Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-14646-0

Sign language recognition from digital videos using feature pyramid network with detection transformer

Yu Liu[1] · Parma Nand[1] · Md Akbar Hossain[1] · Minh Nguyen[1] · Wei Qi Yan[1]

Received: 11 March 2022 / Revised: 13 June 2022 / Accepted: 4 February 2023

Abstract
Sign language recognition is one of the fundamental ways to assist deaf people to communicate with others. An accurate vision-based sign language recognition system using deep learning is a fundamental goal for many researchers. Deep convolutional neural networks have been extensively considered in the last few years, and a slew of 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 paper, we propose a Vision Transformer-based sign language recognition method called DETR (Detection Transformer), aiming to improve the current state-of-the-art sign language recognition accuracy. The DETR method proposed in this paper is able to recognize sign language from digital videos with a 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 up to 1.70% on the test dataset of sign language compared to the standard Detection Transformer models. Besides, an overall accuracy 96.45% was attained by using the proposed method.

Keywords Sign language recognition · ResNet152 · Detection transformer · Feature pyramid network

1 Introduction

Sign language recognition is significant for deaf or hearing-impaired people [2]. 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

✉ Wei Qi Yan
wyan@aut.ac.nz

[1] Auckland University of Technology, Auckland 1010, New Zealand

Published online: 28 February 2023

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, it is true that sign language is one of the great linguistic discoveries. It acts as a fundamental mode of communications for the hearing and speech impaired people, without it the communications between them and others might be difficult.

Sign language recognition from digital images and videos is regarded as a type of behavior identification. Usually, it is implemented by using machine learning approaches. Nowadays, deep learning is employed [12] for sign language recognition. Deep learning models have peformed extremely well in image processing as well as natural language processing, however a fundamental requirement of this class of models is a large dataset. In the case of sign language recognition with deep learning, we would need a large dataset with sign language gestures spanning the range of the sign language vocabulary. In general, sign language recognition has three steps: Detecting, tracking, and recognizing gestures. The difficulty of this recognition is that we need to extract task-related data or features in an efficient way. We apply Detection Transformer (DETR) as a basic structure to solve this problem of the RNN models based on sequential computations, hence large matrix multiplications could not be parallelized for computational efficiency. As a solution to this, the encoder-decoder framework was proposed which takes use of attention mechanism [27], hence it can be easily parallelized for machine understanding tasks.

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. In this paper, the main contributions are:

- We create a new model that makes use of a novel backbone network ResNet152 and Feature Pyramid Network (FPN) as the neck. The structure is able to increase input features, which boosts the quality of the final output.
- As one part of this research work, we create our own dataset. This dataset is now publicly available for model training and testing at github.com.
- Regarding the purpose of evaluations, we also compare our proposed method with other DETR-based models, the results surpass ResNet34, ResNet50 and ResNet101 in terms of AP, AP_{50}, AP_{75}, and F_1 scores.

The paper is organized as follows. In Section 2, we highlight previous work related to this research project. The methodology is explained in Section 3. In addition, our experimental results are detailed in Section 4. Finally, in Section 5, we conclude this work by highlighting the findings and future work.

2 Related work

Computational sign language recognition has been a hot topic over the past decades [3, 19, 23–25]. In recent years, using Transformers to detect visual objects has become a mainstream methodology. One of them is Vision Transformer (ViT) [7]. Firstly, ViT segments an image into a grid of squares and flattens each square into a single vector by concatenating all pixel channels in a square. Transformer is independent on the structure of the input images, so positional embeddings are added to each square, which enables the model to be trained with the input images. The feature maps were identical in the top layers of the deep ViT model; a re-attention method was proposed to enhance the features. As a result, the Top-1 accuracy was improved up to 1.6% by using the datasets ImageNet [33].

Though Vision Transformer has a good performance, it still needs to be pre-trained based on massive data (e.g., JFT-300 M, 300 million images) and fine-tuned based on the ImageNet dataset to achieve comparable performance to the CNN method, which requires enormous computational resources that limits the applications of the ViT method.

The computational complexity is related to the square of the token. The token is a non-overlapping patch sequence from cutting images. If the input feature map is a 56×56 image, it will have matrix operations around 3000+, which requires a large number of computational calculations. At the same time, the number of tokens in the original Transformer and the hidden size remain unchanged. Regarding the ResNet structure and the pyramid structure, the higher the number of layers, the less the number of tokens. To this end, we make use of local window self-attention by considering a part of the feature map for self-attention, and find a way to interact with this local information. Convolutions are made in order to replace the fully connected layer, which reduces the parameters and hence the computational cost.

