Chinese Sign Language Recognition Model Based on YOLOv5 and MediaPipe

Zixi Zhou
23011611
14/03/2024

Introduction

When deaf people communicate with hearing people, hearing people often do not understand sign language. While for deaf people, written language is their second language so they are not able to express themselves as accurately as hearing people. Therefore, sign language interpreters are very valuable for communication between deaf and hearing people. With the development of artificial intelligence, many machine learning models can recognise and understand gestures after extensive training, providing a solution for sign language interpretation.

Although China has the largest number of deaf people in the world, there is still much room for development in the field of Chinese sign language interpretation. Therefore, this project trains gesture recognition models based on YOLOv5 and MediaPipe, two models with wide applications in the field of human pose recognition, and a small self-made Chinese sign language dataset, in order to compare the strengths and weaknesses of each model in the field of Chinese gesture recognition, and to explore the possibility of realising Chinese sign language translation.

Background

In China, to communicate with hearing people who do not understand sign language in work and life, deaf people use written Chinese, but most deaf people follow the grammar of sign language when using written Chinese, which makes it confusing for hearing people to understand and leads to misunderstandings. Due to their hearing impairment, deaf people have great difficulty in learning pinyin and therefore often make mistakes when using pinyin typing techniques. Sign language interpreting techniques are extremely important in breaking down communication barriers between deaf and hearing people. Machine learning offers a solution to sign language interpretation technology by training neural networks that allow computers to recognise hand gestures.

YOLO (You Only Look Once) is one of the most popular algorithms for object detection. It takes an image as input and then uses a simple deep Convolutional Neural Network (CNN) to detect objects in the image [1], which offers highly accurate recognition results and fast processing speed. YOLOv8 and YOLOv5 are two of the most popular and advanced models created by

Ultralytics. While YOLOv8 performs noticeably better during training and validation to some extent, YOLOv5 is more easily for practical applications [2]. Therefore, YOLOv5 was chosen for this project.

MediaPipe is a a versatile framework developed by Google, it offers pre-trained models that excel at detecting and tracking human body landmarks and poses [3]. MediaPipe Hands is another a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame [4]. With these landmarks, the model can infer the pose and shape of the hand, enabling it to predict gestures. It is a lightweight ML model that also maintains accuracy, I chose it as another technological support for this project.

In 2022, a conversation between a deaf food delivery driver and a customer was posted on Chinese social media. The driver's words in the conversation seemed rude, causing confusion for the customer. The reason for this was the inability of deaf people to use Chinese correctly, which highlighted the communication difficulties between deaf and hearing people. Therefore, I self-learned some sentences that deaf food delivery drivers are likely to use in their work scenarios and create a small hand gesture dataset for training. I hope this project can help deaf food delivery drivers communicate more effectively at work and explore the possibilities of the two technology models in practical applications.

Method

Collect images

There is no open-source Chinese sign language dataset available online. Therefore, I have listed three sentences that deaf food delivery drivers are most likely to use in their work scenarios: "Hello, your delivery is here"; "Sorry, I will be late"; "Thank you, could you give me a good rating". I self-learned these three sentences in sign language from the internet and simplified them into nine gestures: "arrive", "food", "you", "good", "give", "I", "can", "sorry", and "thank you". I used the "collect images from webcam" [5] script introduced in week 4 of this course to grab images from my computer's webcam.

Figure 1: Nine gestures deaf food delivery drivers are most likely to use

In practice, the two gestures "give" and "can" involve positional changes in space, while the two gestures "sorry" and "thank you" involve both hands, and the two gestures "you" and "I" differ only in direction. Considering the difficulty of training the YOLO model, the only gestures used for YOLO training are "arrive", "food", "you" and "good", each with 15 original images. The gestures used for MediaPipe training contain all nine types of gestures, with 200 images of each type.

Image processing and data augmentation

MediaPipe allows for direct training using the collected images, whereas YOLOv5 requires pre-processing and data augmentation for the raw images.

