Sign Language Recognition from Digital Videos Using Feature Pyramid Network with Detection Transformer

Yu Liu

A thesis submitted to the Auckland University of Technology

in partial fulfillment of the requirements for the degree of

Master of Computer and Information Sciences (MCIS)

2022

School of Engineering, Computer & Mathematical Sciences

Abstract

Sign language recognition is one of the fundamental ways to assist deaf people to communicate with others. An accurate visual-based sign language recognition system using deep learning is a long-term research goal. Deep convolutional neural networks have been extensively considered in the last few years, many architectures have been proposed. Recently, Vision Transformer and other Transformers have shown apparent advantages in object recognition compared to traditional Computer Vision models such as Faster R-CNN, YOLO, SSD, and other deep learning models. In this thesis, we propose a Vision Transformer-based sign language recognition method related to DETR (Detection Transformer) to improve the current state-of-the-art sign language recognition accuracy. The method proposed in this thesis is able to recognize sign language from digital videos with high accuracy using a new Deep Learning model, ResNet152 + FPN (i.e., Feature Pyramid Network), which is based on Detection Transformer. Our experiments show that the method has excellent potential for improving sign language recognition accuracy. For instance, our newly proposed net ResNet152+FPN is able to enhance the detection accuracy by up to 1.70% on the test dataset of sign language. Besides, an overall accuracy 96.45% was achieved using the proposed method.

Keywords: Sign language recognition, ResNet152, Detection Transformer, Feature pyramid network

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: 刘宇 Date: 12 October 2022

Acknowledgment

First of all, I would like to thank my parents for their financial help. Their selfless love and care is the key of my motivations, which enabled me to successfully complete my master's study.

I would like to thank my main supervisor Dr. Parma Nand, who gave me a lot of critical guidance and advice on my writing. Also, I am truly grateful to my secondary supervisor Dr. Wei Qi Yan, who gave me a lot of help. During the study, they demonstrated not only professional knowledge, but also spiritual encouragement and support, which enabled me to successfully complete my study tasks. Finally, I would like to thank the Auckland University of Technology for the care given to students during the epidemic, which allows us to complete our studies in a safe and comfortable environment.

Yu Liu

Auckland, New Zealand

October 2022

Chapter 1
Introduction

In this chapter, five parts constitute the majority of this introduction chapter. For the first part, the background and introduction will be described. For the second part and the third part, the research question and our contribution will be presented respectively. In the fourth part, we will discuss the goal of this study. In the final part, the structure of this thesis will be presented.

1.1 Background and Motivation

In recent years, computational vision has developed a lot. Sign language recognition is significant for deaf or hearing-impaired people. Sign language is the main communication tool between hearing impaired people, hearing impaired people and people without hearing impairment. It builds a communication bridge in the communication process of these groups. Sign language comprises a series of gestures that can be recognized and translated into semantical symbols in texts.

The history of sign languages does not correspond to that of spoken languages. For example, though the uses of the same spoken language (with minor differences), NZSL, BSL and American Sign Language (ASL) are unrelated languages and are not mutually intelligible. It is now universally accepted in the linguistic community that sign languages such as NZSL (i.e., New Zealand Sign Language), ASL (i.e., American Sign Language), CSL (i.e., Chinese Sign Language), etc., are natural languages with comparable power to that of spoken languages. Indeed, the realization that sign languages are true languages is one of the great linguistic discoveries. It acts as a fundamental mode for hearing and speech-impaired people, without which communication between them and others might be difficult.

When communicating with the hearing impaired, it is difficult to obtain the effective information that the other party wants to express in time, which leads to great difficulties for deaf mutes in using public infrastructure and services. As only a small number of people suffer from hearing impairment, it is difficult for the state to invest special sign language gesture translators in public places for long-term sign language translation. Therefore, if sign language gesture recognition technology can be used to convert sign language gesture into text information for expression, it will undoubtedly provide an effective solution for communication with deaf mutes (Zhao et al., 2020).

Sign language recognition from digital images and videos is regarded as a type of behavior identification. With the development of science and technology, the technology of sign language recognition has undergone a series of improvements. The original researchers wore the device on the position of the hand and arm, and the angle and position of the hand and arm joints were detected and transmitted to the computer system. This method can effectively and completely transfer gesture information to the recognition system. However, these devices are expensive, which greatly increases the experimental cost. Therefore, the optical labeling method has replaced the data glove as a popular one. The researchers placed optical markers on the hand and transmitted changes in the position of a person's fingers and palms into the system via infrared transmission. However, such a method still requires complicated equipment so far. Experimenting with this method still requires high costs and cannot be applied to people's daily life. Nowadays, it is implemented by using machine learning approaches.

Although the accuracy and stability of sign language recognition can be improved with the help of external devices, the way of wearing external devices limits the flexibility of sign language expression. Therefore, a new vision-based sign language recognition method has emerged. This method trains images containing gestures through a network model, the trained model can effectively identify various sign language gestures.

In the initial stage of sign language recognition, machine learning methods were employed for sign language recognition. However, machine learning-based algorithms often require complex feature extraction engineering. Deep learning-based methods outperform traditional machine learning methods in natural language, vision, etc. In many recognition tasks, methods based on machine learning are not even comparable to deep learning. In terms of the classification accuracy of different methods on the ImageNet dataset, the classification error rate of deep learning methods is much lower than that of classical machine learning methods.

With the development of deep learning, there are a great deal of methods for sign language recognition (Krizhevsky, Sutskever & Hinton, 2012). Outstanding methods include You Only Look Once (YOLO) and Transformer.

In this thesis, we apply Detection Transformer (DETR) as a basic structure to solve this problem. The Transformer was proposed in 2017 to solve the problem that RNN models can be computed sequentially only.

The attention mechanism is employed to form an encoder-decoder framework for machine understanding. In 2020, Transformer was applied to Vision Transformer for image classification. In the work, the image is cut into blocks as serialized data for an encoder, and an attention mechanism is applied to match the image and classification labels. The novelty of this proposed method is the use of an attention mechanism to increase the speed of model training. It is a deep learning model entirely based on self-attention mechanism because it is suitable for parallel computing.

1.2 Research Questions

In this thesis, our main research is based on sign language recognition. We will explore deep learning methods suitable for sign language recognition and methods that can improve accuracy and efficiency. Therefore, the research questions of this thesis are:

(1) *Which deep learning techniques are best suitable for sign language recognition?*

For this research question, we will evaluate the existing deep learning methods for sign language recognition, mainly on the current hot topics YOLO series and Transformer series.

(2) *How much accuracy of the sign language recognition can be improved by Transformer based methods?*

Since the advent of the vision transformer, the research work using transformers for vision has continued to increase. We will create a transformer-based model and describe its structure. Besides, we will compare its performance with the existing YOLO series and transformer series to demonstrate its superior performance.

The core idea of this thesis is sign language recognition using deep learning. The selected deep learning methods will be compared for their performance, and the best method will be selected to improve the recognition accuracy. In addition, we will propose new evaluation metrics for the model and method. At the same time, we will train the experimental model and show our training results.

1.3 Contributions

In this thesis, we will compare the various models in the experiments through specific tasks. According to the experimental results, each index of the model is analyzed. In this research project, our model achieves the highest accuracy in real-time recognition compared to other models in the experiments. By the end of this report, we are able to,

(a) We proposed ResNet152 that makes use of a new backbone network. Furthermore, we proposed the addition of a FPN to the ResNet152 backbone which provides a superior prediction by using better manipulating the features from a sign language image. This structure is able to increase input features, which increases the quality of the final output.

(b) As part of this research work, we create our own dataset. This dataset is now publicly available for model training and testing at github.com.

(c) Improve the accuracy of sign language recognition by adding an autoencoder to thecompletion network.

Moreover, for result evaluations, we also compare our proposed method with other DETR-based models, the results surpass ResNet34, ResNet50 and ResNet101 in terms of AP, AP50, AP75, and F1 scores. We are going to apply all the models and obtain results based on our own dataset. The result and analysis will be discussed in order to get the conclusion for all methods.

1.4 Objectives of This Report

First of all, we will introduce all existing methods for sign language recognition. Secondly, we will present our proposed model. The framework of ResNet152+FPN+DETR is proposed to achieve sign language recognition. Thus, the objectives of this report are divided into three parts: Backbone, neck and detection part, corresponding to ResNet, feature pyramid network and Transformer. The backbone part will be the focus of the description. We will present the process that Feature Pyramid Network improves the quality of features and the combination of FPN and ResNet152. Finally, in this thesis, we will introduce methods of AP and mAP for comparisons by analyzing the experimental results. In addition, the loss function will be a part of the introduction.

1.5 Structure of This Report

The structure of this report is described as follows:

In Chapter 2, we will present the literature review. The research work in recent years will be reviewed. Besides, their methods, results and analysis will be invaluable experiences for researchers who explore future work. In brief, our work in this section is to introduce the basic deep learning methods and main achievements for sign language recognition. Based on their study, we will introduce our proposed method in a later chapter.

In Chapter 3, we will demonstrate our research method. Firstly, we will show the overall structure of our proposed method. Secondly, we will focus on describing our backbone part. We will introduce the principle of our proposed Feature Pyramid Networks (FPN) and ResNet152 and the combination between them. Thirdly, we will introduce the collection of our own database and the size of the data used for training and testing. We will describe the steps and results of the experiment later. We will conduct a controlled experiment by changing the backbone part of our proposed structure and later, work for the same experiments with the changed structure and record the results. Finally, we will introduce the methods for evaluating the results.

In Chapter 4, we will implement the algorithms and models. All the experimental data will be presented in the form of graphs and tables. Furthermore, the limitation of the proposed method will be mentioned in this section.

In Chapter 5, we will discuss the results from the last chapter and analyze them in terms of our evaluated methods. Finally, the conclusion and future work will be presented in Chapter 6.

Chapter 2
Literature Review

The focus of this thesis is on sign language recognition based on Detection Transformer (DETR). In this chapter, we will introduce previous research and methods in the order of development. Besides, the related work for the Transformer will be included.

2.1 Introduction

Nowadays, sign language recognition has increasingly become a hot research topic. Its application fields are increasingly extensive. In education, sign language recognition can help teachers have effective conversations with deaf children. In daily life, sign language recognition can help ordinary people and deaf people communicate normally. Even in wartime, intelligence personnel can use encrypted gestures to pass information through the camera. The sign language recognition system can realize the recognition of self-created sign language by training the model.

In the previous chapter, with the development of technology, sign language recognition technology has undergone a series of improvements. The original sign language recognition relied on expensive equipment to detect the position of each joint of the hand through a device worn on the hand, this device is called 'sensor gloves' (Mehdi & Khan, 2002).

Later, infrared scanning of hand and arm movements were harnessed to create 3D images to recognize sign language. However, this identification method still requires expensive and not portable equipment to complete. These methods are difficult to apply in the daily life of ordinary people. In order to enable sign language recognition to enter the daily life of the general public, in this thesis, we used ordinary cameras combined with deep learning methods for sign language recognition. In this way, people can complete sign language recognition by simply downloading a program from a smartphone and combining it with the mobile phone camera. Such a method has low cost and easy portability of the equipment and can be a method that is able to be applied to the public.

2.2 Recognition Using Sensor Gloves

The key to sign language recognition lies in the capture of hand motion information, the extraction and classification of gesture information, and the processing of gesture posture information. Object detection and recognition in motion is a hot topic in recent years. Intelligent monitoring has been applied in various industries, including manufacturing, medicine and military. Fast and accurate detection of moving objects has become the focus of research. (Han, 2015).

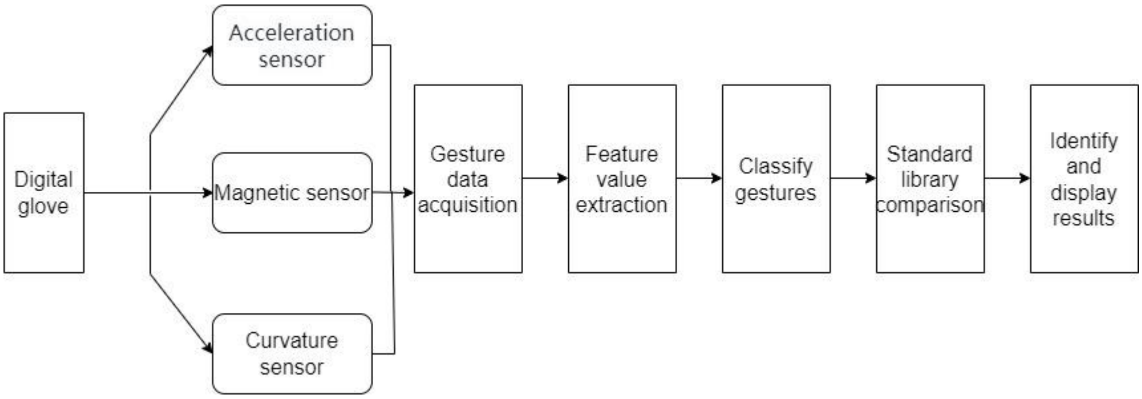


Figure 2.1: Recognition process of sensor gloves

As shown in Fig.2.1, the data glove records the real-time state of the hand by directly detecting the palm, the position of the fingers and the angle between the fingers through the magnetic sensor. The track information of the hand is detected by the acceleration sensor. Moreover, the curvature sensor records the curvature of the palm and fingers in real-time. Using these three sensors can obtain completed gesture motion information. Next, effective features are extracted from the gesture information obtained in the previous step. The next step is to perform preliminary action classification through the extracted features, and then compare and match the classified actions with those in the standard library. Finally, output and display the recognition results.

Kanwal et al. proposed a pattern recognition approach system by using sensor gloves and employed Principal Component Analysis (PCA) for feature extraction (Kanwal, Abdullah, Ahmed, Saher, & Jafri, 2014). In the experiment, they adapted five curvature sensors on each glove to obtain finger curvature information, one for each finger, an accelerometer is applied to track the position of the hand. To enhance the accuracy of recording flexion values, they tested the maximum and minimum values at full flexion and extension in five men and five women, respectively, and took the average as a calibration value.

2.2.1 Classification Using Machine Learning

In traditional sign language recognition algorithms, sensor gloves are worn to obtain sign language information and perform feature extraction. Next, the extracted features are classified using machine learning methods. Different machine learning algorithms have different effects.

We firstly introduce the principle of the classifier based on naïve Bayes theory. This method includes prior and posterior probabilities based on probability theory. Prior probability is a probability estimation of random events based on existing conditions without considering any relevant factors. Posterior probability refers to the probability prediction of random events under the condition of considering relevant factors. This classifier works on input vectors such as x, where x represents a subset of the attribute and objective function, the input vector value, feeds a dataset containing training vectors to the classifier and feed new instances for testing (Sahoo, 2021). Let V_{MAP} is the most suitable target for the attributes $(a_1, a_2, a_3 \dots \dots ., a_n)$,

$$V_{MAP} = \arg\max_{v_j \in V} P\left(v_j \middle| a_1, a_2, a_3 \dots \dots ., a_n\right) \tag{2.1}$$

After applying Bayes theorem, eq. (2.1) is rewritten as

$$V_{MAP} = \arg\max_{v_j \in V} \frac{P\left(v_j \middle| a_1, a_2, a_3 \dots \dots ., a_n\right) P(v_j)}{P(a_1, a_2, a_3 \dots \dots ., a_n)} \tag{2.2}$$

$$= \arg\max_{v_j \in V} \frac{P(v|a_1, a_2, a_3 \dots \dots., a_n|v_j)P(v_j)}{P(a_1, a_2, a_3 \dots \dots., a_n)} = \arg\max_{v_j \in V} P(v_j|a_1, a_2, a_3 \dots \dots., a_n)P(v_j) \quad (2.3)$$

In Eq. (2.3) are determined by the number of training samples. The frequency of the target value *(v_j)* in the training samples will determine the estimated value of *$P(v_j)$*. In the case of a few training samples, the evaluation of multinomial *$P\ (a_1, a_2, a_3 \dots \dots., a_n|v_j)$* properties are not possible.

The probability of an attribute value is the product of the independent attributes, so the above formula can be written as eq. (2.4). Therefore, eq. (2.3) is written as eq. (2.5).

$$P(a_1, a_2, a_3 \dots \dots., a_n\ v_j)\ = \prod_i P(a_i|\ v_j) \quad (2.4)$$

$$\text{Naive Bayes classifier } (VNB) = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|\ v_j) \quad (2.5)$$

VBN refers to the target value of the input test sample, $P(a_i|v_j)$ are all from the training sample, and different attribute values are multiplied by different numbers of target values. Equation (2.4) presents the main principle of Naive Bayes. Naive Bayes estimates and predicts based on $P(v_j)$ and $P(a_i|v_j)$ in the training samples and is also the main mechanism of this method for classification.

The next is *k*-nearest neighbours (*k*-NN) algorithm. The algorithm is a classification algorithm, and its application fields include character recognition, text classification, image recognition and other fields. The main principles of the algorithm are described below. Assuming that a sample is similar to the *k* samples in the dataset, the class to which the sample with the largest number of *k* samples belongs is regarded as the class to which the hypothetical sample belongs. And the closer the distance, the greater the similarity.

In the *k*-NN algorithm, for two points **x** ($x_1, x_2, .., x_n$) and **y** ($y_1, y_2,..,y_n$) in *n* -dimensional space, the distance between them can be expressed as If there are two testing vectors **x** and **y**, the distance between them written as *d*(*x, y*). Besides, the distance is following eq.

(2.6).

$$d_{xy} = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \tag{2.6}$$

For the input training dataset: $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$. In this dataset, $x_i \in X \subseteq R^n$ where x represents the n-dimensional instance feature vector. $y_i \in Y = \{m_1, m_2, \ldots, m_k\}$. The y_i represents the category in the instance, where i = 1, 2, . . ., n. The predicted instance is x. The output is the class **y** to which the predicted instance x belongs.

$$y = \arg max \sum_{x_j \in N_{k(x)}}^{n} I(y_i, m_i), i = 1, 2, \ldots, N; j = 1, 2, \ldots, K \tag{2.7}$$

In eq. (2.7), we represent the indicator function:

$$I\ (x, y) = \begin{cases} 1, if\ x = y \\ 0, if\ x \neq y \end{cases} \tag{2.8}$$

If the value of k is large, the input training samples form a convex polyhedron in the classification. In the case of k = 1, only two classes are involved. At this point, the classification space is divided into two areas, one for a specific class and the remaining space for the second class. For any point xi in space, its class distinction will be calculated according to eq. (2.7).

2.3 Recognition Using Infrared Devices

The approaches for pattern classification using machine learning algorithms can achieve considerable recognition accuracy. However, methods of classification and learning require large datasets and handcrafted features. In addition, it requires a lot of time to train a classifier. Researchers began to explore an easier way to perform sign language recognition. One of the most popular devices used to detect gesture information using infrared devices is called the Kinect. The biggest advantage of Kinect is the fingertip detection and gesture recognition algorithms that can obtain 3D depth information. Kinect includes an RGB camera, infrared camera and infrared transmitter. These three components can make up a depth sensor. Depth sensors can detect the distance between objects in space and the sensor uses black and white spectra and generate 3D depth images.

Aliyu et al proposed Fisher Linear Discriminant Analysis (LDA) for sign language recognition based on Microsoft Kinect. LDA is used for feature extraction and classification. Its main function is to maximize the removal of data with less information, thereby maximizing category classification. The principle of removing data with less information is to project the data through a scatter matrix, sort the eigenvalues in the matrix, and discard the features that are close to 0. In the end, they obtain an overall accuracy of 99.8%. In summary, higher recognition accuracy can be achieved using infrared devices. However, complex devices still cannot be used in people's daily life. Next, we will introduce sign language recognition based on deep learning using ordinary cameras.

2.4 Recognition Based on Deep Learning

In recent years, deep learning methods have been applied by more and more researchers in visual object detection. Deep learning methods are suitable for text-based natural language processing, image recognition and speech recognition, and more importantly, deep learning is applied to object detection in computer vision (LeCun, Huang, & Bottou, 2004; Hinton, Osindero, & Teh, 2006). Deep neural networks continuously improve the network model through pre-training. In the model, the greater the depth of the network, the greater the number of hidden layers. Currently, widely used artificial neural networks include convolutional neural networks, single-shot multi-frame detectors, you only look once, etc. Sign language recognition is one of the most popular research areas. Sign language recognition based on visual methods is mainly divided into static gesture recognition and dynamic gesture recognition. From a visual point of view, a static gesture is the static state of a dynamic gesture. Continuous dynamic video or real-time detection can be detected through frame-by-frame static gesture recognition. In the recognition of consecutive frames, the relationship between the previous and previous frames is further analyzed to improve and perfect the gesture system.

In deep learning, Long and Short-Term Memory (LSTM) has played a huge role as an alternative to Hidden Markov Model (Baum, 1968). LSTM can combine the discrete temporal information with contextual information. This feature has been widely used in the field of recognition, especially for the recognition of continuous actions.

2.4.1 Recognition Using CNN

LeNet-5 network structure was firstly proposed by Yann LeCun et al. In 1998, convolutional neural networks were first able to be trained end-to-end. Subsequent convolutional neural networks have been developed from versions of LeNet-5. CNN combines image processing and artificial neural network, which can effectively learn and extract features from pictures. Therefore, CNN has been widely used in the field of recognition.