The main advantage of data-efficient image Transformers (DeiT) [26] is that it does not require a massive amount of pre-training data, as it only relies on ImageNet data to generate the results.

One of the reasons that Transformer requires enormous computing power is that the model itself cannot encode the position of an object. The Transformer is different from CNN, which requires positional embedding to encode the position information of tokens. That is, disrupting the order of tokens in sequence will not significantly change the outcome. If the location information of a patch is not provided to the model, the model needs to be trained through the semantics of patches, which increases the training costs. In order to solve this problem, the fixed-position coding has been harnessed in DETR. Positional encoding is a 2D method proposed in 2020 [6]. The positional encoding is added to the self-attention of the encoder and multi-head attention of the decoder; object queries are also plugged into the decoder's attention modules. Multi-head attention makes use of multiple queries to compute multiple inputs in parallel.

Regarding sign language recognition, gesture representation can be of various types, such as class-related attributes [13], class labels [16], and handcrafted features [20, 30, 34]. In addition, typical methods were applied to 2D CNN to extract feature maps from input images and then recognize sign language gestures from temporal information [11, 28]. In the same way, 3D CNN [9, 14, 17] is an updated version of 2D CNN, which extracts visual features by applying 3D convolutional layers. 3D CNN shows excellent performance in extracting feature maps [22]. In this paper, our proposed method is to adopt CNN to extract feature maps from input images and add FPN to enhance the features from each layer of CNN.

A spate of computational methods have been proposed to translate sign languages from digital videos to natural languages, textual sentences [8, 10, 18, 31]. Yin et al. [32] proposed STMC-Transformer to improve the state-of-the-art ways by using 7 BLEU based on the video-to-text translation of the 2014 T dataset. Camgoz et al. [5] put forward the method by using connectionist temporal classification loss based on the Transformer to have an end-to-end translation. The performance was evaluated based on PHOENIX-Weather-2014 dataset; the performance was improved from 9.58 to 21.80 in BLEU-4 scores. Rastgoo et al. proffered a method called zero-shot sign language recognition (ZS-SLR). In the research work, a Transformer was employed for hand detection with AutoEncoder (AE) based on Long Short-Term Memory (LSTM). As a result, the proposed method showed better performance than other methods based on four datasets: RKS-PERSIANSIGN, First-Person, ASLVID, and isoGD [21].

Besides, the languages based on multimedia technology have attained great progress. Bastanfard et al. proposed a speech therapy system for hearing impaired children [1]. Minoofam et al. proffered an adaptive reinforcement learning framework called RALF through Cellular Learning Automata (CLA) to produce semantic meanings [15]. Additionally, an algorithm called spatial-spectral HSI classification has been put forward for extracting more effective features [4].

Pertaining to Detection Transformer, as shown in Fig. 1, this structure consists of the ResNet152, which replaces ResNet50 as the backbone. DETR adopts a regular CNN backbone to get 2D representations of input images. The model flattens it and passes it to the Transformer encoder for positional encoding. The Transformer decoder then takes a small number of positional embeddings as input, the target query additionally participates in the output of the encoder. Each output embedding of the decoder is passed to predict bounding boxes and detect object classes or shared feedforward network (FFN) without target class.

As shown in Fig. 2, the contribution of this paper is to propose a new model that consists of ResNet152 + FPN. Deep residual nets make use of residual blocks to improve the accuracy of the existing models. The concept of "skip connections" lies in the core of the residual blocks. It takes use of output from one layer of the network and quickly feed it into the next layer or even deeper into the neural network. We are use of jump connections to construct the ResNet, which can resolve the computational burden problem of a deep neural network. This is the strength of the new type of artificial neural networks.

ResNet152 is a ResNet model, which has 150 convolutional layers along with one max-pooling layer and one average-pooling layer. The FPN naturally exploits the pyramidal form of CNN features and generates feature pyramids with strong semantic information on all scales.