First, I manually annotated each raw image, using the labelme [6] tool. After removing some unclear photos, I annotated the coordinates of the hands in each image one by one and named different gestures as different categories.

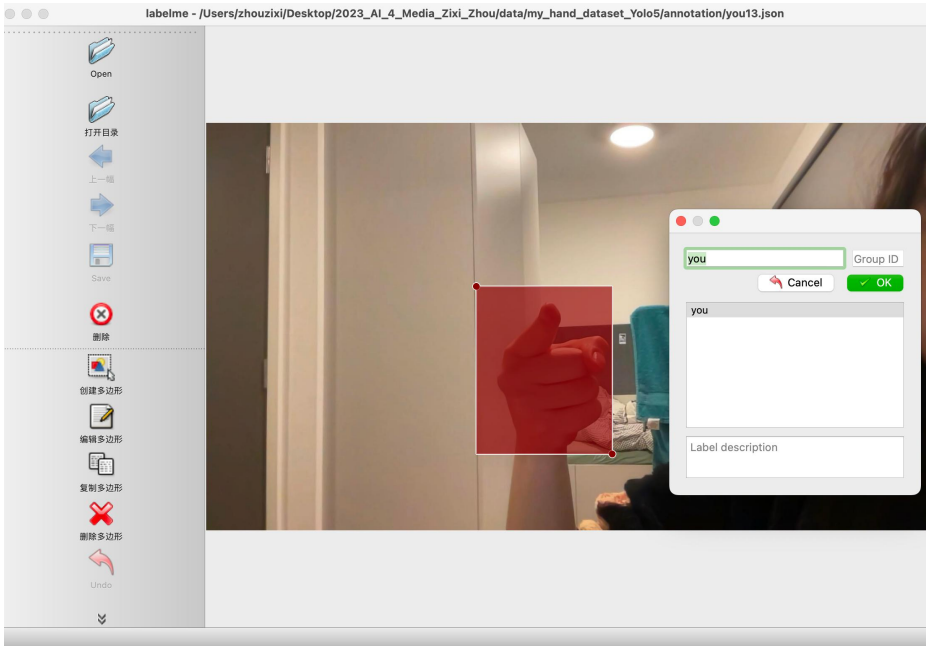

Figure 2: labelme

After the annotation is complete, labelme generates a JSON file for each image, but this format is not available for YOLOv5. I refer to the blog "Tutorial on Converting json to txt format in LabelMe Annotation" [7] and just keep the category, hand relative coordinates in the annotation file, create a new coordinate file for each original image with the format '.txt'.

Since annotating hundreds of images for each category is impractical due to the significant cost involved, I considered data augmentation, using the Albumentations library to perform operations such as resizing, padding, moving, scaling, rotating, and others on the original images.

I referred to the blog "Interactive ABCs with American Sign Language using Yolov5" [8] and applied 20 augmentations to each original image. This process resulted in 300 training images for each category. I divided the augmented data into training and validation sets at a ratio of 9:1.