2D CNN is a technique based on single-frame image recognition and is used for feature map extraction to recognize gestures (Koller, Ney, & Bowden, 2016; Wu, Ishwar, & Konrad, 2016). Furthermore, 2D CNN is extended to 3D CNN. This method performs feature learning by employing 3D filters in 3D convolutional layers (Liu, Zhang, & Tian, 2016; Molchanov, Yang, Gupta, Kim, Tyree, & Kautz, 2016). Neverova et al. proposed a method to detect and segment hands in a dataset, and the model achieved 82% accuracy (Neverova, Wolf, Taylor, & Nebout, 2014). CNN models have been widely used in image classification and recognition tasks. A Gaussian skin color model and background subtraction for gesture recognition are proposed. The Gaussian skin tone model detects the effect of light on skin tones and filters out non-skinned parts of the image. This method achieves 93.80% accuracy in the tested dataset (Han, Chen, Li, & Chang, 2016).

2.5 Recognition based on YOLO

YOLO is called You Only Look Once. The name means that objects and categories can be identified in a picture with just one glance. The biggest feature of YOLO is based on full image scanning. In terms of network design, the significant difference between YOLO and RCNN and Faster RCNN is that the training and detection of YOLO are performed in a separate network. The training process of RCNN and Faster RCNN needs to be carried out in multiple modules. Therefore, compared to models such as RCNN, YOLO has a faster detection speed. Region-based methods are also known as two-stage methods. YOLO-based detection is an end-to-end detection. End-to-end means that the input is the original data and the final result is the output. In this method, the intermediate process does not require special processing or separate processing. Traditional machine learning requires the manual extraction of image features. However, there are two obvious difficulties in manually extracting features. One is that feature extraction is difficult to design, and the second is that manual feature extraction is extremely time-consuming in the case of a huge amount of data.

With the developed object detection, Girshick et al. came up with the R-CNN target detection network in 2014, followed by detection algorithms in two steps including Fast R-CNN and Faster R-CNN, the recognition accuracy and speed will be further improved (Girshick, Donahue, Darrell, & Malik, 2014). Meanwhile, Redmon et al. proposed the YOLO algorithm for the detection of one-step in 2016. Considering previous algorithms, its speed of detection has been greatly improved, which has been favored by more people.

Redmon et al. put forward the YOLOv2 algorithm in 2017, which further improved the recognition rate (Redmon, Divvala, Girshick, & Farhadi, 2016). Redmon and Farhadi proffered the YOLOv3 algorithm in 2018, which further improved the recognition rates of small targets (Redmon, & Farhadi, 2018). Ni et al. (2018) improved the YOLOv2 algorithm and pruned it on this basis and proposed a lightweight model with only 4M models (Ni, Chen, Sang, Gao, & Liu, 2018). With the appearance of 3D convolution, a few methods of gesture recognition based on 3D CNN have been explored, for example, Abavisani et al. proposed multimodal knowledge to train single-mode 3D CNN in 2019.

To solve the detection problems such as track plate cracks, irregular crack growth, and low light environment at night that occur on high-speed railroad tracks, an improved algorithm based on the YOLO algorithm was proposed. The application of YOLO algorithm in sign language recognition is investigated to improve the accuracy and speed of detection under the backgrounds of near skin colors and light intensity (Huang, Pedoeem, & Chen, 2018). YOLO algorithm is a one-stage and end-to-end detection method. It can greatly improve the operation speed by automatically extracting the characteristics of the target through the convolutional neural network. Because of the excellent performance of the YOLO algorithm in the target detection task, the YOLO algorithm is applied to sign language recognition (Jiang, Ergu, Liu, Cai, & Ma, 2022).

The comparison of YOLO algorithms show that the YOLO algorithm performs well in sign language recognition. YOLO algorithm models are available in the YOLO series. YOLOv1 network structure adds 4 convolution layers and 2 full connection layers on the basis of 20 layers of the GoogLeNet. YOLOv2 took advantage of Darknet-19 as its basic structure (Ćorović, Ilić, Đurić, Marijan, & Pavković, 2018). The structures of the Darknet-19 network are similar to that of the VGG network. Both of them use small convolution kernel operations. YOLOv2 borrows from the Faster R-CNN algorithm and introduces anchor boxes to generate more candidate boxes for each mesh. YOLOv3 algorithm model was improved on the basis of the YOLOv2 model, using a deeper network structure Darknet-53 (Liu, Nouaze, Touko Mbouembe, & Kim, 2020). Darknet-53 is similar to that of the ResNet-101 network, but the recognition speed of the Darknet-53 is twice that of the ResNet101 network. Darknet-53 alternates 3 × 3 and 1 × 1 convolution and residual structures, while using the FPN architectures (feature pyramid network for object detection) to achieve multi-scale detection (Lin, Dollar, Girshick, Hariharan, & Belongie, 2017). YOLOv3 takes use of 9 anchor boxes, each scale corresponds to 3 anchor boxes, the small scale uses a large anchor box, and the large scale uses a small anchor box, which is conducive to small target detection.

As a basic research field in computer vision, visual object detection has been widely adopted in the industry, among which YOLO series algorithms have gradually become the preferred frameworks for most industrial applications due to their good comprehensive performance. So far, the industry has delivered a group of YOLO detection frameworks, among which YOLOv5, YOLOX and PP-YOLOE are the most representatives (Rivera-Acosta, et al, 2021).

2.5.1 Prediction Process of YOLO

Based on the basic task of finding an object in an image and recognizing its category and location, the principle of YOLO is based on the prediction of the entire image. YOLO outputs the information, category and location of all detected objects at once. In the first step, YOLO will firstly segment the image into s^2 grids, and each of them has the same size. In YOLO, each square can identify an object. Different from the sliding window method, YOLO only requires the center of the object to be in the square. In the specific implementation method, each square must predict B bounding boxes. Each bounding box has 5 quantities including the center position of the object (x, y), h (height), w (width) and this is the confidence level of the secondary prediction. This means that in s^2 grids, the number of bounding boxes in each grid is B, and the classifier can identify C different objects, then the length of the ground truth is expressed as $S \times S \times (B \times 5 + C)$.

In YOLO, there is a very important parameter called confidence. Equation (2.9) for calculating confidence is,

$$C = \Pr(obj) * IOU_{truth}^{pred} \tag{2.9}$$

The full name of IOU is intersection over union. It reflects the similarity of the two boxes.

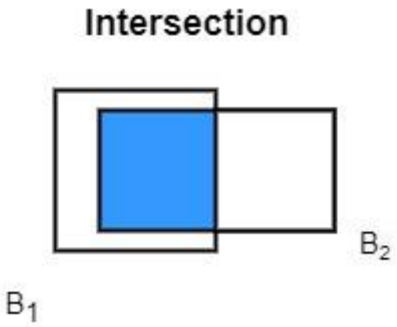


Figure 2.2: Intersection of bounding box and true object

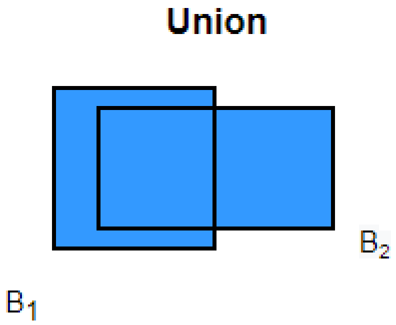


Figure 2.3: Union of bounding box and true object

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} \tag{2.10}$$

In equation (2.9), IOU_{truth}^{pred} is to predict the intersection over union of the bounding box and the real object box. $\Pr(obj)$ is the probability that a grid has an object. The ground truth value is 1 when there are objects in the grid, and the ground truth value is 0 when there are no objects.

According to the principle of YOLO, a phenomenon is that a large object is recognized by many small bounding boxes. However, in the end, only one bounding box will be selected. The technique used here is called non-maximal suppression (NMS). For example, an object has four prediction boxes $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$ and $\mathbf{B}_4$. Confidence can predict the probability value of the object in the bounding box, and each prediction box generates corresponding confidence. In the generated prediction box, pick the box with the largest confidence value and delete the rest. Another function of NMS is to determine whether the bounding box recognizes the same object. The specific steps are to assume that B1 is the largest bounding box and calculate the IOU of the largest bounding box and several other bounding boxes. If it exceeds a set threshold, it is considered that the two bounding boxes predict the same object and delete the confidence. A bounding box with a small value. Finally, the type and specific location of the object in the picture are judged by combining the type recognized by the large bounding box and grid.

2.5.2 Recognition Based on YOLOv4

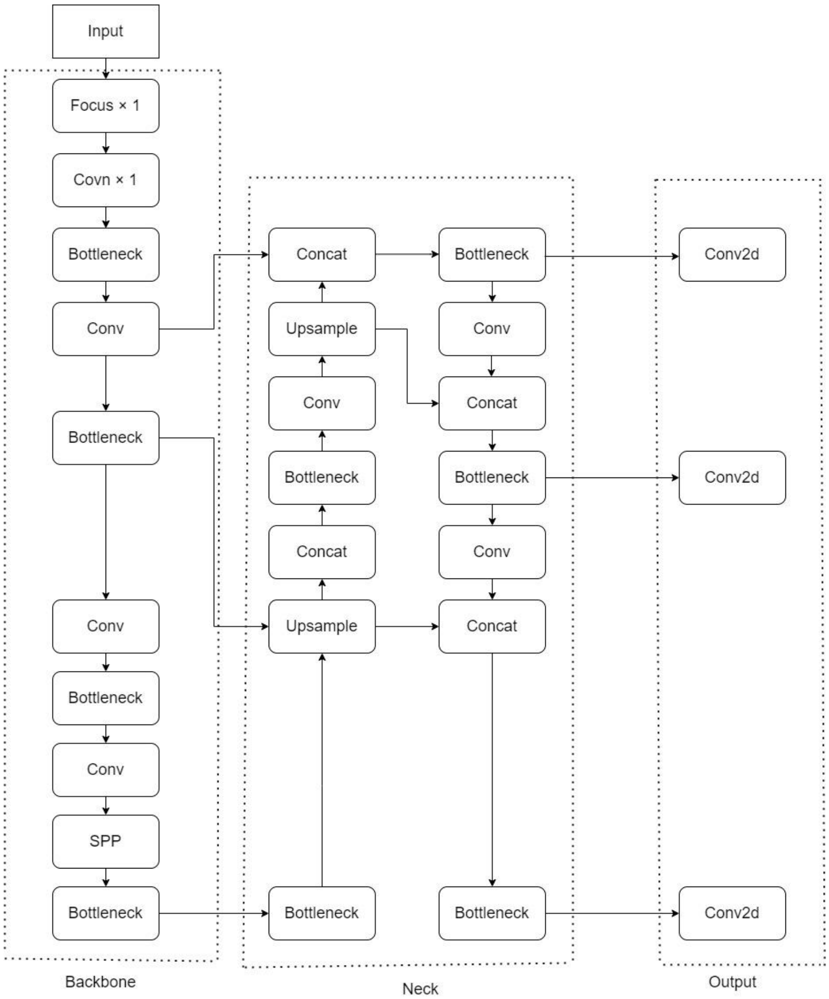


Figure 2.4: YOLO structure

In Fig. 2.4, YOLOv4 can be split into three parts: Backbone, neck and output. A series of innovations have been carried out on the YOLOv4 input side to enable the device to achieve good results with a single GPU, the most significant of which is the data enhancement Mosaic. Based on the CutMix data enhancement method proposed in 2019 (Yun, Han, Joon Oh, Chun, Choe, & Yoo, 2019), Mosaic continued to use the method of image stitching and increased the number from two to four. The way of stitching pictures is random scaling, random cropping and random arrangement.

In the stitching process, mosaic is use of a loop to perform image stitching processing. This loop is divided into four times and a batch of images is obtained each time. Stitching is performed while acquiring images, and the image stitching process is completed after four cycles are completed. There are two ways to acquire images: the sequential acquisition method and the random acquisition method. The sequence acquisition method is suitable for the case where the image data set is small. After the image is acquired, operations such as gelatinization, Gaussian noise, exposure, and hue shift are performed on the image. The second method is the random acquisition method. This method is suitable for large image datasets. Besides, it is suitable for directly combining and splicing images, which is able to avoid duplication of combined images. Compared with the random acquisition method, the advantage of this method is that it does not increase the computational load excessively and can ensure that the data enhancement needs are met.

The introduction of Mosaic data enhancement can effectively alleviate the problem of uneven distribution of small targets, and cropping multiple images into one image can improve the accuracy of target recognition. As shown in Table 2.1, Kisantal et al. defined the size of small, medium and large targets (Kisantal, Wojna, Murawski, Naruniec, & Cho, 2019).

Table 2.1: Small, medium and large objects definitions

	Min rectangle area	Max rectangle area
Small object	0×0	32×32
Medium object	32×32	96×96
Large object	96×96	∞×∞

In Table 2.1, the definition of a small target is an object whose length and width of the target frame are between 0×0~32×32. However, in the dataset, the proportion of small, medium and large targets is not uniform. As shown in Table 2.2, the proportion of small targets in the coco data set reaches 41.4%, and the number is higher than that of medium and large targets. However, in the training data set, only 52.3% of the pictures have small objects, compared to the proportion of medium objects and large objects more evenly. For this situation, Mosaic data augmentation is proposed. The advantage of this approach is that stitching with random distributions enriches the test dataset. Immediate scaling can increase the small target images, so that the proportion of small, medium and large target images is more uniform, thereby improving the robustness of the network.

Table 2.2: Ratio of small, medium and large object images

	Small	Mid	Large
Ratio of total boxes(%)	41.4	34.3	24.3
Ratio of images included(%)	52.3	70.7	83.0

A new Backbone structure CSPDarknet53 appeared in YOLOv4. Based on the experience of Cross Stage Partial Network (CSPNet) proposed in 2020, the resulting Backbone structure contains 5 CSP modules (Wang, Liao, Wu, Chen, Hsieh, & Yeh, 2020). The size of the convolution kernel in front of each CSP module is 3×3, *stride*=2, so downsampling is possible. Among the 5 CSP modules, the size of the input image is 608 × 608, and the size of the feature map is halved each time it passes through a module. Therefore, the size of the feature map after 5 passes of the CSP module is 19 × 19.

The application of CSPNet has brought great improvement to the network. Its main function is to reduce the problem of a large amount of inference calculation. The CSP proposer believes that the high computational cost of inference is caused by the repetition of gradient information in the network. The feature map of the base layer is divided into two parts by the CSP module, and then they are merged through a cross-stage hierarchy, which can not only reduce the amount of calculation but also ensure the accuracy. Therefore, applying CSPDarknet53 to the Backbone structure has three advantages. One is to enhance the learning ability of CNN, so that the accuracy can be maintained while being lightweight. The second is to reduce the amount of calculation. The third is to reduce memory costs.

Another significant advantage is the use of the mish activation function in YOLOv4 (Misra, 2019). According to the test experiments on the ImageNet dataset, the accuracy of TOP-1 and TOP-5 using the Mish activation function is higher than when it is not used.

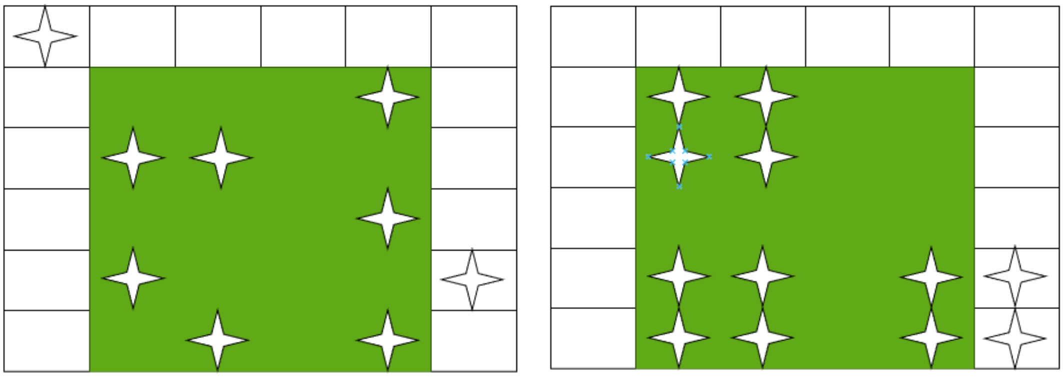


Figure 2.5: Dropout (left) and dropblock (right)

In YOLOv4, Dropblock is a regularization method that is applied to alleviate overfitting (Bochkovskiy, Wang, & Liao, 2020). In conventional Dropout algorithm, random deletion is used to reduce the number of neurons, thereby making the network simpler. However, dropout does not work well on convolutional layers because convolutional layers usually include convolution, activation and pooling layers. The pooling layer itself acts on adjacent units, so even if it is randomly dropped, the convolutional layer can still learn the same information from adjacent activation units. Dropblock also adopts the method of pruning information to alleviate over-fitting. However, based on the effect of dropout on the convolutional layer, Dropblock prunes and discards the entire local area. According to the cutout data augmentation method, cutout removes part of the input image, and Dropblock applies cutout to each feature map (DeVries, Taylor, 2017). The difference is that Dropblock does not use a fixed ratio, but starts with a small ratio that increases linearly during training. In contrast, cutout can only act on the input layer, while Dropblock can modify and prune feature maps at various levels to achieve better results.

In the field of object detection, a plenty of layers that are usually inserted in the backbone and output layers are called Neck. The main function of the Neck is to better extract fusion features. In YOLOv4, an important improvement in the Neck section is the use of the SPP module.

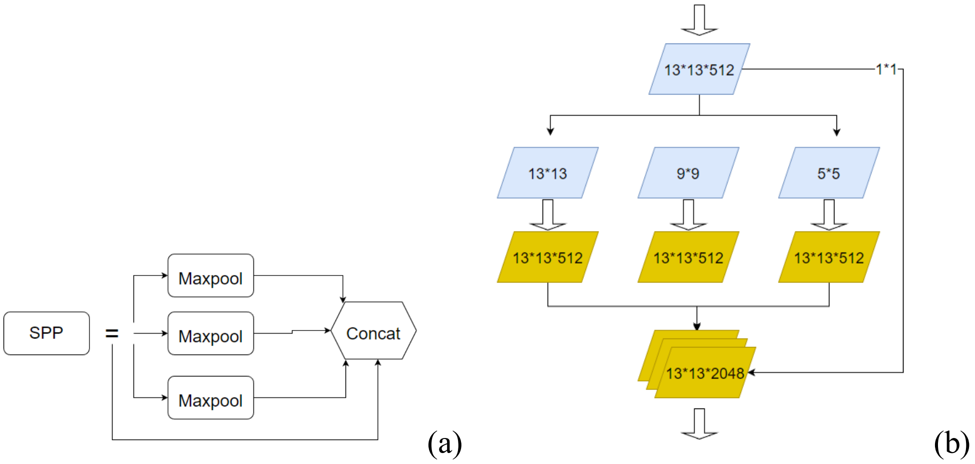


Figure 2.6: SPP module

The SPP module uses the maximum pooling method of k={1×1,5×5,9×9,13×13} to concat the feature maps of different scales. Maximum pooling uses padding operation, and the moving step size is 1. For example, using a pooling kernel of 5×5 size, padding=2, the size of the feature map after pooling is still 13×13. After comparing and testing the SPP module of YOLO target detection, they found that using the SPP module is more effective than simply using k×k max pooling to increase the acceptance range of backbone features, and significantly separates the most Important contextual features (Huang, Wang, Fu, Yu, Guo, & Wang, 2020).

2.5.3 Recognition based on YOLOv5

YOLOv5 is a single-stage target detection algorithm released by Ultralytics LLC. It represents Ultralytics' open-source research on future vision AI methods, which contains lessons learned through thousands of hours of research and development. Compared with YOLOv4, YOLOv5 has less average detection accuracy and smaller average weight files, training time and reasoning. The network structure of YOLOv5 is divided into four parts: Input, Backbone, Neck, and Head.

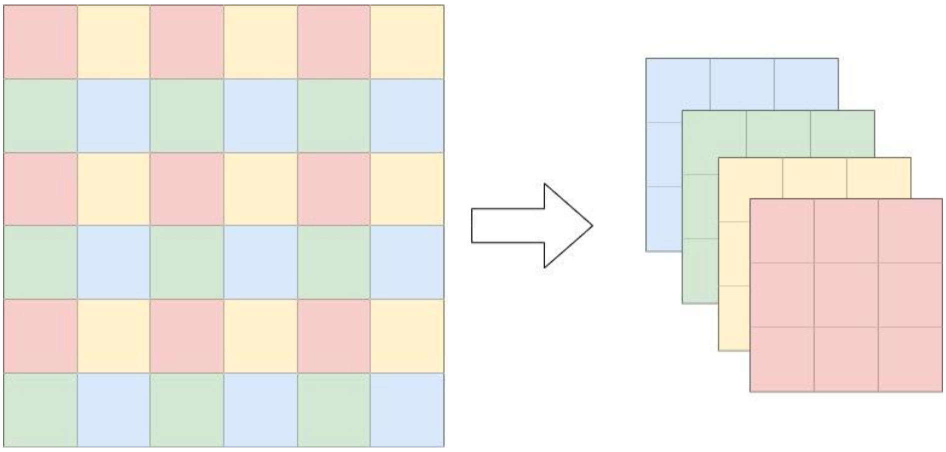


Figure 2.7: Cut slice operation

Next, we explain the improved part of YOLOv5. Compared with YOLOv4, YOLOv5 has several key technical improvements. As shown in the figure below, this is the image-cutting process of YOLOv5, which is one of the important improvements of YOLOv5. In the focus module of YOLOv5, the first layer of the network is replaced with a 6×6 convolutional layer. Divide each 2×2 adjacent pixel into a patch, then stitch together the same position pixels in each patch to get 4 feature maps, and finally connect a 6×6 convolutional layer. The two methods are theoretically equivalent, but for existing GPU devices, using a 6×6 convolutional layer is more efficient than using the Focus module.

