerab S (2021) Sign language recognition: a deep survey. Expert Syst Appl 164:113794
9. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48
10. Pacal I, Alaftekin M (2023) Türk İşaret Dilinin Sınıflandırılması için Derin Öğrenme Yaklaşımları. J Inst Sci Technol 13(2):760–777
11. Nimisha K, Jacob A (2020) A brief review of the recent trends in sign language recognition. In: In 2020 international conference on communication and signal processing (ICCSP)
12. Mujahid A, Awan MJ, Yasin A, Mohammed MA, Damaševičius R, Maskeliūnas R, Abdulkareem KH (2021) Real-time hand gesture recognition based on deep learning YOLOv3 model. Appl Sci 11:4164
13. Liu Y, Nand P, Hossain MA, Nguyen M, Yan WQ (2023) Sign language recognition from digital videos using feature pyramid network with detection transformer. Multimed Tools Appl 82:1380–7501
14. Buttar AM, Ahmad U, Gumaei AH, Assiri A, Akbar MA, Alkhamees BF (2023) Deep learning in sign language recognition: a hybrid approach for the recognition of static and dynamic signs. Mathematics 11:3729
15. Rashtehroudi AR, Akoushideh A, Shahbahrami A (2023) PESTD: a large-scale Persian-English scene text dataset. Multimed Tools Appl 82:34793–34808
16. Sun S, Han L, Wei J, Hao H, Huang J, Xin W, Zhou X, Kang P (2023) ShuffleNetv2-YOLOv3: a real-time recognition method of static sign language based on a lightweight network. SIViP 17:2721–2729
17. Wang S, Guo C, Yang R, Zhang Q, Ren H (2023) A lightweight vision-based measurement for hand gesture information acquisition. IEEE Sens J 23:4964–4973
18. Wang C-Y, Bochkovskiy A, Liao H-YM (2021) Scaled-YOLOv4: scaling cross stage partial network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
19. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
20. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision
21. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. In: IEEE transactions on pattern analysis and machine intelligence, vol 36, no 6
22. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE conference on computer vision and pattern recognition (CVPR)

23. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI
24. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. http://arxiv.org/abs/1804.02767
25. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4: optimal speed and accuracy of object detection. http://arxiv.org/abs/2004.10934
26. Wang C-Y, Liao H-YM, Yeh I-H, Wu Y-H, Chen P-Y, Hsieh J-W (2020) CSPNet: a new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
27. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on machine learning
28. Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional network
29. Misra D (2019) Mish: a self regularized non-monotonic activation function. https://arxiv.org/abs/1908.08681.
30. Tan M, Pang R, Le QV (2020) EfficientDet: scalable and efficient object detection. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)
31. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR)
32. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: IEEE conference on computer vision and pattern recognition (CVPR)
33. Redmon J (2013) Darknet: open source neural networks in c. https://pjreddie.com/darknet/.
34. Pacal I, Karaman A, Karaboga D, Akay B, Basturk A, Nalbantoglu U, Coskun S (2022) An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets. Comput Biol Med 141:105031
35. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: The 31st international conference on neural information processing systems
36. Elbedwehy S, Medhat T, Hamza AMF (2022) Efficient image captioning based on vision transformer models. Comput. Mater. Contin. 73(1):1483–1500
37. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M (2021) An image is worth 16x16 words: Transformers for image recognition at scale In: 9. International conference on learning representations
38. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition
39. Elfwing S, Uchibe E, Doya K (2018) Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Netw 107:3–11
40. Gustineli M (2022) A survey on recently proposed activation functions for deep learning. *ArXiv* abs/2204.02921
41. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D (2020) Distance-IoU loss: faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence
42. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: 3. International conference on learning representations (ICLR)
43. Khari M, Garg AK, Crespo RG, Verdú E (2019) Gesture recognition of RGB and RGB-D static images using convolutional neural networks. Int J Interact Multimed Artif Intell 5(7):22–27
44. Rivera-Acosta M, Ruiz-Varela JM, Cisneros SO, Dominguez JR (2021) Spelling correction real-time American sign language alphabet translation system based on YOLO network and LSTM. Electronics 10(9):1035
45. Gangrade J, Bharti J (2020) Vision-based hand gesture recognition for indian sign language using convolution neural network. IETE J Res 69(2):1–10
46. Sincan OM, Keles HY (2020) AUTSL: a large scale multi-modal turkish sign language dataset and baseline methods. IEEE Access 8:181340–181355
47. Özcan T, Baştürk A (2021) ERUSLR: a new Turkish sign language dataset and its recognition using hyperparameter. J Fac Eng Archit Gazi Univ 36(1):527–542
48. Khan R, Wong WS, Ullah I, Algarni F, Haq MIU, Barawi MHB, Khan MA (2022) Evaluating the efficiency of CBAM-resnet using Malaysian sign language. Comput Mater Contin 71(2):2755–2772
49. Huang Y, Huang J, Wu X, Jia Y (2022) Dynamic sign language recognition based on CBAM with autoencoder time series neural network. Mob Inform Syst. https://doi.org/10.1155/2022/3247781
50. Pacal I, Karaboga D (2021) A robust real-time deep learning based automatic polyp detection system. Comput Biol Med 134:104519

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.