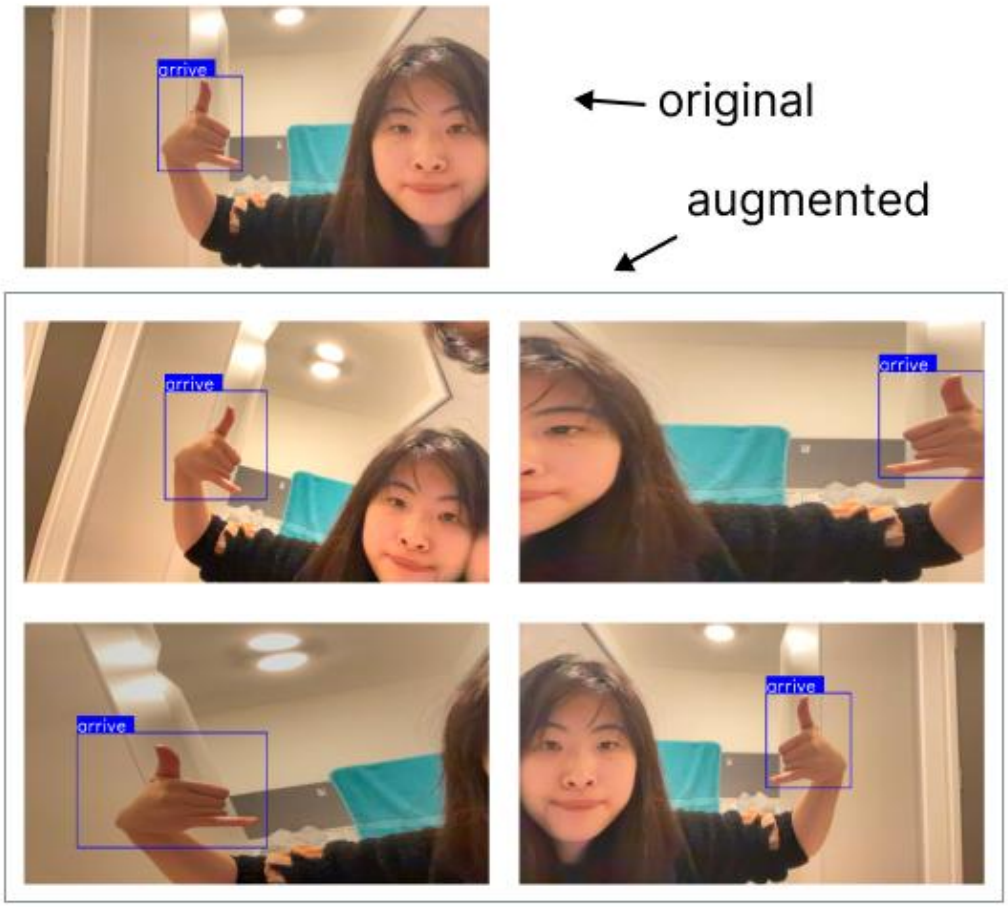

Figure 3: Data augmentation example

Train Yolov5

I followed the official Ultralytics documentation for training YOLOv5 [9]. I created a 'Chinese_Gesture_yolo' file to define training and validation data and to specify classes. Given the performance limitations of my laptop, I decided to use the lightweight yolov5s model. As this model accepts images with a maximum size of 640 pixels, I resized my images from 1920 to 480 pixels accordingly. I also increased the batch size to 16 to speed up the model's data processing. I used the GPU available on the Google Colab platform for training and recorded the model's training results at epochs 5, 20 and 50.

Train MediaPipe

The training of MediaPipe is very simple, I used the default parameters of the model, trained on Google Colab platform using CPU and recorded the model's training results at epochs 10.

Results

YOLOv5 training results

When the number of epochs is 5, 20, and 50, respectively, the parameter changes in training are shown below:

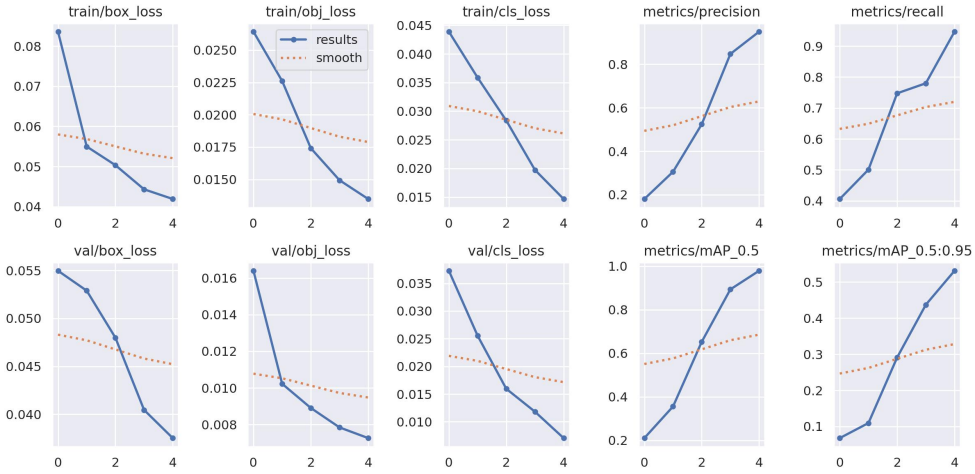

Figure 4: yolov5 results (epoch =5)

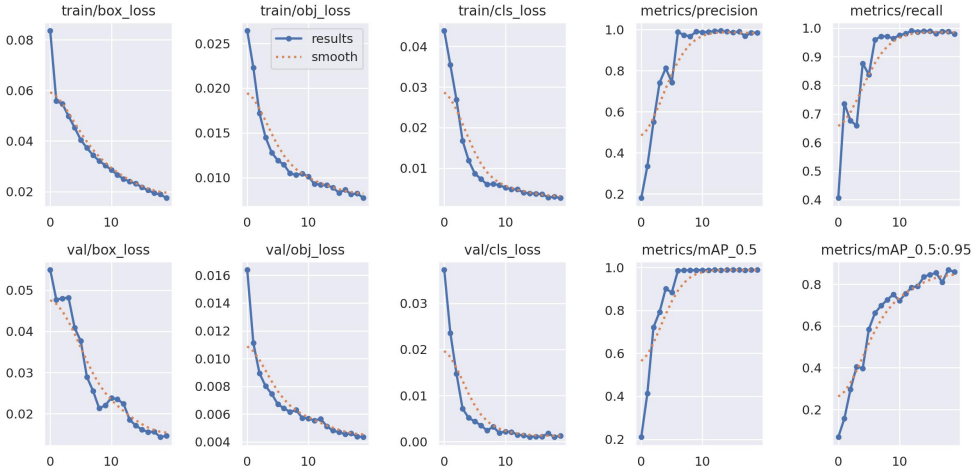

Figure 5: yolov5 results (epoch =20)

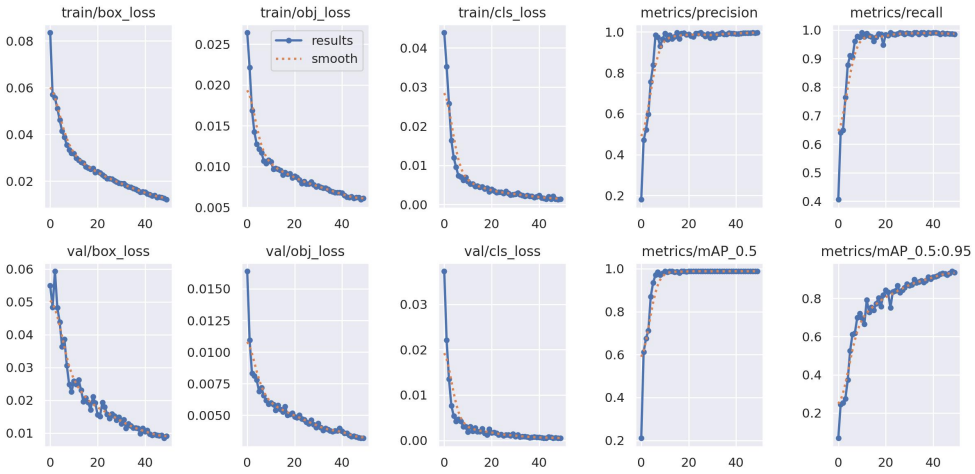

Figure 6: yolov5 results (epoch =50)

YOLOv5 best testing results

The training model with epoch of 50 is used for testing and the test results obtained are shown in Fig 7:

Figure 7: yolov5 test results (epoch =50)

MediaPipe best testing results

The test results obtained are shown in Fig 8:

Figure 8: MediaPipe test results

Discussion

As can be seen in Figure 4, at epoch 5, the model is still in the early stages of training and the loss value continues to decrease, but has not yet reached a stable state. The mean accuracy (mAP) of object detection is only 0.5 in the range of IoU from 0.5 to 0.95, indicating that the performance of the model needs to be improved.