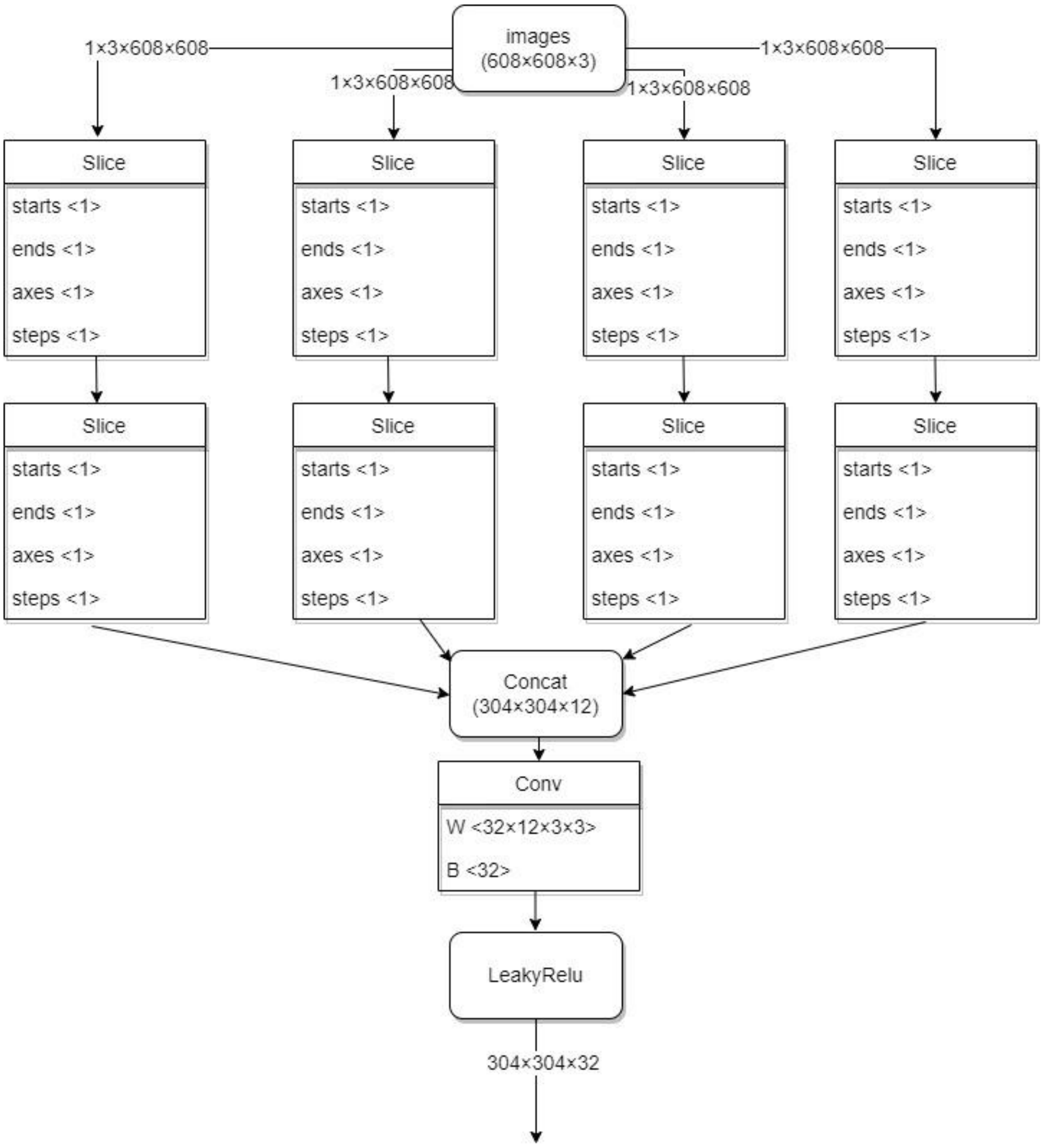


Figure 2.8: Channel changes process

One of the most important parts is the focus structure. As shown in Fig. 2.8, the role of this module is to cut the image. An image with a size of 608×608×3, after being cut by Concat, outputs a feature map with a size of 304×304×12. The next step is to pass through a Conv layer of 32 channels, resulting in a feature map of size 304×304×32. The specific process of cutting the picture is to take a value for every other pixel, and in this way, four images are obtained. The width and height information is concentrated in the channel and expands the input channel by a factor of four. As shown in the figure, the number of image channels after cutting is expanded to 12 compared to the original channel.

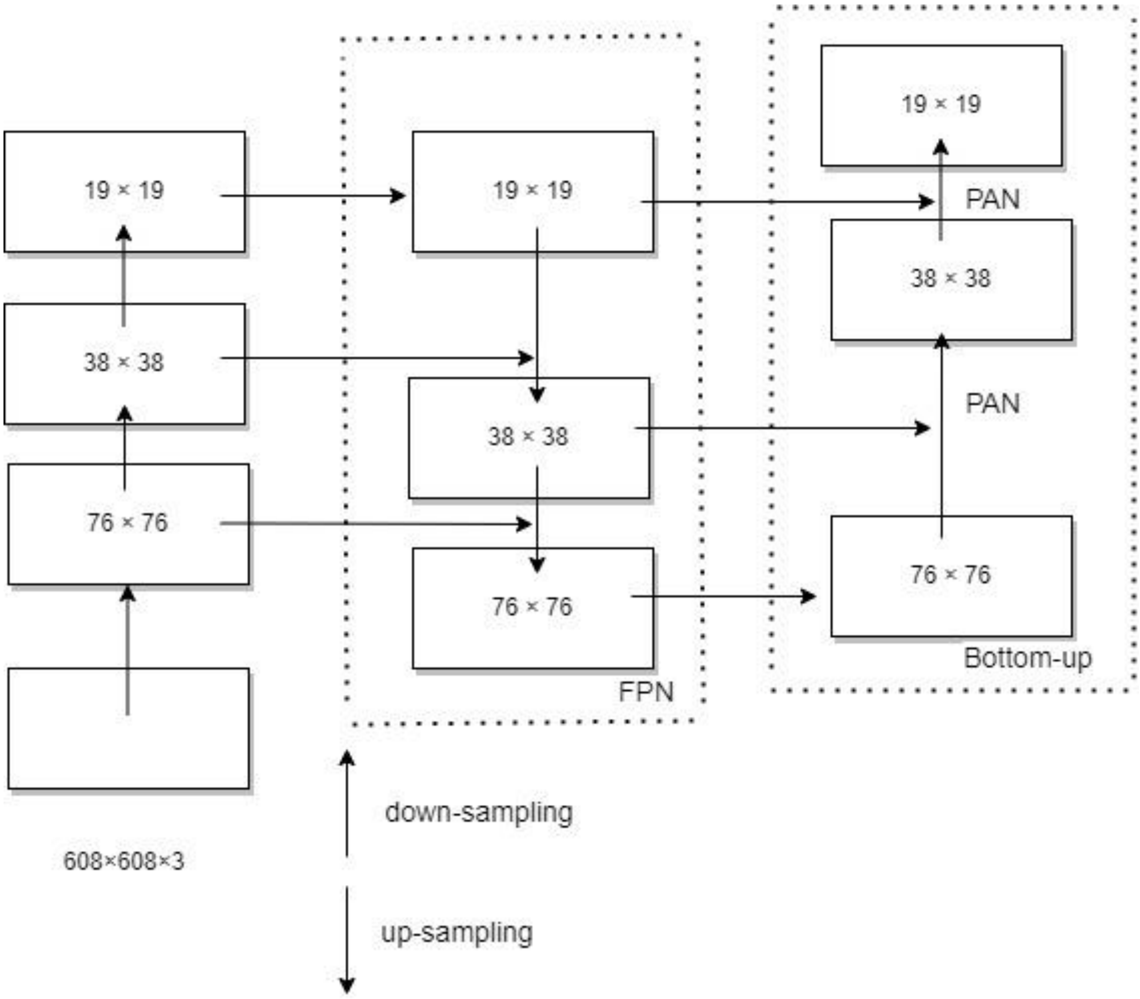


Figure 2.9: Process of FPN + PAN

One of the most significant improvements in YOLOv5 is the Neck section. The Neck part makes use of a combination of Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) structures. As shown in Figure 2.9, FPN passes strong semantic features down from the high layers to the bottom layers. It can strengthen the semantic information of the entire feature pyramid. PAN is mainly used to provide location information. It is passed from the bottom layer to the high layer to complete the corresponding FPN information.

2.5.4 Recognition based on YOLOv6

Recently, Meituan Visual Intelligence Department has developed a target detection framework YOLOv6 dedicated to industrial applications, which focuses on the accuracy of detection and efficiency of reasoning at the same time. In the processes of development, the Visual Intelligence Department has continuously optimized, and learned from cutting-edge developments and scientific research achievements in industry and academia (Ben, 2022). The experimental result on COCO, an authoritative target detection dataset, shows that YOLOv6 outperforms other algorithms with the same volumes in detection speed and accuracy, and supports deployments of a variety of platforms, greatly simplifying the work of adaptation during project deployments. This is open source, hoping to help more students (Zhang et al., 2022).

YOLOv6 has been a target detection framework developed by Meituan Visual Intelligence Department, which is dedicated to industrial applications. This framework also focuses on detection accuracy and reasoning efficiency. In the size models commonly used in the industry, YOLOv6 nano can achieve an accuracy of 35.0% AP on COCO and a reasoning speed of 1242 FPS on T4. YOLOv6-s has an accuracy of 43.1% AP on COCO and a reasoning speed of 520 FPS on T4 (Starzyński et al., 2022). In terms of deployment, YOLOv6 supports the deployment of GPU (TensorRT), CPU (OPENVINO), ARM (MNN, TNN, NCNN) and other different platforms, greatly simplifying the adaptation during project deployment.

To improve the accuracy and speed of the proposed method for sign language recognition, in the thesis, the e-parameterization Visual Geometry Group (RepVGG) is applied in YOLOv6 to replace traditional VGG. Traditional VGG has obvious drawbacks. For example, VGG consumes more computing resources and makes use of more parameters, resulting in more memory usage, and the model detection efficiency is low. Compared with VGG model, RepVGG has a more compact structure, which is more computationally efficient and uses fewer resources. RepVGG is characterized by reparameterization. In the training phase, a multi-branch model will be trained, then the multi-branch model will be equivalently converted into a single-channel model by using parameterization. Finally, the single-channel model will be adopted during inference. Because in the training process, the multi-branch structure often obtains higher performance benefits, which are used to improve the network performance in the training section. However, after the training section, it will be converted into a single-channel structure for inference, so that it will significantly improve the inference speed.

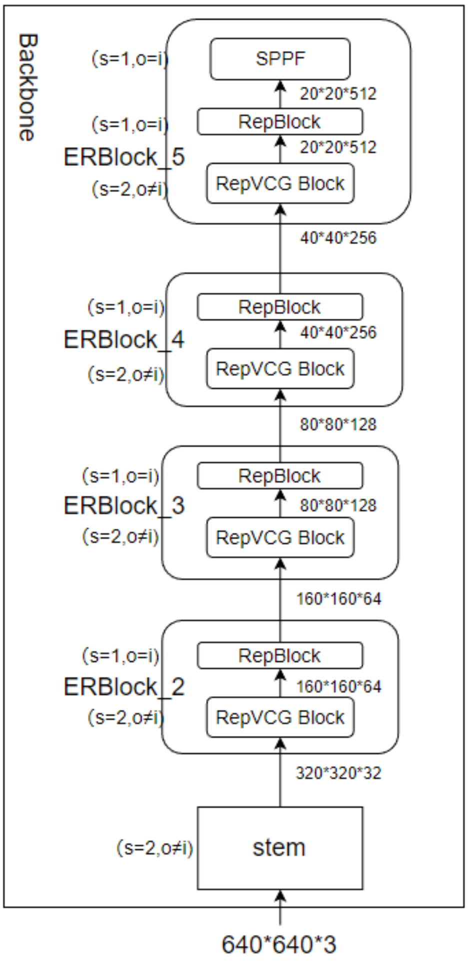


Figure 2.10: Backbone of YOLOv6

The backbone of YOLOv6 is shown in Figure 2.10, where *s* stands for stride, *o* shows out-channel, and *i* represents in-channel, where "*o*=*I*" means out-channel quals to in-channel, "*o*≠*I*" refers to that out-channel does not correlate with in-channel, but the value of out-channel is not equal to in-channel.

YOLOv6 consists of four parts: Input, backbone, neck, and head. The processing of the image is split into the following steps: The first step is to preprocess the image, and uniformly process the image into an RGB image with a size of 640×640, as well as input the backbone network. The second step is to output three layers of feature maps of different sizes in the neck layer through the Rep-PAN network according to the three-layer output in the backbone network. The third step predicts the feature map input to the head layer. Prediction includes classification prediction, background classification prediction, and bounding box prediction. The final step is the output results. Next, we will introduce the main improvement in YOLOv6 – RepVGG.

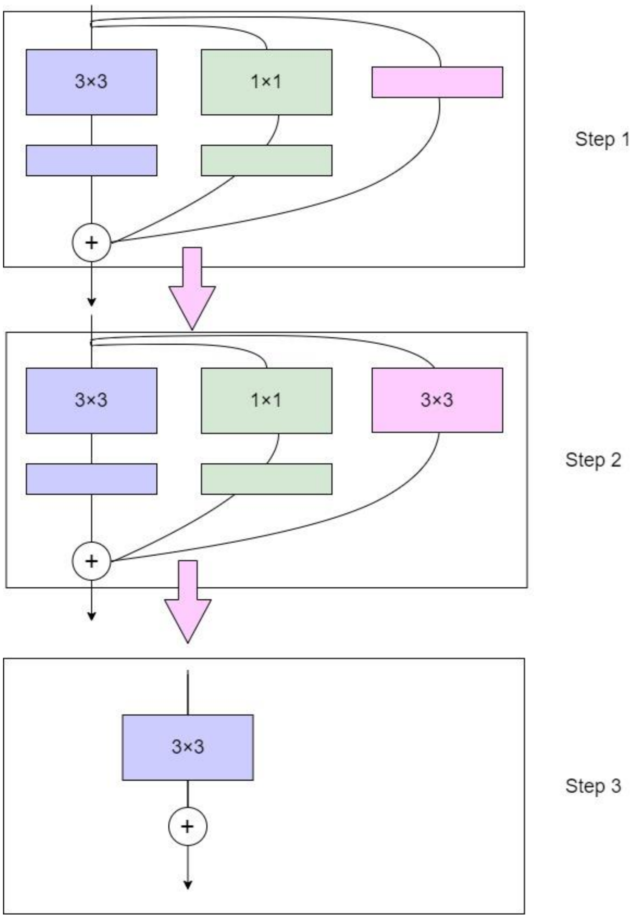


Figure 2.11: Structure of RepVGG

The core of RepVGG is the conversion of a multi-branched structure into a one-way structure. As shown in Fig. 2, in the training phase of RepVGG, its structure borrows from ResNet, while introducing residual structure and 1×1 convolution. The inference stage is a combination of 3×3 convolution and ReLU function. Reparameterization refers to converting a multi-channel model into a single-channel model. That means the training phase of RepVGG is transformed into the inference phase. Next, we will introduce the process from the first to the second step. The first step mainly includes the merging of convolutional layers and BN layers. We introduce the merging process of the convolutional layer and the BN layer by

$$Conv(x) = W(x) + b \tag{2.11}$$

$$BN(x) = \gamma * \frac{(x - mean)}{\sqrt{var}} + \beta \tag{2.12}$$

Equation (1) is the convolution layer formula, and eq. (2) is the BN layer formula. Next, we need to combine the convolution layer and the BN layer.

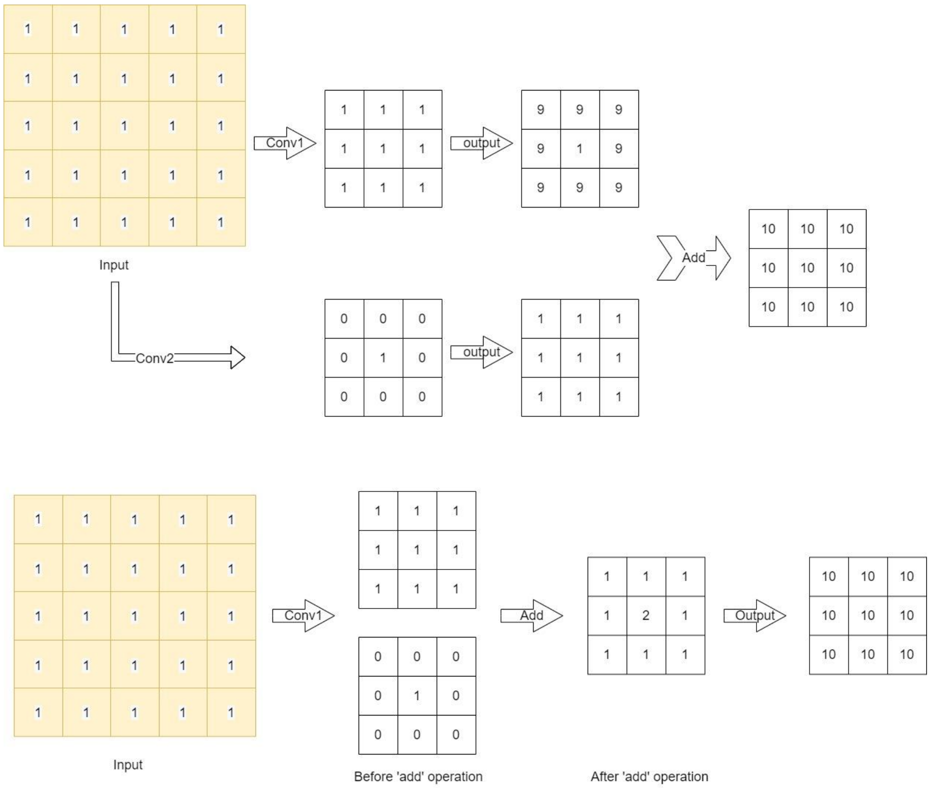


Figure 2.12: The additivity principle of convolution

$$BN\big(Conv(x)\big) = \gamma * \frac{W(x) + b - mean}{\sqrt{var}} + \beta \tag{2.13}$$

Equation (3) is a convolutional layer with weights as BN parameters. However, the $W(x)$ and b parameters have changed in the original convolution as follows:

$$W_{fused} = \frac{\gamma * W}{\sqrt{var}} \tag{2.14}$$

$$B_{fused} = \frac{\gamma * (b - mean)}{\sqrt{var}} + \beta \tag{2.15}$$

The final fusion result is,

$$\hat{\sigma} = BN\big(Conv(x)\big) = W_{fused}(x) + B_{fused} \tag{2.15}$$

Layer merging can effectively improve network performance. The next step is to introduce the conversion from the second to the third step. The three groups of 3×3 convolutional structures are turned into one group of 3×3 convolutional structures.

As shown in Fig. 3, the first way we perform convolution is to obtain the result and then perform the “add” operation. In a second way, we add the values of the convolution kernel to obtain a new convolution kernel and perform the convolution, the obtained result is the same. The additivity principle is applied to YOLOv6. It is able to simplify the convolutional operations that we only need to add the values of three 3x3 convolution kernels in the process of step 3.

To further improve the loss function, YOLOv6 adopts the SIoU bounding box regression loss function to supervise the learning of the network. Common bounding box regression losses include IoU, GIoU, CIoU, DIoU loss, etc. None of these methods consider the matching of the direction between the predicted box and the target box. The SIoU loss function redefines the distance loss by introducing the vector angle between the required regressions, which effectively reduces the degree of freedom of the regression, accelerates the network convergence, and further improves the regression accuracy.

2.5.5 Recognition based on Transformer

Sign language recognition has been a hot topic for researchers over the past decades (Bauer, Hienz, & Kraiss, 2000). In recent years, using transformers to detect objects has become a mainstream methodology. One of them is the Vision Transformer (ViT) (Dosovitskiy et al., 2020). Firstly, ViT divides the image into a grid of squares, then flattens each square into a single vector by concatenating all pixel channels in a square. The transformer is independent of the structure of the input elements, so they add learnable positional embeddings to each square, enabling the model to learn about the image structure. The feature maps are identical in the top layers in deep ViT models, they proposed a Re-attention method to enhance the feature map. As a result, the Top-1 accuracy can be improved by 1.6% on ImageNet (Zhou, Kang, Jin, Yang, Lian, Jiang, Hou, & Feng, 2021).