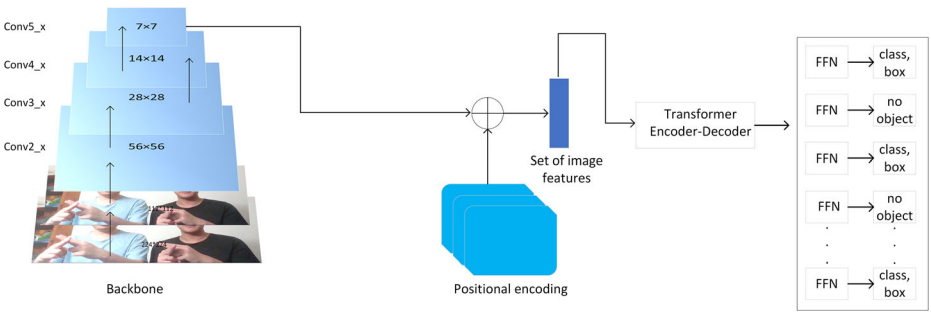


Fig. 1 The structure of DETR

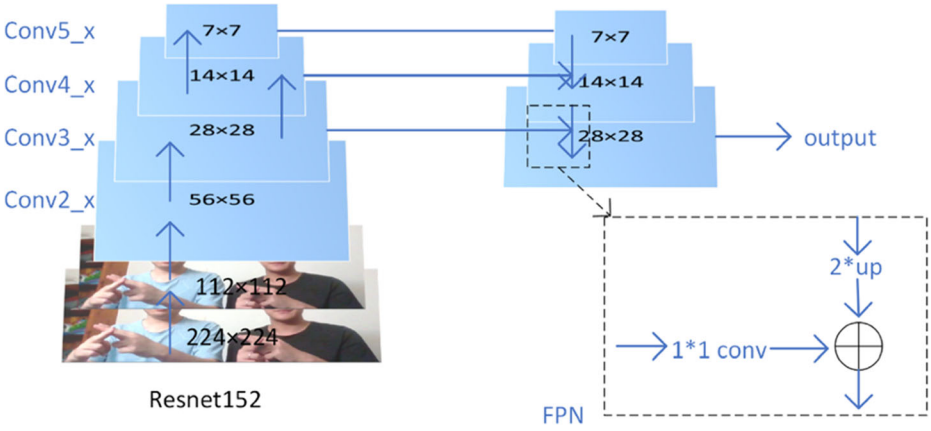


Fig. 2 The structure of ResNet152 + FPN

Therefore, the structure of FPN is designed with a top-down structure and horizontal connections to fuse the shallow layer with high resolution and the deep layer with rich semantic information. Therefore, constructing a feature pyramid with strong semantic information on all scales from a single input image on a single scale is possible without incurring high costs. Thus, FPN can enhance the feature extraction of ResNet152 at multiple scales. Besides, an RTX 3060 GPU accelerates the training process to achieve computational efficiency.

3 Methodology

In order to improve the accuracy and speed of the proposed methodology for sign language recognition, in this paper, we make use of ResNet152 to replace the ResNet50 as a backbone. Its aim is to increase convolutional layers and 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 that can be connected 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, ResNet50 has 6 blocks. Thus, ResNet152 can get better results. Equation (1) is used to calculate the size of the feature map,

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

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 is 224 × 224 pixels. After downsampling 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

$$S_{(i,j)} = (X \times V) \sum_{M} \sum_{N} x(i+m, j+n) v(m,n) \tag{2}$$

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	
		[1×1,64; 3×3,64; 1×1,256] ×3	[1×1,64; 3×3,64; 1×1,256] ×3
Conv3_x	28×28	[1×1,128; 3×3,128; 1×1,512] ×4	[1×1,128; 3×3,128; 1×1,512] ×8
Conv4_x	14×14	[1×1,256; 3×3,256; 1×1,1024] ×6	[1×1,256; 3×3,256; 1×1,1024] ×36
Conv5_x	7×7	[1×1,512; 3×3,512; 1×1,2048] ×3	[1×1,512; 3×3,512; 1×1,2048] ×3

Fig. 3 The structure of ResNet50 and ResNet152

where x is the variable of the input image, v is the convolution kernel, $M \times N$ is the size of the input image [29]. 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.