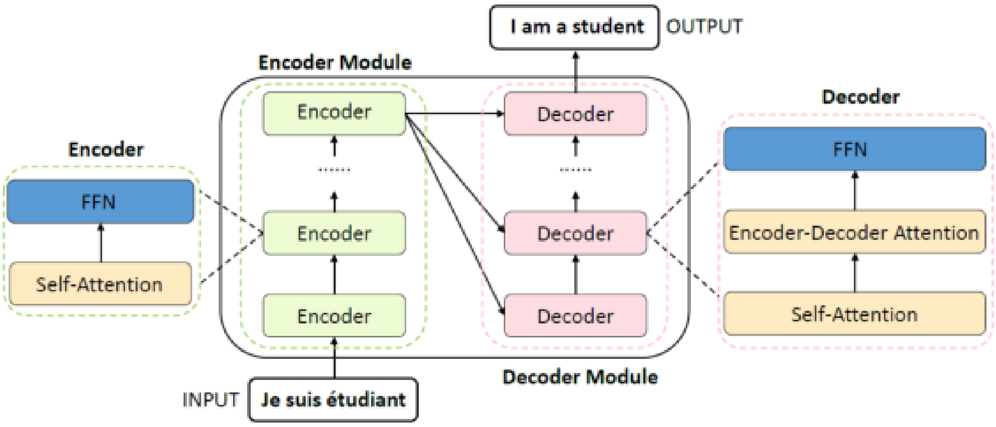


Figure 2.13: The process of FPN + PAN

Although Vison Transformer exhibits high recognition accuracy and recognition rate in object recognition, it requires hundreds of pre-training on a very large dataset to achieve the expected performance. This means that the Vison Transformer requires a lot of computing resources. In addition, Vison Transformer has strict requirements for the quality of the dataset. These conditions largely limit the further application of the Vison Transformer. The amount of computation is largely reflected in the computational complexity and the square of the token. Because the number and size of tokens always remain the same during the calculation.

In the subsequent improvement, the researchers refer to the ResNet structure and the feature pyramid structure for optimization, so that the number of tokens is continuously reduced when the number of layers is higher. Use the local window self-attention method to self-attention a part of the feature map. Replace fully connected layers with convolutions to reduce parameters. The feature maps of K, V are pooled in the process of generating Q, K, V to reduce computational parameters and reduce complexity. Another disadvantage is that, compared to CNNs and RNNs, Transformers have no spatial invariance assumption and cannot use convolutional kernel sliding windows to process the entire feature map. Therefore, Transformer needs more data to learn these assumptions, thus increasing the computational complexity. On the other hand, Transformer has the problem of transition smoothness, and the similarity between blocks increases as the model goes deeper. Furthermore, the model cannot encode position, and algorithms need to be designed to add position encoding. Data-efficient image Transformers (DeiT) do not require massive pre-training data and only rely on ImageNet data to achieve SOTA results while relying on fewer training resources.

One of the reasons why Transformer requires huge computing power is that the model itself cannot encode the position. Transformer is different from CNN, which requires position embedding to encode the position information of tokens, mainly because self-attention is permutation-invariant, that is, disrupting the order of tokens in sequence will not change the result. If the location information of the patch is not provided to the model, the model needs to learn the puzzle through the semantics of the patches, which increases the learning cost. To solve this problem, fixed position coding has been used in DETR (Detection Transformer). Positional encoding is a two-dimensional positional encoding method proposed by Carion et al. (2020). The positional encoding is added to the encoder's self-attention and the decoder's multi-head attention, and object queries are also added to the decoder's two attentions.

The methods aimed to translate sign language videos into spoken language sentences have been proposed (Ko, Kim, Jung, & Cho, 2019). According to Yin and Read, they proposed the STMC-Transformer to improve the state-of-the-art by over 7 BLEU on the video-to-text translation of the 2014T dataset (Yin, & Read, 2020). Camgoz et al. proposed the method using Connectionist Temporal Classification loss based on the novel transformer to have an end-to-end recognition and translation. They evaluated the performance based on the PHOENIX-Weather-2014 dataset and improved the performance from 9.58 to 21.80 BLEU-4 score (Camgoz, Koller, Hadfield, & Bowden, 2020). Rastgoo et al. proposed a method called Zero-Shot Sign Language Recognition (ZS-SLR), they used transformer for hand detection and Auto-Encoder (AE) on top of the Long Short-Term Memory. As a result, the proposed method showed better performance than compared methods on four datasets: RKS-PERSIANSIGN, First-Person, ASLVID, and isoGD (Rastgoo, Kiani, Escalera, & Sabokrou, 2021).

Besides, the methods of multimedia computing have been developed a lot. Bastanfard et al. have proposed a speech therapy system for hearing-impaired children (Bastanfard, Rezaei, Mottaghizadeh, & Fazel, 2010). Minoofam et al. proposed an adaptive reinforcement learning framework called RALF through Cellular Learning Automata (CLA) to produce content. The proposed tool has raised students' learning rate by almost 27% compared with the face-to-face approach. A new algorithm fb-kNN has been proposed for feature extraction (Bhatti, Huang, Wu, Zhang, Mehmood, & Han, 2019). Besides, an algorithm called spatial-spectral HSI classification has been proposed to extract more effective features (Bhatti, 2022) and the Clifford algebra algorithm for image processing (Bhatti, 2021).

Chapter 3
Methodology

In this chapter, we will show our research design and our proposed model. The entire structure will be presented in Section 3.3. The chapter mainly detailed each part of our research methodology.

3.1 Residual Network

Residuals refer to the difference between the actual observed value and the fitted value. In general, the deepening of the network can extract more features. In fact, as the network deepens to a certain extent, the problem of network degradation will occur if the network continues to deepen. The reason is that with the deepening of the network, the difficulty of training will continue to increase, and the difficulty of network optimization will continue to increase.

On the other hand, in the process of increasing the number of layers, the accuracy of the training data tends to be saturated. Continuing to increase the number of layers will lead to the problem of a decrease in the training accuracy. In a deeper network, the structural layer of the network model consists of an identity mapping layer and a nonlinear layer. The degradation problem shows how difficult it is to approximate the identity map through nonlinear layers. When the network degenerates, the shallow network can achieve a better training effect than the deep network. At this time, if we pass the features of the low layer to the high layer, the effect should be at least no worse than that of the shallow network, or if a VGG-100 The network is use of the same features as the 14th layer of VGG-16 in 98th layer, so the effect of VGG-100 should be the same as that of VGG-16.

Therefore, we can add a direct mapping (Identity Mapping) between layers 98 and 14 of VGG-100 to achieve this effect. From the perspective of information theory, due to the existence of DPI (Data Processing Inequality), in the process of forwarding transmission, with the deepening of the number of layers, the image information contained in the Feature Map will be reduced layer by layer, and the addition of the direct mapping of ResNet, it is guaranteed that the network of layers must contain more image information than the layers. Based on this idea of using direct mapping to connect different layers of the network directly, the residual network came into being.

ResNet is also called residual network. ResNet is constructed by Residual Building

Block. In ResNet, two mappings are proposed: Identity mapping, which refers to the curve marked with x on the right side; residual mapping (residual mapping)), the residual refers to $F(x)$ part. The final output is $F(x) + x$. The implementation of $F(x) + x$ can be achieved by a feedforward neural network with "shortcut connections". Shortcut connections are connections that skip one or more layers. The “weight layer” in the figure refers to the convolution operation. If the network has reached the optimal state and continues to deepen the network, the residual mapping will become 0, leaving only the identity mapping, so theoretically the network will always be in the optimal state, and the performance of the network will not decrease as the depth increases.

The residual path can be roughly divided into two types. One has a bottleneck structure, such as the 1×1 convolution layer in the right figure, which is used to reduce the dimension first and then increase the dimension, mainly for the practical consideration of reducing the computational complexity, which is called "bottleneck block", another structure without bottleneck, as shown on the left of the following figure, is called "basic block". The basic block is mainly composed of two 3×3 convolutional layers. The shallow networks ResNet18 and ResNet34 apply basic blocks. Another ResNet is based on Bottleneck, deep network ResNet50, ResNet101, ResNet152 apply Bottleneck. Each layer in ResNet is composed of several blocks stacked, and layers constitute the entire network. Each ResNet consists of four layers. Below we will describe the layers in front of the block.

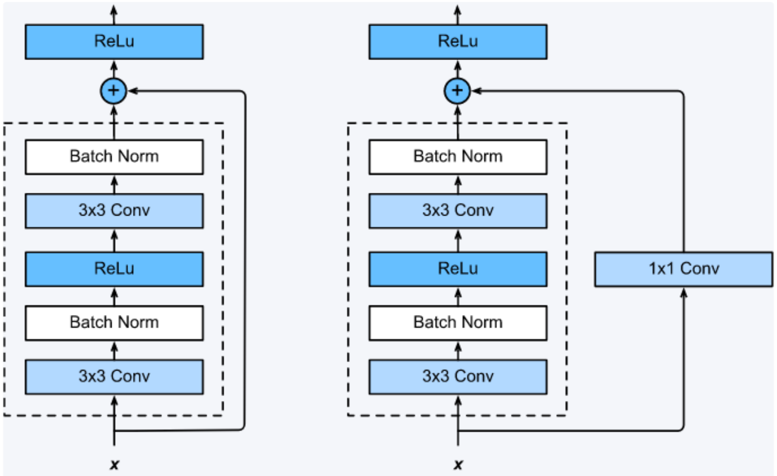


Figure 3.1: Two types of shortcuts

The shortcut path can also be roughly divided into two types, depending on whether the residual path has changed the number and size of feature maps. When the input and output dimensions are the same, the input x can be directly output as it is. But this cannot be directly added when the dimensions are inconsistent. There are two strategies. The first method is to use zero-padding to increase the dimension. First, a down-sampling is required, and the pooling method of *stride*=2 is used to avoid adding parameters. The second way is to use a new mapping, usually called the project shortcut. This method uses 1×1 convolution to up-scale or/and down-sampling. The main function is to keep the output in the same shape as the output of the $F(x)$ path. This method will increase the parameters and the amount of calculation.

The residual blocks that make up the residual network can be expressed as:

$$x_{l+1} = x_1 + \mathcal{F}(x_l, W_l) \tag{3.1}$$

The residual block is divided into two parts, the direct mapping part and the residual part. $h(x_l)$ is a direct mapping, which is reflected in the curve on the left in Figure 3.3. $\mathcal{F}(x_l, W_l)$ is the residual part, which is generally composed of two or three convolution operations, that is, the part that contains convolution on the right side of the figure.

In the convolutional network, when the number of Feature Maps of x and y is different, it is necessary to use convolution to increase or reduce the dimension. At this time, the residual block is expressed,

$$\begin{gathered} \mathcal{Y}_l = h(x_l) + \mathcal{F}(x_l, W_l) \\ x_{l+1} = f(\mathcal{Y}_l) \end{gathered} \tag{3.2}$$

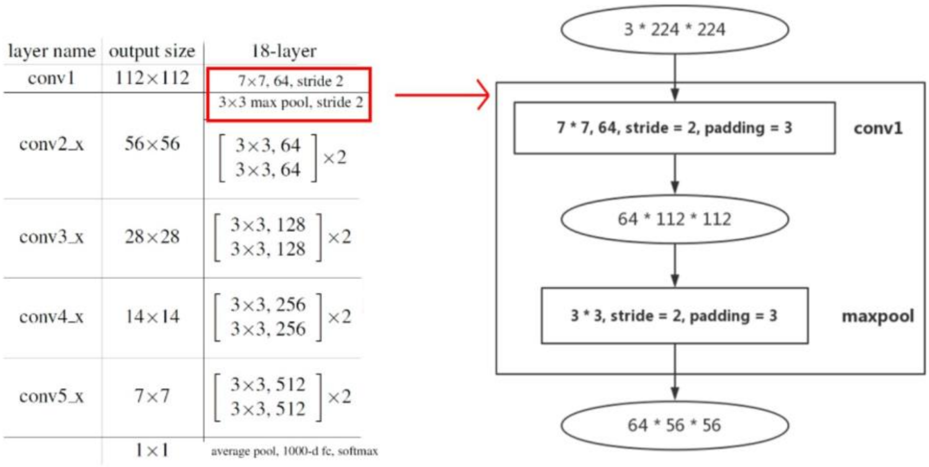
x_l and x_{l+1} respectively represent the input and output of the l residual unit, and each residual unit generally contains a multi-layer structure. $\mathcal{F}$ is the residual function, representing the learned residual, while $h(x_l) = x_l$ represents the identity map, and f is the ReLU activation function. Based on the above formula, we obtain the learning features from the shallow layer l to the deep layer L as:

$$x_L = x_l + \sum_{i=1}^{L-1} \mathcal{F}(x_i, W_i) \tag{3.3}$$

Using the chain rule, the gradient of the reverse process can be found:

$$\frac{\partial loss}{\partial x_l} = \frac{\partial loss}{\partial x_L} \cdot \frac{\partial x_L}{\partial x_l} = \frac{\partial loss}{\partial x_l} \cdot \left(1 + \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} \mathcal{F}(x_i, W_i)\right) = \\ \frac{\partial \varepsilon}{\partial x_L} + \frac{\partial \varepsilon}{\partial x_L} \frac{\partial}{\partial x_l} \sum_{i=1}^{L-1} \mathcal{F}(x_i, W_i)\, L \tag{3.4}$$

The loss function represented by the first factor $\frac{\partial loss}{\partial x_L}$ of the formula reaches the gradient of L. The ‘1’ in parentheses indicates that the short-circuit mechanism can propagate the gradient in a lossless way, while the other residual gradient needs to go through the layer with weights. The gradient is not directly passed over. The residual gradient will not always be -1, and even if it is small, the presence of 1 will not cause the gradient to disappear. This proves that the problem of vanishing gradients does not arise in residual networks. $\frac{\partial \varepsilon}{\partial x_L}$ means that the gradient of layer L can be directly passed to any layer l that is shallower than it. Information can be transmitted between high and low layers very smoothly, which allows residual networks to train deep models.



layer name	output size	18-layer
conv1	112×112	7×7, 64, stride 2
		3×3 max pool, stride 2
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$
	1×1	average pool, 1000-d fc, softmax

3 * 224 * 224
7 * 7, 64, stride = 2, padding = 3
conv1
64 * 112 * 112
3 * 3, stride = 2, padding = 3
maxpool
64 * 56 * 56

Figure 3.2: The layer of ResNet18/ResNet34

The input and output are represented by an ellipse, and the middle is the size of the input and output: Channel×height×width. The rectangular box refers to the convolutional layer or the pooling layer, such as “3×3, 64, *stride*=2, *padding*=3" means that the kernel size of the convolutional layer is 3×3, the number of output channels is 64, the step size is 2, and the padding is 3. The layer type represented by the rectangular box is marked on the right side of the box, such as "conv1".

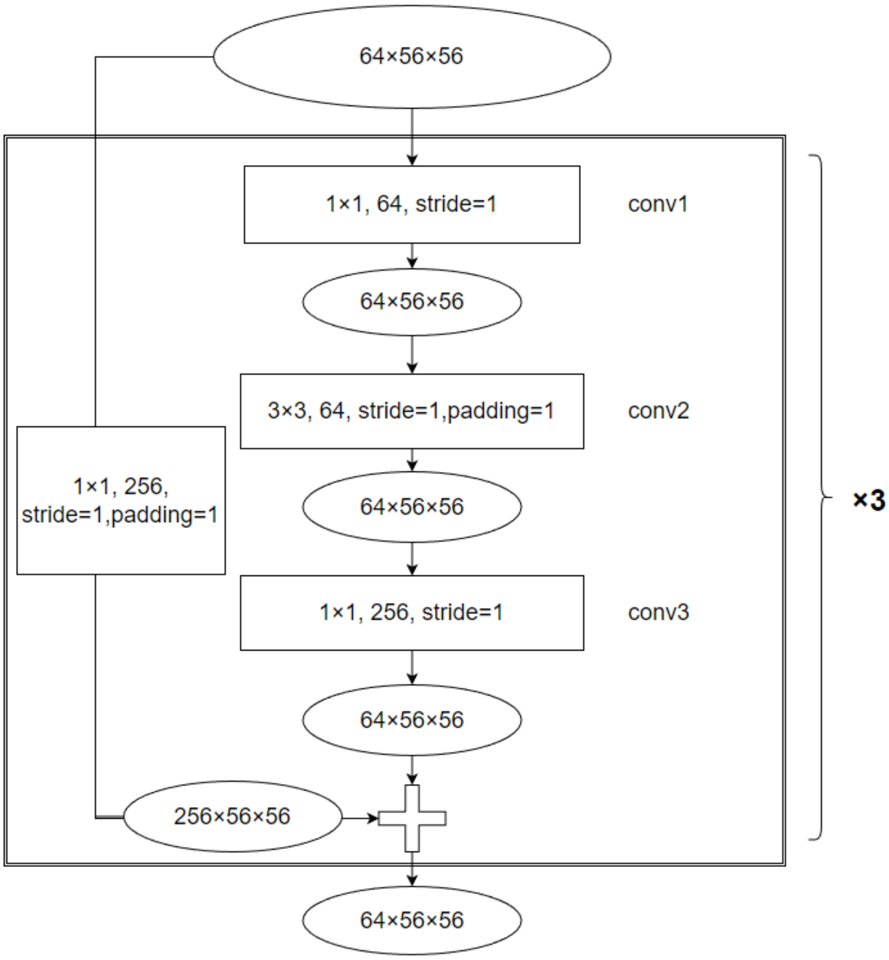


Figure 3.3: The layer of ResNet152

The difference from Basic block is that each bottleneck adds a convolutional layer between the input and output, but there is no down-sampling in layer1, which is the same as the basic block. As for the reason why the convolution layer must be added, bottleneck's conv3 will expand the number of input channels to 4 times the original, resulting in the input and output sizes must be different. The three-block structures in the layer are exactly the same, so “3×3” is used. After the input with the size of 256×56×56 enters the first block of Layer2, the number of channels must be reduced first through Conv1, and then Conv2 is responsible for reducing the size. At the output, due to the change in size, the input needs to be down-sampled, which is also implemented through a 1×1 convolutional layer with *stride*=2. The next three blocks do not need to be down-sampled.

In summary, ResNet solves the degradation problem of deep networks through residual learning, allowing us to train deeper networks.

3.2 Training Data

In this thesis, we use our own dataset for model training and testing to get the stellar performance of experimental results. There are 8,600 frames in total, and 6,450 frames were selected for the training section; 2,150 frames were picked up for the testing section. Another dataset contains twelve video fragments of nine classes with the labels: "Love", 'Good", "You", "Meet", "Yes", "No", "Please", "Name", "My", all these images of sign language gestures were collected by ourselves. The total number of images is 7,192 in this dataset; 5,000 frames were employed for the model training, 2,192 frames were picked for the testing. Fig. 5 shows the gesture samples for the nine classes.

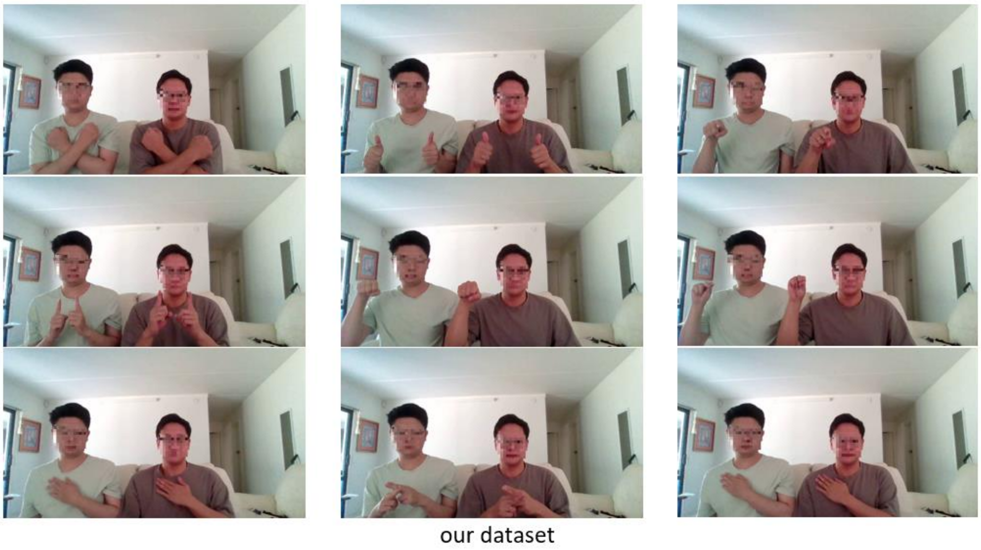


Figure 3.4: Visual samples of our home-grown sign language dataset

As shown in Figure 3.4, we use a dataset we made ourselves. The image above shows nine types of sign language action images that we randomly selected. In this dataset, in order to achieve better sign language recognition, we use a data augmentation method called Mosaic. This method uses sequential acquisition to acquire images. Data augmentation can effectively improve the efficiency of data training.

3.3 Research Design

In this thesis, we adopt ResNet152 and feature pyramid as backbone. ResNet can effectively solve the problem of deep network degradation. The feature pyramid can solve the problem that small targets cannot be effectively recognized and improve the accuracy of small target recognition. Based on this, we added Detection Transformer (DETR) for object detection and classification. Transformer-based object detection can perform parallel semantic analysis, which greatly improves detection efficiency.

The first introduction is about the transformer. The Detection Transformer used in this thesis has two key parts. The first is to use the encoder-decoder architecture of the transformer to generate N box predictions at once. Where N is a preset integer and is much larger than the number of objects in the image. The second is to design a bipartite matching loss, which calculates the size of the loss based on the bipartite graph matching of the predicted boxes and ground truth boxes. Thus, the position and category of the predicted box are closer to the ground truth.

The second thing to talk about is the feature pyramid. Feature pyramids are mainly used in object detection, semantic segmentation and behavior recognition. It can greatly improve the performance of the model. After the targets of different sizes have undergone the same down-sampling ratio, there will be a large semantic generation gap, and the most common manifestation is that the detection accuracy of small targets is relatively low. The feature pyramid has the characteristics of different resolutions at different scales. Objects of different sizes can have appropriate feature representations at the corresponding scales. By fusing multi-scale information, objects of different sizes can be predicted at different scales. It greatly improves the performance of the model.

We combine ResNet152, feature pyramid and transformer, which can not only effectively identify targets of different scales, but also further improve the recognition accuracy and detection efficiency. Below we show the overall structure diagram of the proposed model.

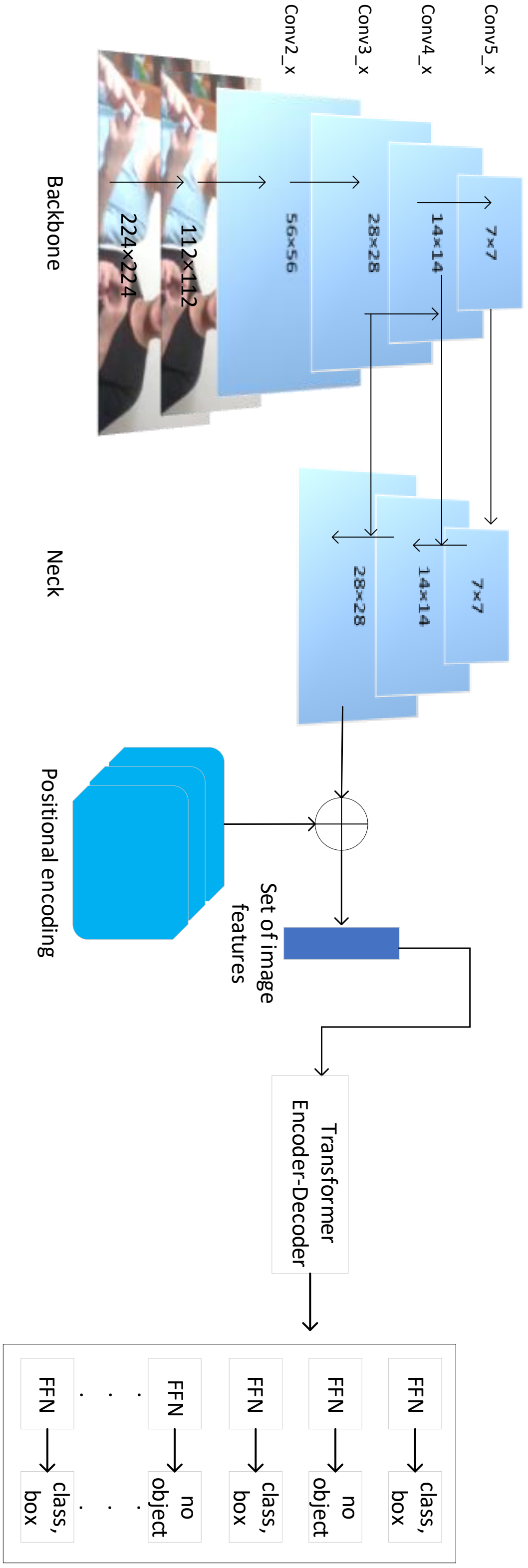


Figure 3.5: The structure of ResNet152+FPN+DETR

3.3.1 ResNet152

In order to improve the accuracy and speed of the proposed methodology for sign language recognition, in this thesis, we make use of ResNet152 to replace ResNet50 as a backbone. Its aim is to increase convolutional layers to improve the feature map. As shown in Fig. 2, the backbone makes use of the improved ResNet152 network. The function of FPN structure is to enhance the feature maps for each scale of the network as the neck part before data processing.

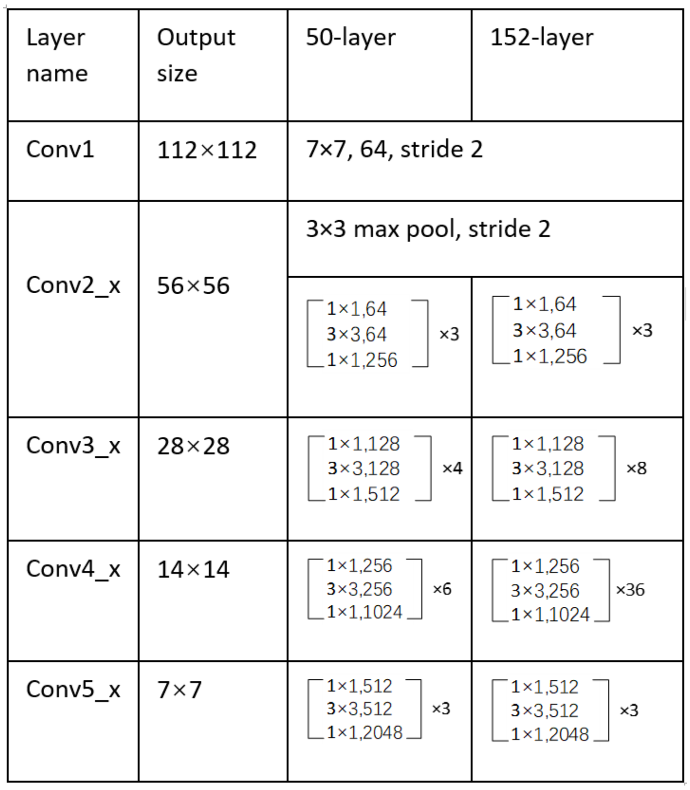
ResNet152 has two basic blocks, called Conv Block and Identity Block. The functionality of Conv Block is to change the dimension of the network. The input dimension and output dimension of Identity Block. The dimensions are the same and can be connected in series to deepen the net. As shown in Fig. 3, ResNet152 is based on ResNet50; the difference between ResNet152 and ResNet50 is that ResNet152 has 36 blocks and ResNet50 has 6 blocks. Thus, ResNet152 can get better results. Equation (3.5) is used to calculate the size of the feature map,

$$w' = \frac{w + 2p - k}{s} + 1 \tag{3.5}$$

where w is the size of convolution input matrix, k is the convolution kernel size, s is the length of convolution steps, and p is the padding. The size of input images in this article is 224×224 pixels. After down-sampling convolutions, multiple 1×1 convolutions and 3×3 convolutions, the scales of output feature maps are 7×7, 14×14, 28×28, 56×56 which are calculated as eq.(3.6),

$$S_{(i,j)} = (X \times V) \sum_{M} \sum_{N} x\,(i + m, j + n) v(m, n) \tag{3.6}$$

where x is the variable of the input image, v is the convolution kernel, $M \times N$ is the size of the input image. Compared with ResNet50, ResNet152 has more convolution blocks and convolution kernels. The semantic information and location information of multi-scale features are output to the neck and enhance object detection accuracy.



Layer name	Output size	50-layer	152-layer
Conv1	112×112	7×7, 64, stride 2	
Conv2_x	56×56	3×3 max pool, stride 2	
		$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
Conv3_x	28×28	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times8$
Conv4_x	14×14	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times36$
Conv5_x	7×7	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$

Figure 3.6: The structure of ResNet152

Pooling operations are also known as subsampling or downsampling. The pooling operation usually occurs after the convolutional layer, and the dimension of the output features of the convolutional layer is reduced by the pooling layer. The purpose is to reduce network parameters and prevent overfitting. Figure 3.3 shows the operations of Max Pooling with 2×2 matrix. After the pooling operation, the image is condensed greatly.

3.3.2 Feature Pyramid Network

Lin et al proposed Feature Pyramid Network, the main problem solved in the work is the insufficiency of target detection in dealing with multi-scale changes (Lin et al., 2017). The main problem solved in this thesis is the insufficiency of target detection in dealing with multi-scale changes. Many networks now use a single high-level feature. For example, Faster R-CNN uses a four-fold down-sampling convolutional layer - Conv4, for subsequent object classification and bounding box regression. However, this method has an obvious defect, that is, the small object itself has less pixel information, which is easily lost in the process of down-sampling. The pyramid method is used for multi-scale change enhancement, but this will bring a great amount of calculation. Therefore, in this thesis, we propose a network structure of feature pyramid, which can handle the multi-scale change problem in object detection with a minimal increase in the amount of computation.

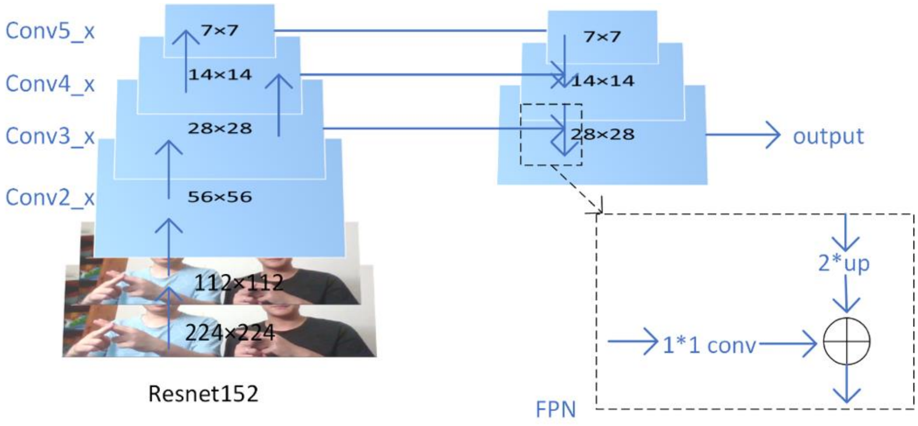


Figure 3.7: The structure of ResNet152 + FPN

This network structure can fuse feature maps with strong low-resolution semantic information and feature maps with weak high-resolution semantic information but rich spatial information under the premise of increasing less computation. In fact, before this thesis, it was also mentioned that a feature map with both high resolution and strong semantic information was obtained for prediction, but the uniqueness of FPN is that it is based on the feature pyramid. The feature maps of each level are predicted separately.

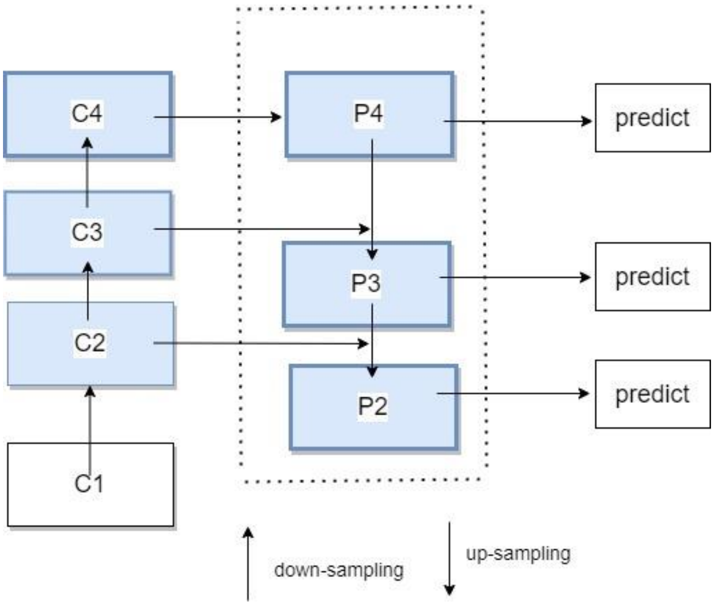
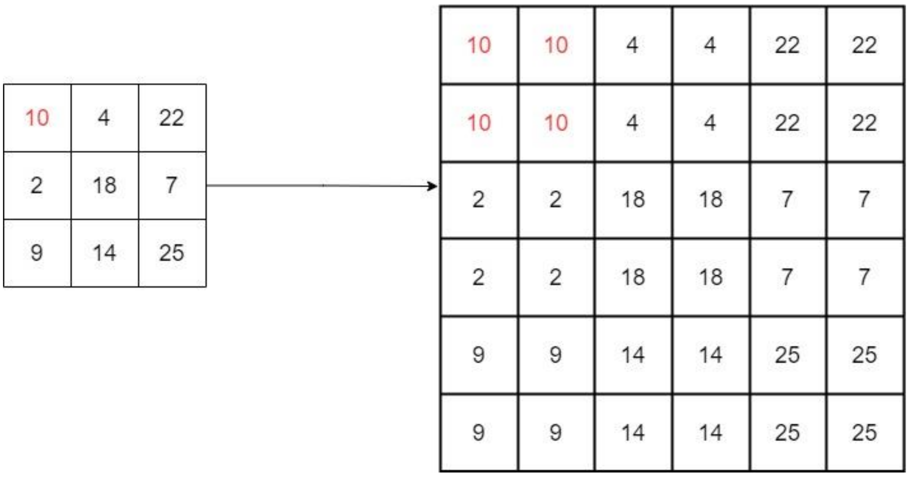


Figure 3.8: Feature Pyramid Network (FPN)

As part of the feedforward backbone, each stage is downsampled with *step*=2. The network part with the same output size is called a stage, and the feature map of the last layer of each stage is selected as the number of layers corresponding to the Up-bottom path, and the reference of element add after 1×1 convolution. For example, as shown in Figure 3.9 , the ResNet in the left column uses the output of the last Residual Block of each layer, denoted as {C1, C2, C3, C4}. FPN is use of 2~5 layers to participate in prediction, because the semantics of the first layer is still too low. {C2, C3, C4} indicates that the output layers of conv2, conv3 and conv4 are FPN features. The downsampling multiples of the corresponding input images are {4, 8, 16, 32}.

The top-down process up-sampling the small feature maps of the top layer. Zoom in to the same size as the feature map of the previous stage. As shown in Figure 3.10, in the feature pyramid network, the method of up-sampling applies nearest neighbor interpolation.



10	4	22
2	18	7
9	14	25

10	10	4	4	22	22
10	10	4	4	22	22
2	2	18	18	7	7
2	2	18	18	7	7
9	9	14	14	25	25
9	9	14	14	25	25

Figure 3.9: Nearest neighbor Interpolation

Using the nearest neighbor interpolation method, the semantic information of the feature map can be preserved to the greatest extent during the upsampling process, so that it can be fused with the corresponding feature map with rich spatial information in the bottom-up process to obtain a strong semantic feature map. Spatial information has feature maps with strong semantic information. This approach is beneficial for both classification and localization.

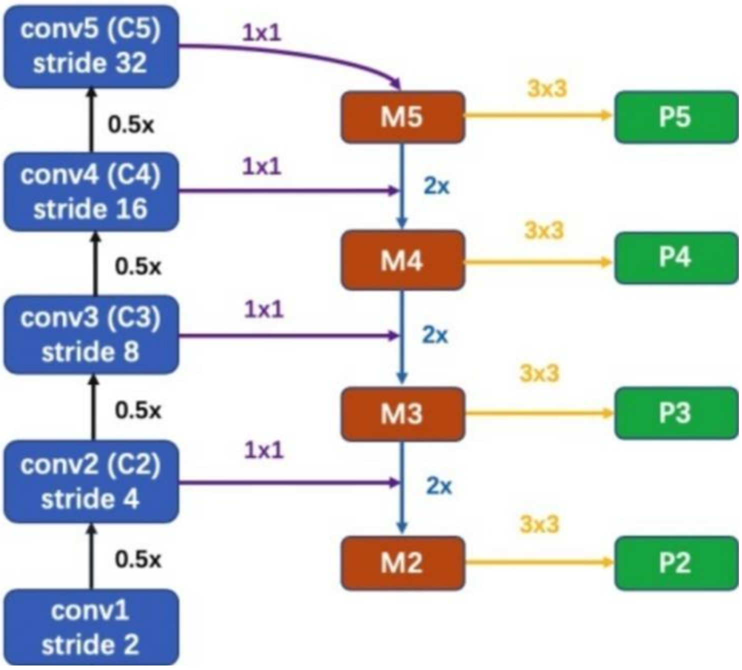


Figure 3.10: Layered overlay of feature maps

The specific process is as follows: C5 layer firstly undergoes 1 x 1 convolution to change the number of channels of the feature map. According to the principle of direct addition of elements at the same position in the feature map, M5 is obtained by adding up-sampling and the feature map of C4 after 1 x 1 convolution to obtain M4. This process is done twice to get M3 and M2 respectively. The M-layer feature map is convolved 3 x 3 to reduce the aliasing effect caused by the nearest neighbor interpolation, and the final P2, P3, P4, and P5 layer features are obtained.

3.3.3 Transformer

In DETR, the size of $3\times H_0\times W_0$ image is first processed with the CNN backbone to obtain the size of $C\times H\times W$ feature map. Then add the feature map output by the backbone and the position encoding, input it into the Transformer Encoder for processing, and get the image embedding for input to the Transformer Decoder.

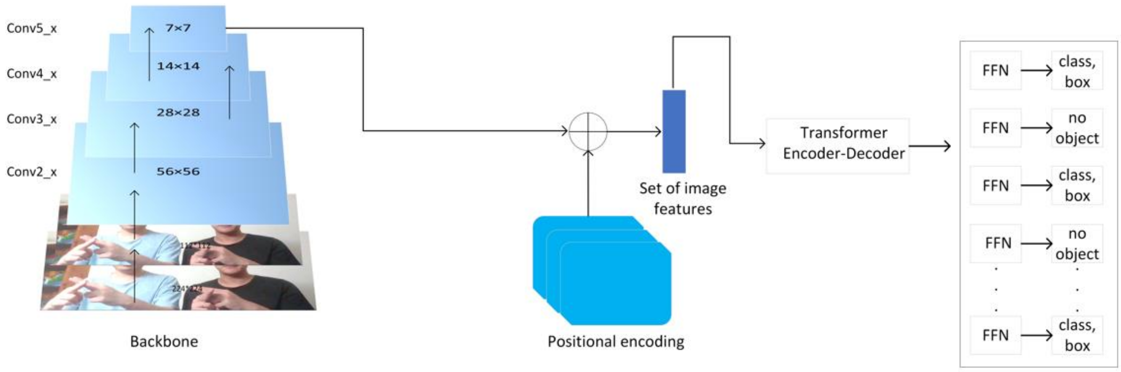


Figure 3.11: The structure of DETR

Next, we will introduce the process of converting the feature map output by the backbone into serialized data that can be processed by the transformer encoder. The process is divided into three steps. The first step is to process the feature map of size C×H×W output by the backbone with 1×1 convolution. In addition, the number of channels is compressed from C to d and a new feature map of $d{\times}H{\times}W$ is obtained. The second step is to compress the dimension of the space into one dimension and reshape the $d{\times}H{\times}W$ feature map obtained in the previous step into a $d{\times}H{\times}W$ feature map. The third step is to add position encoding to the feature map generated in the previous step to reflect the position information. Since the transformer model is order-independent, and the dimension in the feature map generated in the previous step is obviously related to the position of the original image, it is necessary to add position encoding to reflect the position information.

The next step is the transformer decoder. The decoder is similar to the standard transformer structure, with the difference that our model decodes N targets in parallel at each decoding layer. Since the decoder is also ordered invariant, the embeddings of the N inputs must be different to generate different results. These learnable positional encodings are called object queries and take as input embeddings, add them to each attention layer, and pass through the decoder to complete the output. Then they are sent to the FFN layer for independent decoding into box coordinates and class labels and generate N predictions. Using auto-encoder-decoder attention to act on these embeddings enables global inference of their pairwise relationships, using the entire picture as contextual information.

Initialization of object queries and their corresponding positional embedding. The object query loads the object information on the picture. Before entering the decoder, the model actually knows nothing about the objects in the picture, so it is initialized to 0. Positional embedding loads the position and area concerned by each query. These 100 queries can cover the entire picture as evenly as possible, so random initialization is used.

DETR has two core advantages. The first is the end-to-end detection. Anchor-based object detectors mostly use a one-to-many label assignment algorithm, so NMS becomes an essential post-processing step that removes redundant boxes. For DETR, end-to-end detection is particularly natural and straightforward. In addition to one-to-one bipartite matching, the Transformer mechanism introduces information exchange between queries and adds a self-attention mechanism to the decoder to prevent multiple queries from converging to the same target.

The second is to decouple the input and output spaces. In the logic of the Transformer, the picture is expanded into a one-dimensional sequence, and the absolute position information is described by the positional embedding to maintain the picture form. The encoder uses one set of positional embedding, and the decoder uses another set of positional embedding. This actually gives the model the ability to decouple the input and output spaces. For example, the input space is uniformly sampled points on the image (stride=32), and the output space is 100 randomly distributed points on the image. Due to the Attention mechanism, the query obtains the ability of global receptive field and information exchange between samples, achieving sparse sampling and end-to-end detection.