In FPN nets, the convolution kernel calculates the feature map, the feature map usually becomes smaller than the last few layers. However, multiple feature layers whose output is as same as the original size, are called the same network stage. For the feature pyramid in this paper, a pyramid level is defined for every stage, and the output of every stage at the last layer is selected as the reference of feature maps. The choice is usually natural. The reason is that there should be the most robust features in the deepest layer for each stage. Specifically, pertaining to ResNets, we take use of the output of residual structure of each stage, denote these residual module outputs as C2, C3, C4, C5 corresponding to the outputs of conv2,

conv3, conv4, and conv5. We know that they have steps 4, 8, 16, 32 related to the input image. Conv1 is not included in this pyramid framework by considering the memory footprint.

Regarding 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 $c_\rangle$ representing the class of the visual object, the other is $b_\rangle$ which shows the bounding box of the object. The prediction probability is $\widehat{p}_{\widehat{\sigma(\rangle)}}(c_\rangle)$. In Fig. 4, $N = 5$ is set as an example. This satisfies eq. (3), and the loss function of this optimization is calculated by using eq. (4).

$$\widehat{\sigma} = argmin_{\sigma}\sum_{i}^{N}\mathcal{L}_{match}\left(y_i, \widehat{y}_{\sigma(i)}\right) \tag{3}$$

$$\mathcal{L}_{Hungarian}\left(y, \widehat{y}\right) = \sum_{i=1}^{N}\left[-log\widehat{p}_{\widehat{\sigma(i)}}(c_i) + 1_{\{c_i \neq \varnothing\}}\mathcal{L}_{box}\left(b_i, \widehat{b}_{\widehat{\sigma}}(i)\right)\right] \tag{4}$$

4 Experimental results

In this article, we utilize our own dataset for model training and testing to get the stellar performance of experimental results. There are 8600 frames in total, 6450 frames were selected for the training section, 2150 frames were picked up for the testing. Another dataset contains 12 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 created and collected by ourselves. The total number of images is 7192 in this dataset, 5000 frames were employed for the model training, 2192 frames were picked for the testing. Figure 5 shows the gesture samples for the nine classes. Fig. 6.

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 is the proportion of true examples that should be predicted as positive, calculated by using eq. (5). As shown in eq. (6), $TP + FN$ is the number of all positive samples, recall (recall rate) is the proportion of all

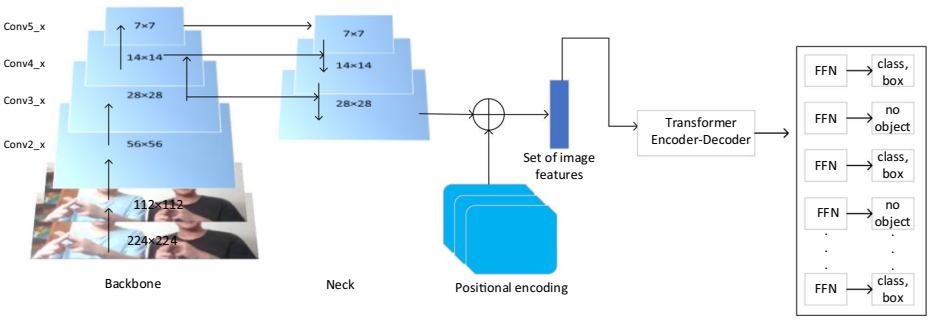


Fig. 4 The structure of ResNet152 + FPN + DETR

Fig. 5 The samples of our own sign language dataset

positive samples that are correctly predicted. In our experiments, the average precision is calculated by using eq. (7).

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$AP = \frac{TP + TN}{TP + TN + FP} \tag{7}$$