3.4 Loss function

In this section, we will combine CNN and RNN together to construct a DNN About the loss function, the output of Transformer is N predictions of visual object classes, where N is larger than the number of visual objects. The annotation of the dataset consists of two parts: One is c_i representing the class of the visual object, the other is b_i which shows the bounding box of the object. The prediction probability is $\hat{p}_{\hat{\sigma}(i)}(c_i)$. In Fig. 3.10, N=5 is set as an example. This satisfies equation (3.7), and the loss function of this optimization is calculated by using equation (3.8).

$$\hat{\sigma} = argmin_{\sigma} \sum_{i}^{N} \mathcal{L}_{match}\,(y_i, \hat{y}_{\sigma(i)}) \tag{3.7}$$

$$\mathcal{L}_{Hungarian}(y, \hat{y}) = \sum_{i=1}^{N}[-log\hat{p}_{\hat{\sigma}(i)}(c_i) + 1_{\{c_i \neq \emptyset\}}\mathcal{L}_{box}(b_i, \hat{b}_{\hat{\sigma}}(i))] \tag{3.8}$$

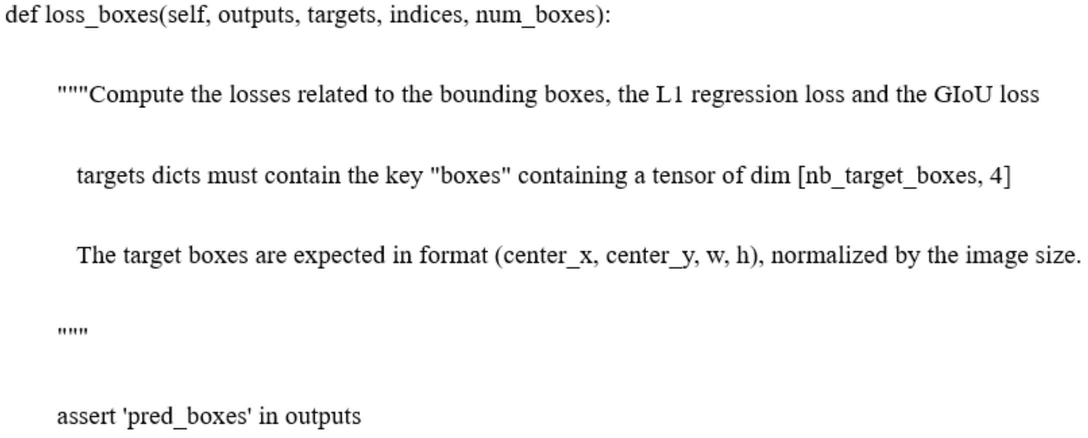
where $\hat{\sigma}$ is the optimal match obtained in equation (3.7). Similar to the setting of faster-RCNN for the weight of negative samples, when $c_i = \emptyset$, the weight is one-tenth of the original. The matching loss of target to $\emptyset$ does not depend on the predicted value and is therefore a constant. In matching loss, we use probability instead of log probability. This is to balance the loss of class prediction with box prediction, which we found to be better.

3.4.1 Bounding Box Loss

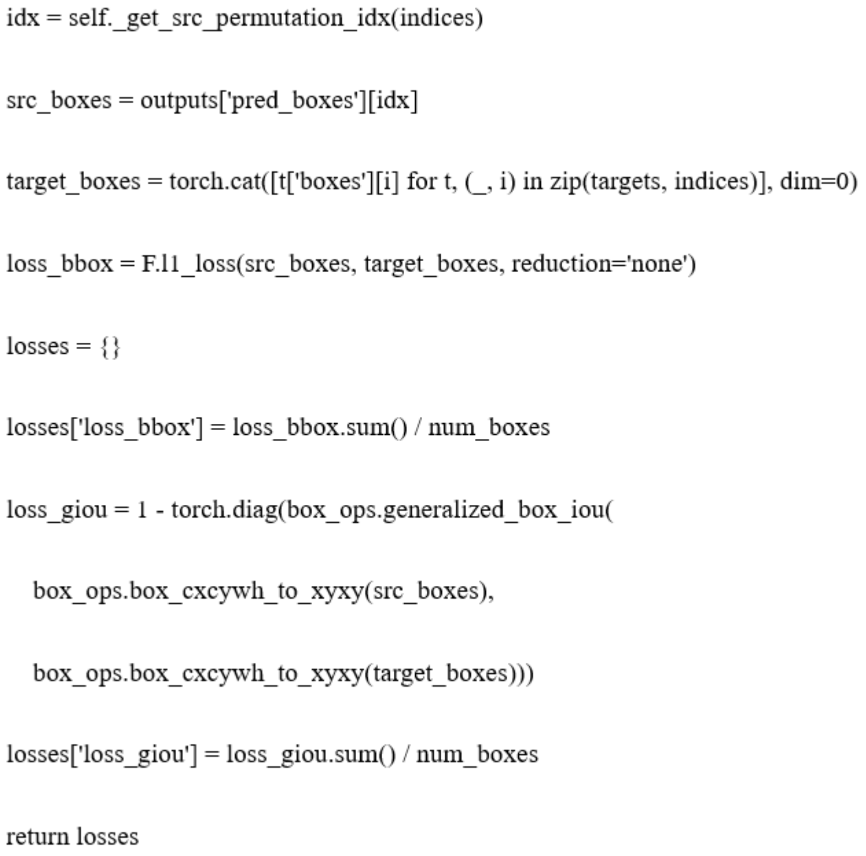
Since the method used in the article does not have a pre-designed anchor and directly predicts the bounding box, if the L1 loss is directly calculated like other methods, it will lead to inconsistent penalties for large boxes and small boxes. Therefore, while using L_1 loss, the article also uses scale invariant IoU loss.

$$\mathcal{L}_{box}(b_i, \hat{b}_{\sigma(i)}) = \lambda_{iou}\mathcal{L}_{iou}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\mathcal{L}_1}\,||b_i - \hat{b}_{\sigma(i)}||_1 \tag{3.9}$$

The specific implementation of the feed function is as follows,



```
def loss_boxes(self, outputs, targets, indices, num_boxes):

    """Compute the losses related to the bounding boxes, the L1 regression loss and the GIoU loss

      targets dicts must contain the key "boxes" containing a tensor of dim [nb_target_boxes, 4]

      The target boxes are expected in format (center_x, center_y, w, h), normalized by the image size.

    """

    assert 'pred_boxes' in outputs
```



```
idx = self._get_src_permutation_idx(indices)

src_boxes = outputs['pred_boxes'][idx]

target_boxes = torch.cat([t['boxes'][i] for t, (_, i) in zip(targets, indices)], dim=0)

loss_bbox = F.l1_loss(src_boxes, target_boxes, reduction='none')

losses = {}

losses['loss_bbox'] = loss_bbox.sum() / num_boxes

loss_giou = 1 - torch.diag(box_ops.generalized_box_iou(

    box_ops.box_cxcywh_to_xyxy(src_boxes),

    box_ops.box_cxcywh_to_xyxy(target_boxes)))

losses['loss_giou'] = loss_giou.sum() / num_boxes

return losses
```

3.5 Evaluation Method

Apropos the evaluations, the metrics for evaluating our model are AP (Average Precision) and FPS (Frames Per Second). Regarding multiclass object detection, an introduction is employed for calculating the evaluation parameters: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN). As shown in Table 1, all the experimental indexes will be calculated separately for AP, Recall, and Precision. Precision (precision rate) is the proportion of true examples that should be predicted as positive, calculated by using eq. (3.9). As shown in eq. (3.10), TP+FN is the number of all positive samples, Recall (recall rate) is the proportion of all positive samples that are correctly predicted. In practical applications, the average precision value is calculated using eq. (3.11).

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Boxes_{pre}} \tag{3.9}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Boxes_{gt}} \tag{3.10}$$

$$AP = \frac{TP + TN}{TP + TN + FP} \tag{3.11}$$

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{3.12}$$

In eq. (3.9), a represents the number of all prediction boxes. B represents the number of all ground truth. However, using only Precision or Recall has limitations and cannot fully reflect the effect of object detection. The reason is that the setting of the threshold can seriously affect the evaluation results. In the target detection network, the confidence of the target and the threshold of IOU are important factors for screening the final target frame. When the threshold is set to a low value, many negative samples are left to be classified as positive samples. At this time, Precision will decrease, and Recall will increase. On the contrary, when the threshold is set to a high value, the screening of the target is stricter, and the samples with low confidence but which are positive samples will be filtered out. At this time, the Precision will increase, and the Recall will decrease. Therefore, we need a comprehensive evaluation index to balance the effects of Precision and Recall on target detection. AP and mAP are used to solve this problem.

AP represents the accuracy of a class of objects on all maps. For example, given a class *C*, *AP* is equal to the sum of the accuracy values for all images in the validation set for class *C* divided by the number of target images containing class *C*. In eq. (3.12), *C* is the total number of classes and AP_i is *AP* value of class *i*. *mAP* represents the accuracy of all classes on all graphs. Assuming that there are 20 classes in a set, we use *AP* to calculate the accuracy of each class. The sum of the accuracies for all classes divided by the number of classes is the mean average precision. Using *mAP* as the evaluation criterion can show the overall advantages of the model.

Chapter 4
Results, Analysis and Discussions

The main content of this chapter is to show the result after using our home-grown dataset. Besides, our proposed model and tested models will be compared based on the experiment results. In addition, experimental results are analyzed and compared, two evaluation methods will be used to compare our model with test models. Besides, we will also discuss the limitations of the project.

4.1 Data Collection and Experimental Environment

The evaluation methods in the previous sections are *AP* (average precision) and *FPS* (frames per second). In Table 4.1, *TP*, *TN*, *FP*, and *FN* are mainly employed to count two types of classification problems, and multiple classes can also be counted separately. The samples are split into positive samples and negative samples. The first letters in *TP*, *TN*, *FP*, and *FN* indicate whether the recognition result of the classifier is correct. The focus of this thesis is mainly on the proposed deep learning methods based on DETR and its impact on the result. We mainly emphasized four state-of-the-art backbones to fulfill the sign language recognition, which are ResNet34, ResNet50, ResNet101, and ResNet152+FPN. Throughout our experiments, we made use of multiple deep learning methods to compare our experimental results. The deep learning models with the feature pyramid networks are much more stable and robust in sign language recognition. In Table 4.1, we compare our deep learning models for sign language recognition using our dataset.

Table 4.1: The deep learning models for sign language recognition

Models	**APs**	**AP$_{50}$**	**AP$_{75}$**	**Param**	**F1**	**FPS**
ResNet18+DETR	30.6	49.6	29.7	40M	62.2	20
ResNet34+DETR	31.8	50.2	30.2	41M	63.0	21
ResNet50+DETR	32.5	51.0	29.3	50M	64.2	27
ResNet101 + DETR	33.3	51.8	29.8	62M	66.8	20
YOLOv3	31.3	52.5	30.6	66M	67.7	22
YOLOv4	31.7	52.8	31.8	65M	68.5	23
YOLOv5+Attention	32.4	53.9	31.5	68M	70.8	24
YOLOX + ViT	34.6	54.3	32.6	70M	71.6	26
ResNet152 + FPN +DETR (proposed)	**35**	**54.8**	**33.9**	**73M**	**72.2**	**28**

As shown in Table 4.1, compared with ResNet34, ResNet50, ResNet101 and YOLO series, our method ResNet152+FPN reaches the highest performance on Average Precision (AP) rating at 31.50% in our dataset. Comparative experiments show that the new method improves the detection accuracy by around 1.70% compared to DETR based on our dataset. The detection accuracy is higher than the standard DETR model in AP, AP_{50}, AP_{75}.

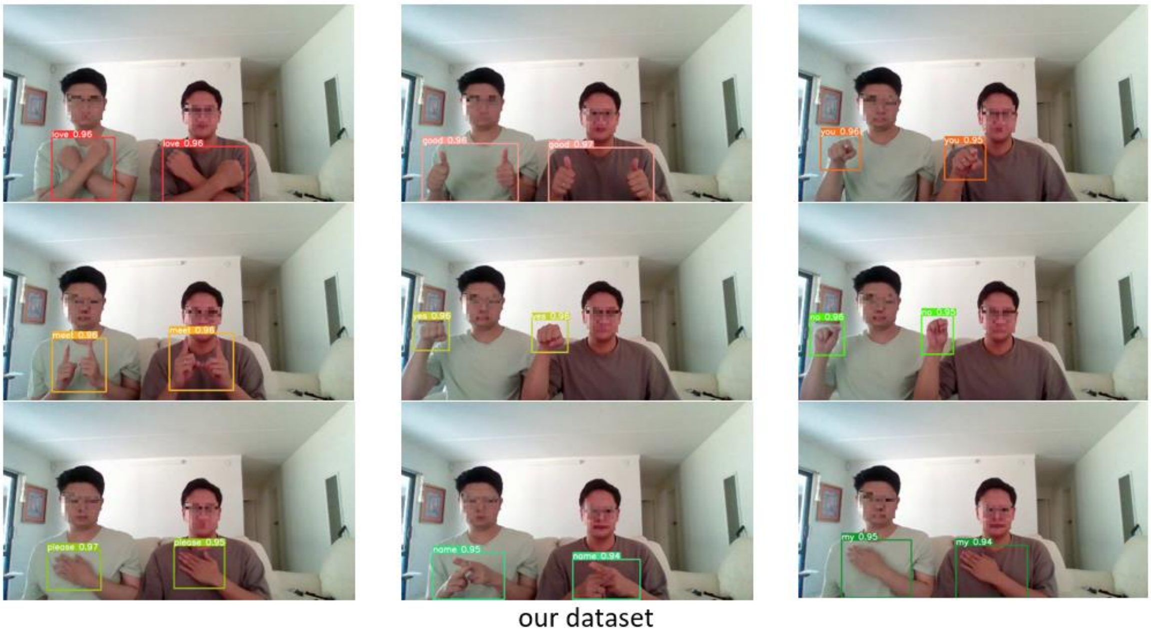
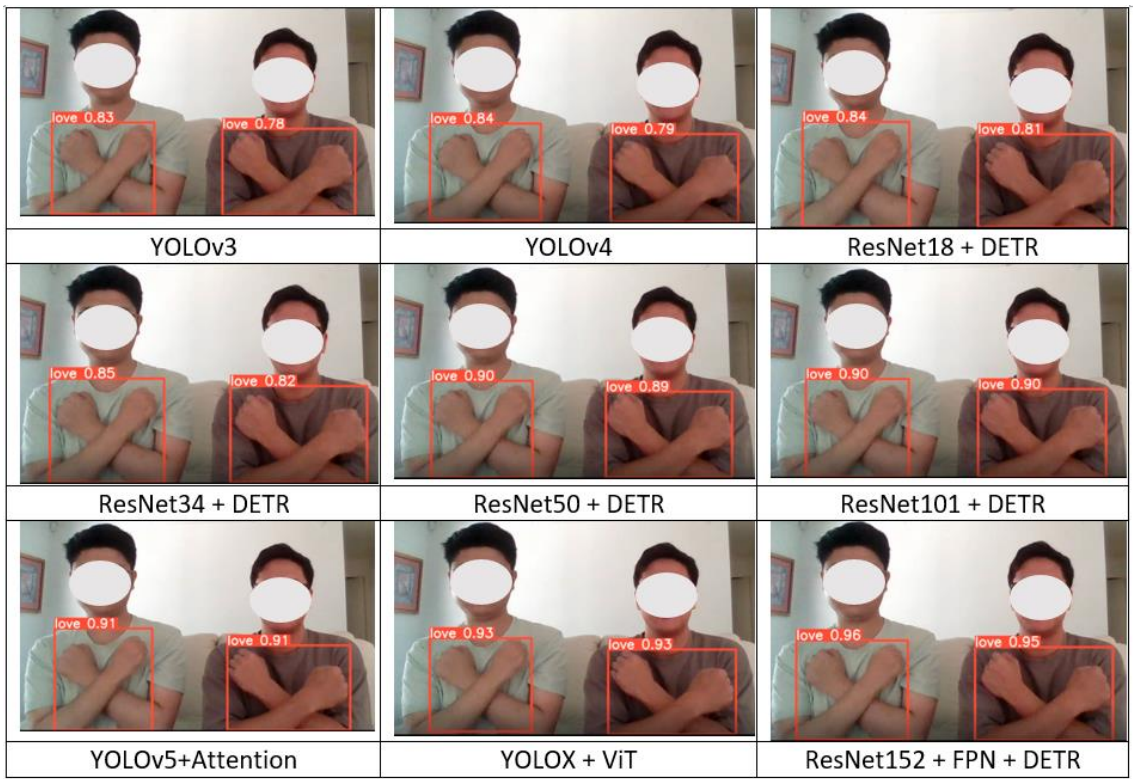


Figure 4.1: The results of sign language recognition

Fig. 4.1 shows our recognition results. We sampled nine sign language poses corresponding to the dataset in the previous section. The above results are obtained by the method proposed in this thesis. For our proposed method and all compared methods, we harnessed an RTX 3060 GPU and AMD Ryzen 5 5600H CPU to accelerate the training and detecting process to achieve efficiency. Next, we compare the accuracy of single sign language gestures under multiple models.



YOLOv3 | YOLOv4 | ResNet18 + DETR

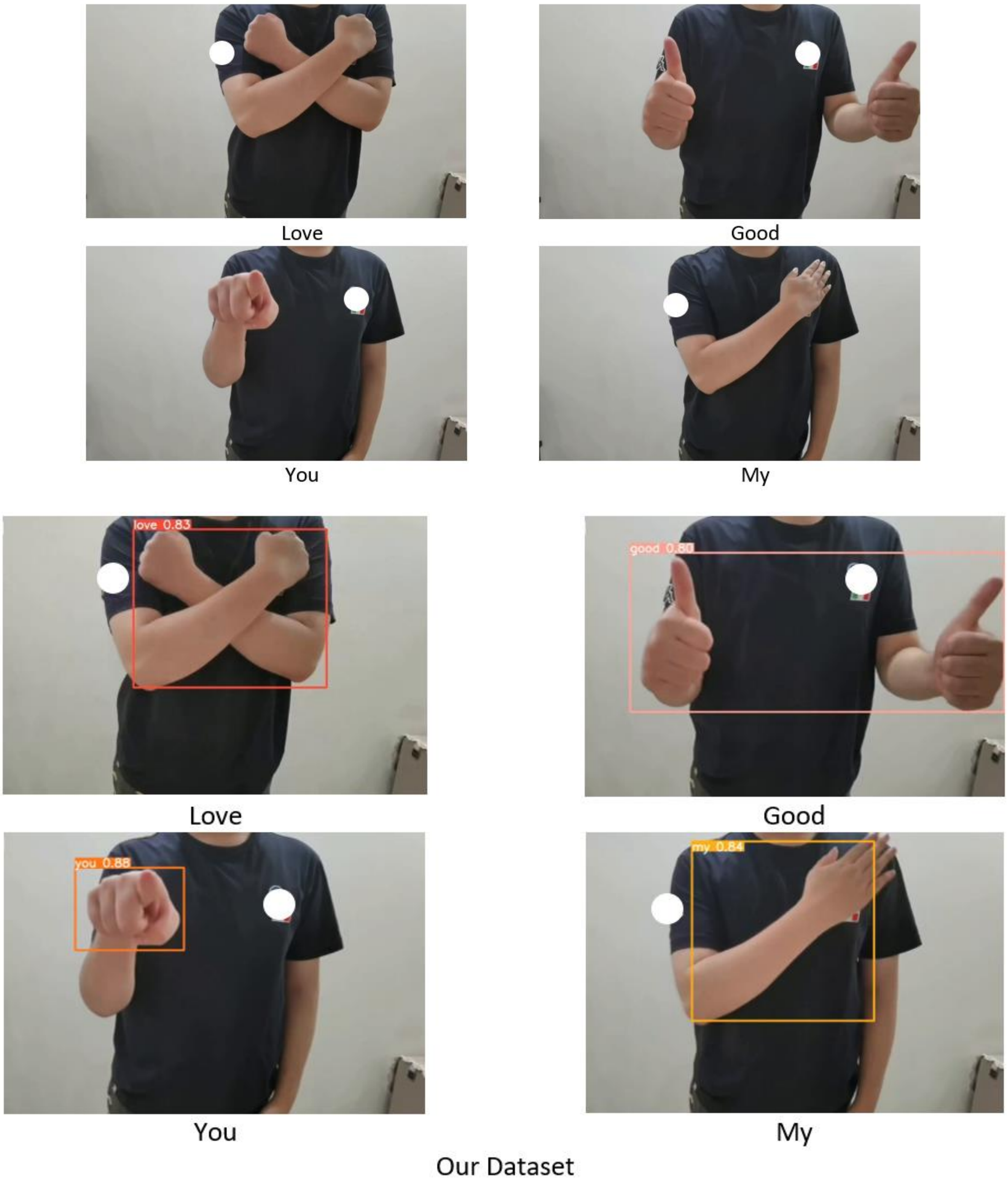
ResNet34 + DETR | ResNet50 + DETR | ResNet101 + DETR

YOLOv5+Attention | YOLOX + ViT | ResNet152 + FPN + DETR

Figure 4.2: Love gesture for tested models

In the experiments, we randomly sample a class from the dataset to test the experimental model. The experimental results are shown in the figure above. In the love class, our model achieves the highest accuracy of 96.45%. In the comparison of the models, with the increase in the number of ResNet layers, the recognition accuracy has been gradually improved. In the final model, the addition of the feature pyramid further improves gesture recognition accuracy. The recognition accuracy of the improved model increased by about 5%.

At the same time as the experiment of the transformer, we used YOLO to conduct a control experiment. The purpose of using another set of control experiments is to compare the performance of the Detection Transformer and YOLO series models in terms of recognition accuracy. As shown in the figure below, we used another group of data as the test dataset.


Love Good

You My

Love Good

You My

Our Dataset

Figure 4.3: Another group of samples and recognition results

We are use of YOLOv6, which has the highest recognition accuracy in the YOLO series, as a control model. In the same gesture dataset, YOLOv6 peaks in accuracy after 120 epochs of training. As shown in the figure, YOLOv6 has an average accuracy only 83.75% in the four categories tested. And our model has an accuracy 96.45%.

Models/Classes	“Love”	“Good”	“You”	“Meet”	“Yes”	“No”	“Please”	“Name”	“My”	AP
ResNet18 + DETR	85.23%	83.82%	84.28%	84.11%	83.37%	84.45%	85.68%	84.79%	84.25%	84.44%
ResNet34 + DETR	84.77%	85.27%	86.89%	85.83%	84.12%	85.35%	86.63%	85.76%	85.47%	85.56 %
ResNet50 + DETR	87.35%	88.26%	89.71%	88.12%	87.07%	88.34%	89.33%	88.23%	88.26%	88.29 %
ResNet101 + DETR	89.37%	90.49%	91.55%	89.35%	90.23%	91.30%	90.24%	89.66%	91.95%	90.46%
YOLOv3	82.63%	83.66%	84.36%	83.11%	82.19%	83.59%	84.87%	83.41%	84.92%	83.63%
YOLOv4	84.89%	83.02%	84.93%	85.96%	84.53%	84.75%	83.83%	84.29%	85.81%	84.66%
YOLOv5+Attention	91.28%	92.79%	90.33%	91.28%	92.72%	91.85%	91.97%	90.38%	91.45%	91.56%
YOLOX + ViT	93.76%	92.96%	93.94%	94.22%	93.18%	93.27%	94.39%	93.78%	92.96%	93.61%
ResNet152 + FPN + DETR (proposed)	95.64%	96.73%	97.15%	96.27%	96.55%	97.40%	96.16%	95.52%	96.69%	**96.45%**

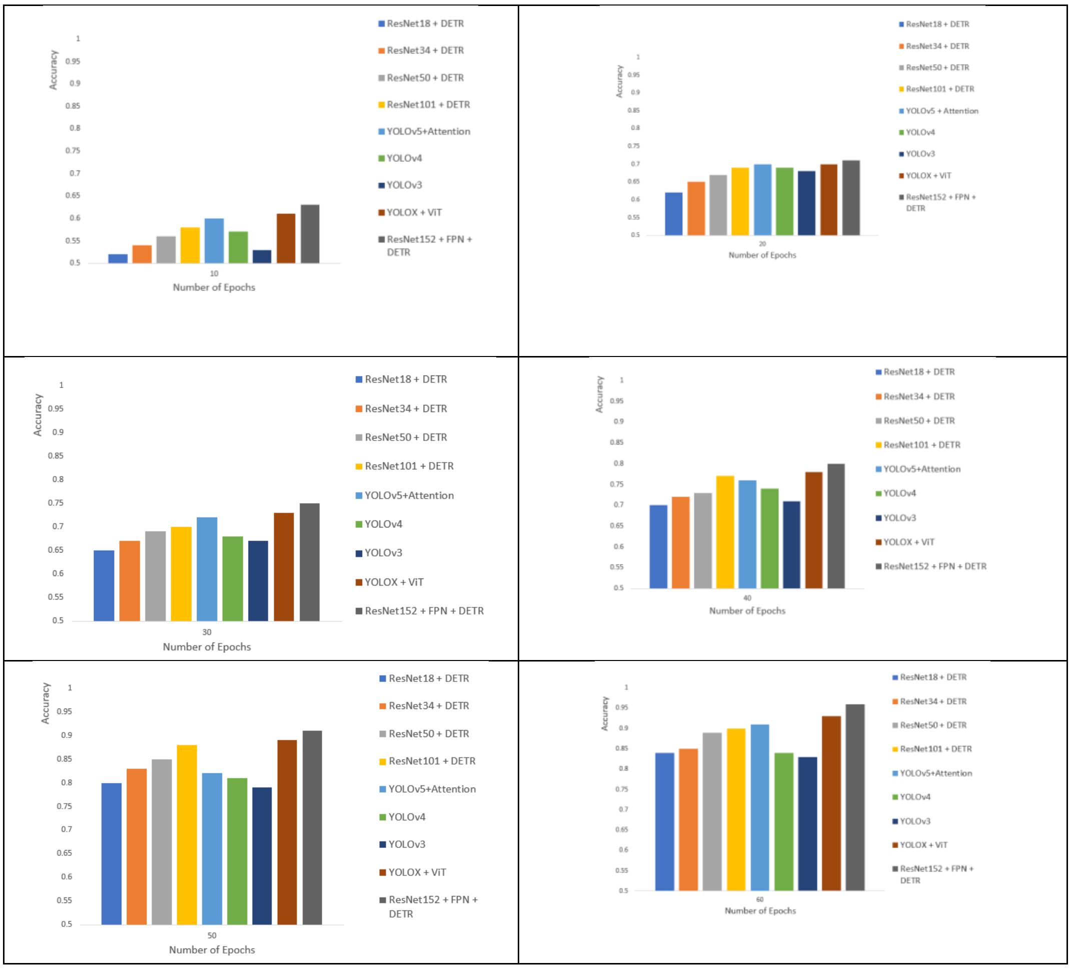
Table 4.2: The comparisons of deep learning models

In Table 4.2, our proposed method shows excellent results for sign language recognition. It is able to get 96.45% accuracy which has a 5.99% growth of the total accuracy compared with the ResNet101 + DETR. YOLOX + Vision Transformer for sign language recognition attains 93.72% accuracy.

In Table 4.2, it is observed that the proposed method has a better recognition rate that can reach 28 FPS compared to experimented methods due to its jump connection structure to avoid gradient dispersion. ResNet152 as the feature extraction network, contains more feature information and more semantic information in the upper layer of the feature map.

Combined with FPN structure to fuse high-level and low-level information, ResNet152 improved the average accuracy to 96.45%.

As shown in Figure 4.4, the accuracy and validation losses are listed. The black bar represents the proposed method. During training, we set the number of training epochs to 60. Because the test models all peaked before 60 epochs. All the methods get the max value. The proposed method reaches the highest accuracy of 96%. The proposed method also attains the best performance for the validation process than other methods. From the loss function curve, we can see that our curve is smoother than other models and shows better results.



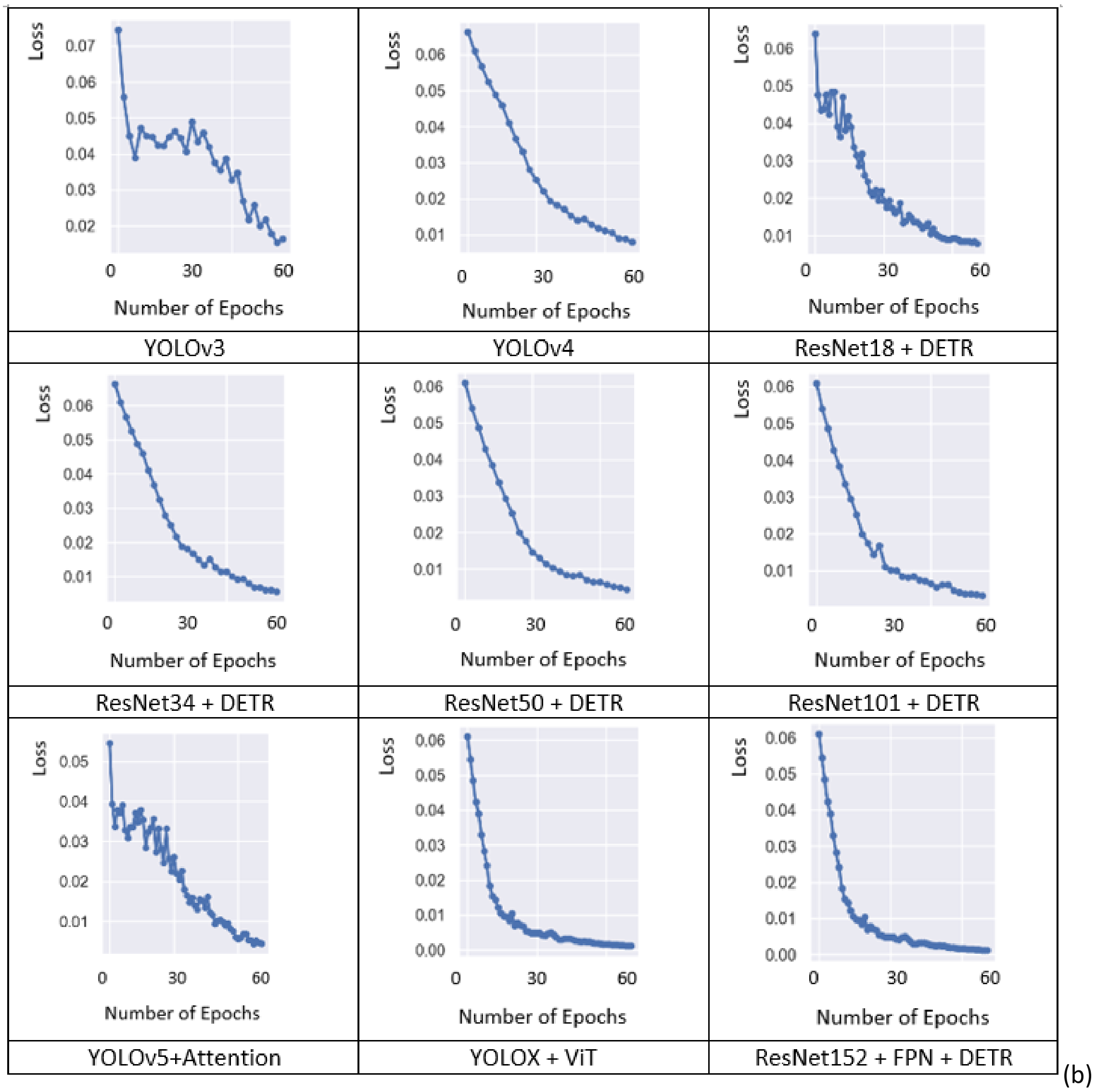


Figure 4.4 The accuracy (a) and validation (b) losses for our proposed methods

4.2 Limitations of This Thesis

In this thesis, the limitations still exist in the experiment:

(1) As we created our own dataset, the data corpus and the size of the dataset are limited. The total images and the number of classes for sign language gestures are less than the standard dataset.

(2) The complexity of this model is higher than the previously proposed recurrent neural

network (RNN).

(3) Although our model is able to obtain better results than other models, we need more epochs for training data. It will be more time wasted than other models.

4.3 Analysis

Our main purpose in this thesis is to improve the accuracy of sign language recognition. In the evaluation method section, we mentioned evaluating the model using AP and mAP. mAP is a model evaluation criterion for datasets containing multiple classes. In this thesis we comprehensively evaluate the test model on the basis of nine gesture categories. Thus, mAP is selected in this thesis instead of AP. For the test model, the main comparison model is the ResNet+DETR series. For the YOLO series model, we select the model with the highest accuracy in the test dataset as the representative. The evaluation results are shown in the figure below.

4.3.1 mAP

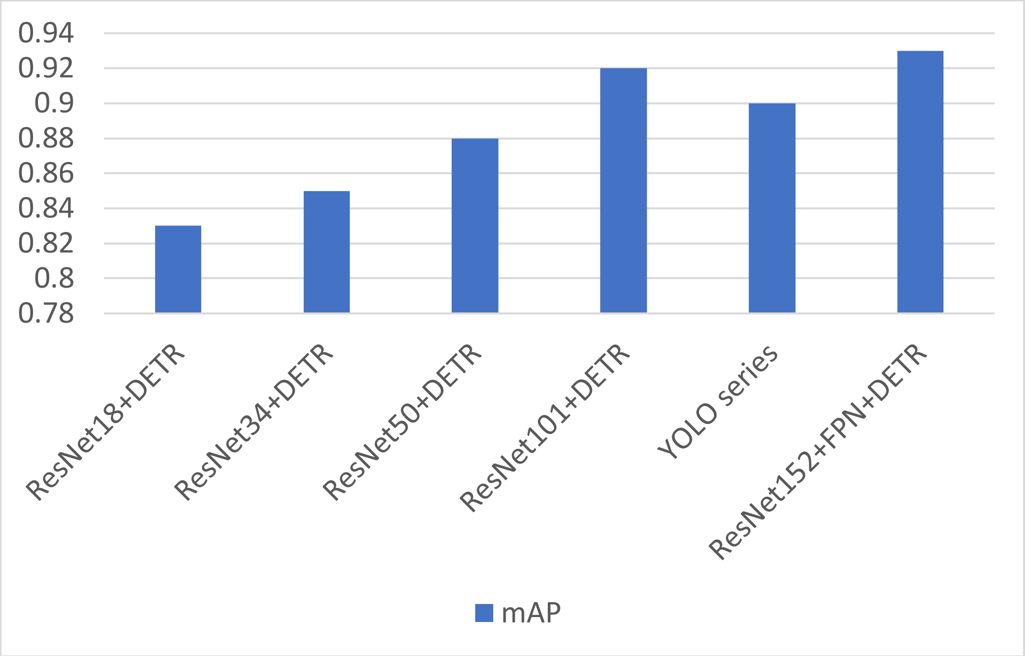


Figure 5.1: mAP of the tested models

As shown in Figure 5.1, each model in the test adopts the same dataset. Besides, they are trained on the dataset for 100 epochs under the condition of the same type and number of recognitions. The results in the figure show that our proposed model achieves the highest accuracy. Compared to the YOLO series of models, our model has a 3% improvement in accuracy.

4.3.2 AP$_s$, AP$_M$ and AP$_L$

Another very important test point is the accuracy for different target sizes. In this thesis, we adopt the feature pyramid network to improve the recognition accuracy of small objects. For the evaluation metric, AP(S) represents the AP measurement of target boxes with a pixel area of less than 32 square. AP(M) represents the AP measurement of target boxes with a pixel area between 32 and 96 squares. AP(L) represents the AP measurement of target boxes with a pixel area larger than 96 squares.

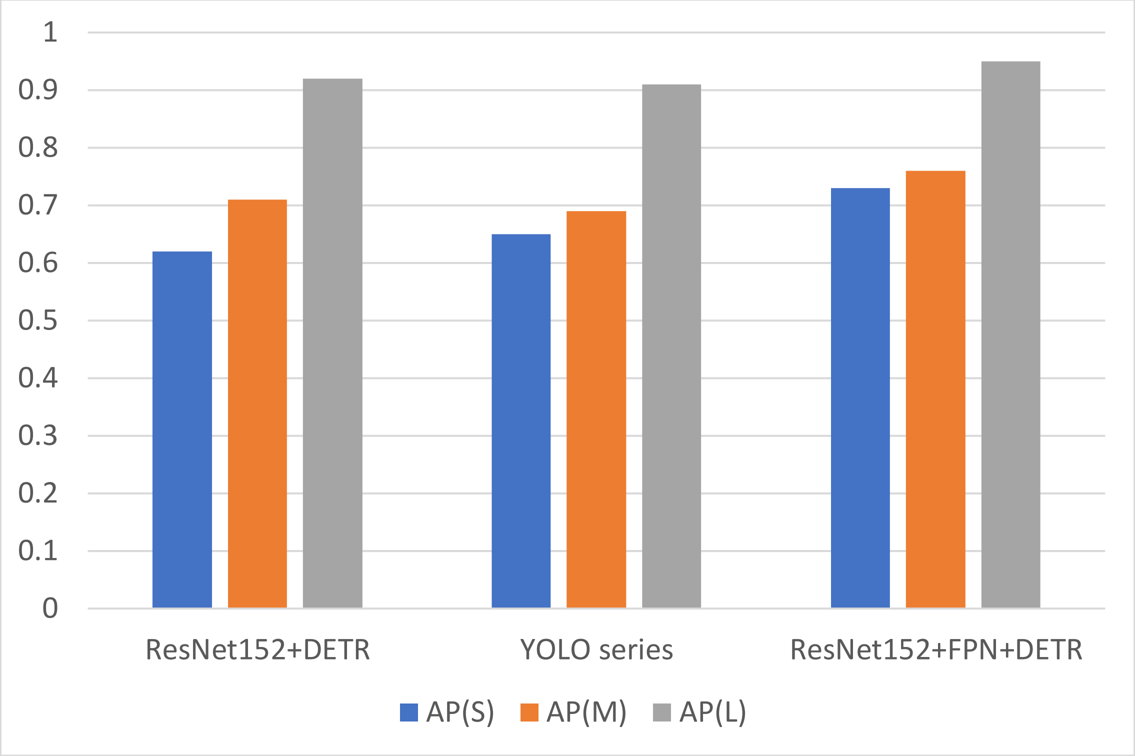


Figure 5.2: AP for different sizes in tested models

As shown in Fig. 5.2, with the proposed model as the test model, we compare the accuracy of ResNet152 with and without feature pyramids added. In addition, the comparison of YOLO series models with ResNet152 is also used as part of the experiments. The results in the figure show that our model has the highest recognition accuracy for small targets, medium targets and large targets.

4.4 Discussion

In our experiments, we make use of the average accuracy to evaluate targets of different sizes. The experimental results show that after using the feature pyramid, the recognition accuracy of small targets has been significantly improved. The experimental results show that our model has an 11% increase in accuracy compared to the model after removing the feature pyramid. Compared with the YOLO series of models, the recognition accuracy of our model for small objects is improved by 8%. Our model surpasses other experimental models in the recognition accuracy of both medium and large objects. In terms of mean average accuracy, by comparing ResNet series models, YOLO series models and our proposed model, our model achieves the highest accuracy rate of 93%. Compared with the ordinary ResNet series model, it is improved by 6%. In addition, our model is up three percentage points over the YOLO series model.

In summary, our applied model is better able to cope with gesture recognition of different sizes. At the same time, it has better performance than other test models in terms of recognition accuracy. Furthermore, our model performs better than other models in recognizing different categories of gestures.

Chapter 5
Conclusion and Future Work

In this chapter, we will summarize the subject and methods of this project and propose a new research direction according to the result and insufficiency of the experiment, preparing for future work.

5.1 Conclusion

In this thesis, we employed ResNet152+FPN+DETR model to achieve superior performance of sign language recognition. The experiments show that the new model shows better results than the existing methods, which has a 1.7% growth of accuracy by adding the FPN nets.

The results show that Transformer still has excellent potential for improving sign language recognition by adding the convolutional layers and increasing the feature maps to improve the model's accuracy. Although the computational complexity and parameters have increased compared to the previous method, this problem can still be continuously improved in the future. Besides, applying the FPN nets in the DETR-based models shows great betterment in sign language recognition.

5.2 Future Work

In our future work, we will combine YOLO model and Transformer to obtain better results, which will uplift the performance of sign language recognition. In addition, more phrases and terms of sign language will be considered in our future work.

.

References

An, N., Yan, W. (2021) Multitarget tracking using Siamese neural networks. ACM Transactions on Multimedia Computing, Communications and Applications.

An, N. (2020) Anomalies Detection and Tracking Using Siamese Neural Networks. Master's Thesis. Auckland University of Technology, New Zealand.

Abavisani, M., Joze, H.R.V. and Patel, V.M. (2019). Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1165-1174).

Abdullah, S., Ahmed, Y. B., Kanwal, K., Saher, Y., & Jafri, A. R. (2014). Assistive glove for Pakistani sign language translation. In *IEEE International Multi Topic Conference,* (pp. 173-176).

Abdulrahman, A., & Iqbal, K. (2014). Capturing human body dynamics using RNN Based on persistent excitation data generator. In *International Symposium on Computer-Based Medical Systems (CBMS),* (pp. 221-226). IEEE.

Anderson, C. H., Burt, P. J., & Van Der Wal, G. S. (1985). Change detection and tracking using pyramid transform techniques. In *Cambridge Symposium* (pp. 72-78). International Society for Optics and Photonics.

Agarwal, R., Raman, B. and Mittal, A. (2015). Hand gesture recognition using discrete wavelet transform and support vector machine. In *International Conference on Signal Processing and Integrated Networks* (pp. 489-493).

Ali, A.S., ÇEVİK, M. and ALQARAGHULI, A. (2022). American Sign Language Recognition using YOLOv4 Method. International Journal of Multidisciplinary Studies and Innovative Technologies, pp.61-65.

Baum, L. E., & Sell, G. (1968). Growth transformations for functions on manifolds.