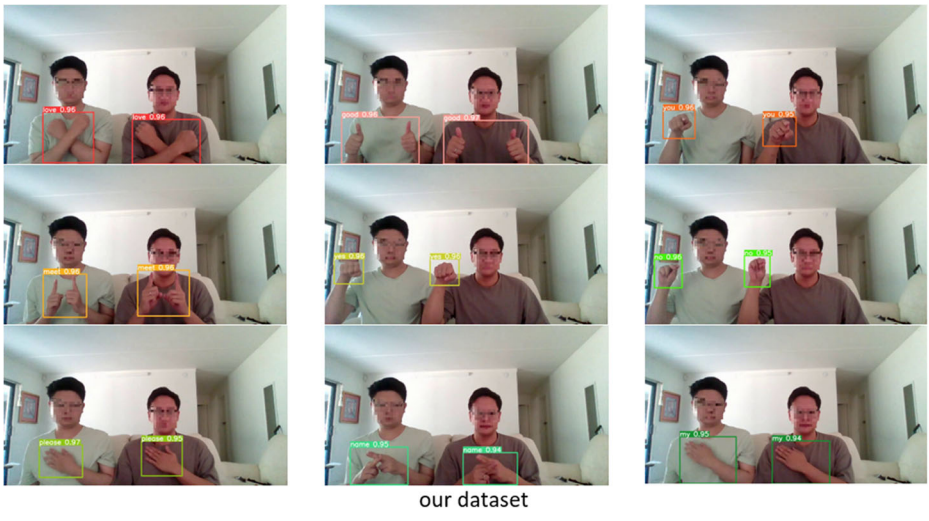


Fig. 6 The results of sign language recognition

Table 1 The deep learning Models for sign languagr recognition

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

In Table 1, TP, TN, FP, and FN are mainly employed to count two types of classification problems, and multiple classes were counted separately. The samples are split into positive samples and negative samples. The first letter in TP, TN, FP, and FN indicates whether the recognition result of the classifier is correct.

The focus of this paper is mainly on the proposed deep learning methods based on DETR and its impact on the result. We mainly emphasized on four state-of-the-art backbones to fulfill the sign language recognition, which are ResNet34, ResNet50, ResNet101, and ResNet152 + FPN. In Fig. 7, we demonstrate the result of sign language recognition from the video frames.

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 1, we compare our deep learning models for sign language recognition using our dataset.

Fig. 7 The examples of our implemented methods

As shown in Table 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%. Comparative experiments show that the new method improves the detection accuracy 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}.

In Table 2, our proposed method shows excellent results for sign language recognition. We are able to obtain 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.

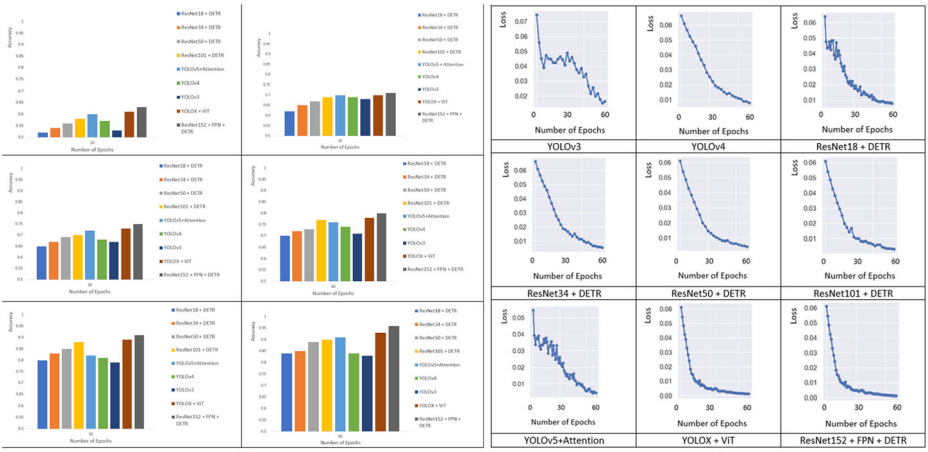
From DETR, we see the structure as shown in Fig. 3, the convolution blocks and convolution kernels are increased step by step from ResNet18 to ResNet152. With the increase of convolution kernel, the feature map increases accordingly, the accuracy thus has been improved.

Figure 8 shows the accuracy and validation losses. The black bar represents the proposed method. All the methods get the maxima, 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 Table 2 and Fig. 8, we see that the proposed method has a better recognition rate that can reach 28 FPS compared to existing methods due to its jump connection structure to avoid gradient vanishing problem. ResNet152 as the feature extraction network, contains more feature information and more semantic information in the upper layer of the feature map. Combined with the FPN structure to fuse high-level and low-level information, ResNet152 is able to improve the average accuracy of 96.45%. For our proposed method as well as the compared methods, we employed an RTX 3060 GPU and AMD Ryzen 55600H CPU to accelerate the training and detecting process to achieve better computational efficiency.

Table 2 The comparison of deep learning Models

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%**



(a) (b)

Fig. 8 The accuracy rates (**a**) and losses (**b**) of our implemented methods

5 Conclusion and future work

In this article, we employed ResNet152 + FPN + DETR model to achieve a superior performance for sign language recognition. The experimental results show that the new model has better results compared to the existing methods, which attained 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 a lot compared to the previous method, this problem can still be continuously improved in the future [2]. Besides, applying the FPN nets to the DETR-based models shows great betterment in sign language recognition.

The limitation of this work is that the data corpus and the size of the dataset are limited because we created our own dataset. The complexity of this model is higher than the previously proposed recurrent neural network (RNN), however this can be easily compensated using more GPU power.

In our future work, we will combine YOLO model and Transformer to obtain better results, which in turn will uplift the overall performance of sign language recognition. In addition, we intend to expand the dataset for a much wider range of vocabulary to increase the validation of our experiments so far.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations This work was supported by our school's summer research grant, it has not any conflicts of interests or competing interests.

References

1. Bastanfard A, Rezaei NA, Mottaghizadeh M, Fazel M (2010) A novel multimedia educational speech therapy system for hearing impaired children. Springer, pp. 705–715
2. Bauer B, Hienz H, Kraiss KF (2000) Video-based continuous sign language recognition using statistical methods. In: International Conference on Pattern Recognition (ICPR), pp. 463–466
3. Bauer, B., Hienz, H., Kraiss, K. (2000) Video-based continuous sign language recognition using statistical methods. In: International Conference on Pattern Recognition (ICPR)
4. Bhatti UA, Huang M, Wu D, Zhang Y, Mehmood A, Han H (2019) Recommendation system using feature extraction and pattern recognition in clinical care systems. Enterprise Inform Syst 13(3):329–351
5. Camgoz NC, Koller O, Hadfield S, Bowden R (2020) Sign language Transformers: Joint end-to-end sign language recognition and translation. arXiv: 2003.13830
6. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with Transformers. arXiv: 2005.12872
7. 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
8. Duarte A (2019) Cross-modal neural sign language translation. In: IEEE International Conference on Multimedia and Expo
9. 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
10. Ko SK, Kim CJ, Jung H, Cho C (2019) Neural sign language translation based on human keypoint estimation. Appl Sci 9(13):2683
11. 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
12. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105
13. Liu J, Kuipers B, Savarese S. (2011) Recognizing human actions by attributes, In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3337–3344
14. Liu Z, Zhang C, Tian Y (2016) 3D-based deep convolutional neural network for action recognition with depth sequences. Image Vis Comput 55:93–100
15. Minoofam SAH, Bastanfard A, Keyvanpour MR (2022) RALF: an adaptive reinforcement learning framework for teaching dyslexic students. Multimed Tools Appl 81:6389–6412
16. 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
17. 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
18. Orbay A, Akarun L (2020) Neural sign language translation by learning tokenization. arXiv:2002.00479
19. Özdemir O, Camgöz NC, Akarun L (2016) Isolated sign language recognition using improved dense trajectories. In: Sig Proc Commun Appl Conf (SIU)
20. 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
21. Rastgoo R, Kiani K, Escalera S, Sabokrou M (2021) Multi-modal zero-shot sign language recognition. arXiv: 2109.00796
22. Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

23. Starner T, Pentland A (1997) Real-time American sign language recognition from video using hidden Markov models. In: Shah M, Jain R (eds) Motion-based recognition. Computational Imaging and Vision, vol 9, pp 227–243
24. Süzgün M et al (2015) Hospisign: an interactive sign language platform for hearing impaired. J Naval Sci Eng 11(3):75–92
25. Tamura S, Kawasaki S (1988) Recognition of sign language motion images. Pattern Recogn 21(4):343–353
26. 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
27. Vaswani A, Shazeer N, Parmar N, Yang L, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. arXiv: 1706.03762
28. 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
29. Xiang N, Pan C, Li X (2021) An object algorithm combining FPN structure with DETR. In: ACM ICCCV, pp. 57–63
30. 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
31. Yin, K. (2020) Sign Language translation with Transformers. arXiv:2004.00588
32. Yin K, Read J (2020) Better sign language Translation with STMC-Transformer. In: International Conference on Computational Linguistics, pp. 5975–5989
33. Zhou D, Kang B, Jin X, Yang L, Lian X, Jiang Z, Hou Q, Jiashi FJ (2021) DeepViT: Towards deeper Vision Transformer. arXiv: 2103.11886
34. Zhu Y, Long Y, Guan Y, Newsam S, Shao L(2018) Towards universal representation for unseen action recognition, In: IEEE Conference on Computer Vision and Pattern Recognition