Pacific Journal of Mathematics, pp. 211-227.

Bauer, B., Hienz, H., Kraiss, K.F. (2000). Video-Based Continuous Sign Language Recognition Using Statistical Methods. In *International Conference on Pattern Recognition (ICPR)* (pp. 463-466)

Bastanfard, A., Rezaei, N. A., Mottaghizadeh, M., & Fazel, M. (2010). A novel multimedia educational speech therapy system for hearing impaired children. In *Pacific-Rim Conference on Multimedia* (pp. 705-715).

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 157-166.

Bhatti, U. A., Huang, M., Wu, D., Zhang, Y., Mehmood, A., & Han, H. (2019). Recommendation system using feature extraction and pattern recognition in clinical care systems. *Enterprise Information Systems*, pp. 329-351.

Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934.

Cao, X. (2022) Pose Estimation of Swimmers from Digital Images Using Deep Learning. Master's Thesis, Auckland University of Technology.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020). End-to-end object detection with Transformers. arXiv: 2005.12872

Casado-García, A., del-Canto, A., Sanz-Saez, A., Pérez-López, U., Bilbao-Kareaga, A., Fritschi, F.B., Miranda-Apodaca, J., Muñoz-Rueda, A., Sillero-Martínez, A., Yoldi-Achalandabaso, A. and Lacuesta, M. (2020). LabelStoma: A tool for stomata detection based on the YOLO algorithm. Computers and Electronics in Agriculture, p.105751.

Chatzis, S. P., & Kosmopoulos, D. I. (2011). A variational Bayesian methodology for hidden Markov models utilizing Student's-t mixtures. *Pattern Recognition*, *44*(2), 295-306.

Chen, Y. N., Han, C. C., Wang, C. T., Jeng, B. S., & Fan, K. C. (2006). The application of a convolution neural network on face and license plate detection. In *International Conference on Pattern Recognition,* (Vol. 3, pp. 552-555).

Chen, Y., Zhao, L., Peng, X., Yuan, J., & Metaxas, D. N. (2019). Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. In *British Machine Vision Conference* (pp. 1-13).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cho, K., Murawski, J., Naruniec, J., Wojna, Z., & Kisantal, M. (2019). Augmentation for small object detection. arXiv:1902.07296.

Cui, W., Yan, W. (2016) A scheme for face recognition in complex environments. International Journal of Digital Crime and Forensics (IJDCF) 8 (1), 26-36.

Cui, W. (2015) A Scheme of Human Face Recognition in Complex Environments. Master's Thesis, Auckland University of Technology.

Cummins, F., Gers, F. A., & Schmidhuber, J. (1999). Language identification from prosody without explicit features. In *EUROSPEECH'99* (pp. 371–374).

Chun, S., Han, D., Yun, S., Joon Oh, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. arXiv:1905.04899.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Collins, R. T., Lipton, A. J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., & Wixson, L. (2000). A system for video surveillance and monitoring. Research Report, CMU.

Conley , G. , Zinn , S.C. , Hanson , T. , McDonald , K. , Beck , N. and Wen , H. (2022).

Using a deep learning model to quantify trash accumulation for cleaner urban stormwater. Computers, Environment and Urban Systems, pp.101752.

Ćorović, A., Ilić, V., Ðurić, S., Marijan, M. and Pavković, B. (2018). The real-time detection of traffic participants using YOLO algorithm. In *Telecommunications Forum* (pp. 1-4).

Dadashzadeh, A., Targhi, A. T., Tahmasbi, M., & Mirmehdi, M. (2019). HGR-Net: A fusion network for hand gesture segmentation and recognition. *IET Computer Vision,* pp. 700-707.

Daniels , S. , Suciati , N. and Fathichah , C. (2021). Indonesian Sign Language Recognition. IOP Conference Series: Materials Science and Engineering, p. 012029.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X.; Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv:2010.11929

Duan, J., Zhou, S., Wan, J., Guo, X., & Li, S. Z. (2016). Multi-modality fusion based on consen-sus-voting and 3D convolution for isolated gesture recognition. arXiv:1611.06689.

Eickeler, S., & Muller, S. (1999). Content-based video indexing of TV broadcast news using hidden Markov models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing.*(Vol. 6, pp. 2997-3000).

Elboushaki, A., Hannane, R., Afdel, K., & Koutti, L. (2020). MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. *Expert Systems with Applications*.

Fu, R., Zhang, Z. and Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. *In Youth Academic Annual Conference of Chinese Association of Automation (YAC)*.

Gao, X., Nguyen, M., Yan, W. (2021) Face image inpainting based on generative adversarial network. International Conference on Image and Vision Computing New Zealand.

Gao, X., Nguyen, M., Yan, W. (2022) A face image inpainting method based on autoencoder and adversarial generative networks. Pacific-Rim Symposium on Image and Video Technology.

Gao, X. (2022) A Method for Face Image Inpainting Based on Generative Adversarial Networks. Masters Thesis, Auckland University of Technology, New Zealand.

Gowdra, N., Sinha, R., MacDonell, S., Yan, W. (2021) Maximum Categorical Cross Entropy (MCCE): A noise-robust alternative loss function to mitigate racial bias in Convolutional Neural Networks (CNNs) by reducing overfitting. Pattern Recognition.

Gers, F. A., & Schmidhuber, J. (2000). Recurrent nets that time and count. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference,* pp. 189-194.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451-2471.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*(Aug), 115-143.

Gers, F. A., & Schmidhuber, E. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, *12*(6), 1333-1340.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*, 115-143.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580-587).

Gu, D., Nguyen, M., Yan, W. (2016) Cross models for twin recognition. International Journal of Digital Crime and Forensics 8 (4), 26-36.

Haritaoglu, I., Harwood, D., & Davis, L. S. (2000). W/sup 4: real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), 809-830.

Han, M., Chen, J., Li, L., & Chang, Y. (2016). Visual hand gesture recognition with convolution neural network. *IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (pp. 287-291).

Han, X., Gao, Y., Lu, Z., Zhang, Z., & Niu, D. (2015). Research on Moving Object Detection Algorithm Based on Improved Three Frame Difference Method and Optical Flow. In *International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)* (pp. 580-584).

Han , J. , Liao , Y. , Zhang , J. , Wang , S. and Li , S. (2018). Target fusion detection of LiDAR and camera based on the improved YOLO algorithm. *Mathematics*, p.213.

Heikkila, M., & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 657-662.

Herrera, A., Beck, A., Bell, D., Miller, P., Wu, Q., Yan, W. (2008) Behavior analysis and prediction in image sequences using rough sets. International Machine Vision and Image Processing Conference (pp.71-76)

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation,* pp. 1527-1554.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hori, T., Kubo, Y., & Nakamura, A. (2014). Real-time one-pass decoding with recurrent neural network language model for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6364-6368).

Huang, J., Zhou, W., Li, H., & Li, W. (2015). Sign language recognition using 3D convolutional neural networks. In *IEEE International Conference on Multimedia and Expo,* pp. 1-6.

Huang, R., Pedoeem, J., & Chen, C. (2018). YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers. In *IEEE International Conference on Big Data* (pp. 2503-2510)

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, *160*(1), 106-154.

Hussain, R., Karbhari, Y., F. Ljaz, M., Woźniak, M., Singh, P. K., & Sarkar, R. (2021). Revise-Net: Exploiting reverse attention mechanism for salient object detection. *Remote Sensing*, pp. 2072-4292.

Ji, H., Liu, Z., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer's disease based on selective kernel network with spatial attention. Asian Conference on Pattern Recognition.

Ji, H., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer’s disease using deep learning. ACM ICCCV.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 221 – 231.

Jiang, P., Ergu, D., Liu, F., Cai, Y. and Ma, B. (2022). A review of YOLO algorithm developments. *Procedia Computer Science*, pp.1066-1073.

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* (pp. 779-788).

Kim, J., & Kim, H. (2016). Classification performance using gated recurrent unit recurrent neural network on energy disaggregation. In *International Conference on Machine Learning and Cybernetics (ICMLC)* (pp. 105-110).

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, (pp. 1097–1105).

Ku, B., Kim, K. and Jeong, J. (2022). Real-time ISR-YOLOv4 based small object detection for safe shop floor in smart factories. *Electronics*, p.2348.

Koller, O., Ney, H., & Bowden, R. (2016). Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3793-3802).

Le, R., Nguyen, M., Yan, W. (2018) A vision aid for the visually impaired using commodity dual-rear-camera smartphones. International Conference on Mechatronics and Machine Vision.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE,* pp. 2278–2324.

LeCun, Y. (1989). Generalization and network design strategies. *Connectionism in perspective*, pp. 143-155.

LeCun, Y., Huang, F. J., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Conference on Computer Vision and Pattern Recognition.*

LeCun, Y., & Ranzato, M. (2013). Deep learning tutorial. In *International Conference on Machine Learning (ICML'13).*

Lee, D.L. and You, W.S. (2018). Recognition of complex static hand gestures by using the wristband‐based contour features. *IET Image Processing*, pp.80-87.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W. and Li, Y. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. International Conference on Pattern Recognition (ICPR), (pp.2734-2739).

Li, R., Nguyen, M., Yan, W. (2017) Morse codes enter using finger gesture recognition. International Conference on Digital Image Computing: Techniques and Applications.

Liang, C., Lu, J., Yan, W. (2022) Human action recognition from digital videos based on deep learning. ACM ICCCV.

Lin, T.Y., Dollar, P., Girshick, R., He K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2117-2125).

Liu, C., Yan, W. (2020) Gait recognition using deep learning. Handbook of Research on Multimedia Cyber Security (pp.214-226)

Liu, J., Yan, W. (2022) Crime prediction from surveillance videos using deep learning. Aiding Forensic Investigation Through Deep Learning and Machine Learning Frameworks. IGI Global.

Liu, G., Nouaze, J.C., Touko Mbouembe, P.L. and Kim, J.H. (2020). YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors*, pp.2145.

Liu, H., & Hou, X. (2012). Moving detection research of background frame difference based on Gaussian model. In *International Conference on Computer Science & Service System (CSSS),* (pp. 258-261).

Liu, J., Kuipers, B. & Savarese, S. (2011). Recognizing human actions by attributes. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3337–3344).

Lu, J. (2016) Empirical Approaches for Human Behavior Analytics. Master's Thesis. Auckland University of Technology, New Zealand.

Lu, J., Shen, J., Yan, W., Boris, B. (2017) An empirical study for human behaviors analysis. International Journal of Digital Crime and Forensics 9 (3), 11-17.

Lu, J., Nguyen, M., Yan, W. (2018) Pedestrian detection using deep learning. IEEE AVSS.

Lu, J., Nguyen, M., Yan, W. (2020) Human behavior recognition using deep learning. International Conference on Image and Vision Computing New Zealand.

Lu, J., Nguyen, M., Yan, W. (2020) Comparative evaluations of human behavior recognition using deep learning. Handbook of Research on Multimedia Cyber Security, 176-189.

Lu, J., Nguyen, M., Yan, W. (2021) Sign language recognition from digital videos using deep learning methods. International Symposium on Geometry and Vision.

Lu, J. (2021) Deep Learning Methods for Human Behavior Recognition. PhD Thesis. Auckland University of Technology, New Zealand.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision* (pp. 21-37).

Liu, Y., Nand, P., Hossain, M., Nguyen, M., Yan, W. (2023) Sign language recognition from digital videos using feature pyramid network with Detection Transformer. Multimedia Tools and Applications.

Liu, Z., Zhang, C., & Tian, Y. (2016). 3D-based deep convolutional neural network for action recognition with depth sequences. *Image and Vision Computing*, pp. 93-100.

Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, *16*(5), 555-559.

Maryam, K., & Reza, K. M. (2012). An analytical framework for event mining in video data. *Artificial Intelligence Review, 41*(3), pp. 401-413.

Mikolov, T., Kombrink, S., Burget, L., Černocký, J., & Khudanpur, S. (2011). Extensions of recurrent neural network language model. In *IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP), (pp. 5528-5531). IEEE.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech* (Vol. 2, p. 3).

Minoofam, S., Bastanfard, A. & Keyvanpour, M. (2022). RALF: An adaptive reinforcement learning framework for teaching dyslexic students. Multimed Tools Appl 81, pp. 6389–6412.

Mishra, A., Kumar V., Shiva, M., Reddy, K., Arulkumar, S., Rai, P., Mittal, A. (2018). A generative approach to zero-shot and few-shot action recognition. In *IEEE Winter Conference on Applications of Computer Vision* (pp. 372–380)

Misra, D. (2019). Mish: A self regularized non-monotonic activation function. arXiv:1908.08681.

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4207-4215)

Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, *1*(1), 4-27.

Neverova, N., Wolf, C., Taylor, G. W., & Nebout, F. (2014). Hand segmentation with structured convolutional learning. In *Asian Conference on Computer Vision* (pp. 687-702).

Ni, Z., Chen, J., Sang, N., Gao, C. and Liu, L., (2018). Light YOLO for high-speed gesture recognition. In *IEEE International Conference on Image Processing* (ICIP) (pp. 3099-3103).

Nimisha, K.P. and Jacob, A. (2020). A brief review of the recent trends in sign language recognition. In *International Conference on Communication and Signal Processing* (ICCSP), pp. 186-190.

Noorit, N., & Suvonvorn, N. (2014). Human activity recognition from basic actions using finite state machine. In *International Conference on Advanced Data and Information Engineering (DaEng-2013)* (pp. 379-386). Springer.

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception. Multimedia Tools and Applications, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. IEEE Transactions on Image Processing.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3)*, *28*, 1310-1318.

Petrushin, V. A. (2005). Mining rare and frequent events in multi-camera surveillance video using self-organizing maps. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 794-800).

Popoola, O. P., & Wang, K. (2012). Video-based abnormal human behavior recognition—A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 865-878.

Qin, J., Liu, L., Shao, L., Shen, F., Ni, B., Chen, J., Wang, Y. (2017). Zero-shot action recognition with error-correcting output codes. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2833–2842)

Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry, A. S. C. S. (2018). Deep convolutional neural networks for sign language recognition. In *The Conference on Signal Processing and Communication Engineering Systems* (pp. 194-197).

Rastgoo, R., Kiani, K., & Escalera, S. (2018). Multi-modal deep hand sign language recognition in still images using restricted Boltzmann machine. *Entropy.*

Remagnino, P., Monekosso, D. N., & Jain, L. C. (Eds.). (2011). *Innovations in Defence Support Systems-3: Intelligent Paradigms in Security* (Vol. 336). Springer Science & Business Media.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1137-1149.

Rivera-Acosta, M., Ruiz-Varela, J.M., Ortega-Cisneros, S., Rivera, J., Parra-Michel, R. and Mejia-Alvarez, P. (2021). Spelling correction real-time American sign language alphabet translation system based on yolo network and LSTM. Electronics, p.1035.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pp. 3856-3866.

Sahoo, A. K. (2021). Indian sign language recognition using machine learning techniques. *Macromolecular Symposia.*

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*.

Santos, C. C. D., Samatelo, J. L. A., & Vassallo, R. F. (2020). Dynamic gesture recognition by using CNNs and star RGB: A temporal information condensation. *Neurocomputing,* pp.238-254.

Schwenk, H. (2007). Continuous space language models. *Computer Speech & Language*, *21*(3), 492-518.

Shen, D., Xin, C., Nguyen, M., Yan, W. (2018) Flame detection using deep learning. International Conference on Control, Automation and Robotics.

Shen, J., Yan, W., Miller, P., Zhou, H. (2010) Human localization in a cluttered space using multiple cameras. IEEE International Conference on Advanced Video and Signal Based Surveillance.

Song, H., 2022. Multi-scale safety helmet detection based on RSSE-YOLOv3. *Sensors*, p.6061.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.

Sisson, J. H., Stoner, J. A., Ammons, B. A., & Wyatt, T. A. (2003). All-digital image capture and whole-field analysis of ciliary beat frequency. *Journal of microscopy*, *211*(2), 103-111.

Starner, T., Pentland, A. (1997). Real-time American sign language recognition from video using hidden Markov models. *Computational Imaging and Vision* (pp. 227–243).

Starzyński, J., Zawadzki, P. and Harańczyk, D. (2022). Machine learning in solar plants inspection automation. *Energies*, pp.5966.

Tamura, S., Kawasaki, S. (1988). Recognition of sign language motion images. *Pattern Recognition* (pp. 343–353).

Tang, Y., Huang, Y., Wu, Z., Meng, H., Xu, M., & Cai, L. (2016). Question detection from acoustic features using recurrent neural network with gated recurrent unit. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6125-6129). IEEE.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning* (pp. 10347–10357).

Trinh, H., Fan, Q., Jiyan, P., Gabbur, P., Miyazawa, S., & Pankanti, S. (2011). Detecting human activities in retail surveillance using hierarchical finite state machine. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1337-1340). IEEE.

Vaswani, A., N. Shazeer, N. Parmar, L. Yang, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, & I. Polosukhin. (2017). Attention is all you need. arXiv: 1706.03762

Wang, C. Y., Liao, H. Y., Wu, Y. H., Chen, P. Y., Hsieh, J., & Yeh, I. (2020). CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 390-391).

Wang, P., Li, W., Liu, S., Gao, Z., Tang, C., & Ogunbona, P. (2016). Large-scale isolated gesture recognition using convolutional neural networks. In *International Conference on Pattern Recognition* (pp. 7-12).

Wang, X., Yan, W. (2019) Human gait recognition based on self-adaptive hidden Markov model. IEEE/ACM Transactions on Biology and Bioinformatics.

Wang, X., Yan, W. (2019) Cross-view gait recognition through ensemble learning. Neural Computing and Applications.

Wang, X., Yan, W. (2019) Gait recognition using multichannel convolutional neural networks. Neural Computing and Applications.

Wang, X., Yan, W. (2019) Multi-perspective gait recognition based on ensemble learning. Springer Neural Computing and Applications.

Wang, X., Yan, W. (2019) Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. International Journal of Neural

Systems.

Wang, X., Yan, W. (2020) Non-local gait feature extraction and human identification. Springer Multimedia Tools and Applications.

Wang, X., Yan, W. (2020) Cross-view gait recognition through ensemble learning. Neural Computing and Applications 32 (11), 7275-7287.

Wang, X., Yan, W. (2022) Human identification based on gait manifold. Applied Intelligence.

Wildes, R. P. (1998). A measure of motion salience for surveillance applications. In *International Conference on Image Processing* (pp. 183-187). IEEE.

Wu, J., Ishwar, P., & Konrad, J. (2016). Two-stream CNNs for gesture-based verification and identification: Learning user style. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 42- 50.

Wu, X., Fang, J., Yan, W. (2023) Contrast optimization for size invariant visual cryptography scheme. IEEE Transactions on Image Processing 32, 2174-2189.

Varol, G., Laptev, I., & Schmid, C. (2016). Long-term temporal convolutions for action recognition. *arXiv preprint arXiv:1604.04494*.

Xiang, N., Pan, C., & Li, X. (2021). An object algorithm combining FPN structure with DETR. In *International Conference on Control and Computer Vision* (pp. 57–63)

Xiaoyang, Y., Yang, Y., Shuchun, Y., Yang, S., Huimin, Y., & Xifeng, L. (2013). A novel motion object detection method based on improved frame difference and improved Gaussian mixture model. In *International Conference on Measurement, Information and Control* (Vol. 1, pp. 309-313). IEEE.

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,* pp. 1492-1500.

Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems* (pp. 802-810).

Xu, T., Hospedales, M., Gong, S. (2016). Multi-task zero-shot action recognition with prioritized data augmentation, In *European Conference on Computer Vision*, pp. 343–359.

Yan, W., Liu, F. (2015) Event analogy-based privacy preservation in visual surveillance. Pacific-Rim Symposium on Image and Video Technology, 357-368.

Yan, W. (2019) Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer London.

Yan, W. (2021) Computational Methods for Deep Learning: Theoretic, Practice and Applications. Springer London.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4694-4702).

Yin, K., & Read, J. (2020). Better sign language translation with STMC-Transformer. In *International Conference on Computational Linguistics*, pp. 5975-5989.

Yu, Z. (2021) Deep Learning Methods for Human Action Recognition. Master's Thesis, Auckland University of Technology, New Zealand.

Yu, Z., Yan, W. (2020) Human action recognition using deep learning methods. International Conference on Image and Vision Computing New Zealand.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zaremba, W. (2015). An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning.*

Zhang, Y., Er, M. J., Venkatesan, R., Wang, N., & Pratama, M. (2016). Sentiment classification using comprehensive attention recurrent models. In *International Joint Conference on Neural Networks (IJCNN)* (pp. 1562-1569).

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang Z., Hou Q., and Jiashi Feng J. (2021). DeepViT: Towards deeper Vision Transformer. arXiv: 2103.11886.

Zhu, Y., Yan, W. (2022) Ski fall detection from digital images using deep learning. ACM ICCCV.

Zhu, Y., Yan, W. (2022) Image-based storytelling using deep learning. ACM ICCCV.

Zhuang, H., Yang, M., Cui, Z. and Zheng, Q. (2017). A method for static hand gesture recognition based on non-negative matrix factorization and compressive sensing. *IAENG International Journal of Computer Science*, pp.52-59.