As the epoch increases to 20, the model has a deeper understanding of the training set and there is an obvious inflection point in the loss curve. At this point, the optimal precision and recall are gradually approaching 1, but there are still some fluctuations, indicating that the model performance is not yet fully stabilised.

As the epoch increases to 50, the loss curve becomes very smooth and there is an obvious inflection point. At this point, the mean accuracy of object detection (mAP) increases dramatically in the IoU range from 0.5 to 0.95, peaking at around 0.9. At the same time, precision and recall gradually approach 1, indicating a significant improvement in the model's performance on the training set. The inflection point occurs at roughly the same location as when the epoch is 20, and this location should be a critical point for the model to be trained on my small dataset.

During the test, I observed that the model performed consistently on the recognition of all four gestures with an accuracy of 0.95 or more, which proves that the training was satisfactory. However, I also noticed a new problem: the model's accuracy is higher when the test background is relatively blank, but the model's recognition ability decreases when other objects appear in the test background. This could be because my datasets were all taken against the same background, causing the model's understanding of hand information to be too dependent on a specific background. This also hints at the risk of overfitting, where the model performs well in specific conditions but poorly in changing test conditions.

In contrast, MediaPipe's performance may not be as impressive in terms of accuracy, but it is less affected by context and is able to recognise two-handed data. This is because MediaPipe focuses on the key point information of the hand, it first determines the position of the hand and then recognises the type of gesture. YOLOv5, on the other hand, is trained based on hand bounding box information in the image and is therefore more affected by noise.

Considering both, training with YOLOv5 is more costly, but its training accuracy is also higher.The training of YOLOv5 requires a large amount of datasets and attempts to denoise the original images during the data labelling stage, which is a process that is much more difficult than the amount of work involved in the training itself. Therefore, I believe that in creating a highly accurate sign language translation system, there is a need for more support from various parties, such as the government, relevant charities and professional research institutes, to work together to collect and optimise Chinese sign language databases.

The lightweight nature of MediaPipe makes it easier to integrate into mobile devices, such as mobile phones, which makes it more suitable for smaller datasets. Its lightweight and flexibility advantages make it more advantageous in limited scenarios when accuracy is not particularly required.

Conclusion

This project compares the performance of two different target detection models, YOLOv5 and MediaPipe, on a small sign language dataset.YOLOv5 performs better in terms of accuracy, while MediaPipe is more flexible. In practical applications, YOLOv5 is more stable but also more costly if we want to build a large-scale high-precision sign language recognition system. For several sign language recognition functions in specific scenarios, MediaPipe is easier to apply, although its accuracy may not be as high as YOLOv5.

Ethical considerations

This project strictly adheres to the ACM Code of Ethics [10] in computing. As this dataset was collected manually by me and contains personal and private information such as my face and hands, I do not intend to release it publicly and it will only be used for my personal research purposes. This has also made me aware of the risk of privacy breaches involved in the collection of human data, and has made me think about how to better protect personal information when using human data. Despite the challenges, I believe that collecting and creating open source Chinese gesture data is very rewarding.

LLM disclaimer

A very small number of code blocks from ChatGPT were used in this project with appropriate comments in the source code. ChatGPT was used for debugging and troubleshooting during the project. Throughout the writing of this document, DeepL, ChatGPT were used to optimise the Chinese-to-English translation work.

Bibliography

[1] Kundu, R. (2023) *YOLO: Algorithm for Object Detection Explained [+Examples].* Available at: https://www.v7labs.com/blog/yolo-object-detection (Accessed: 15/03/2024).

[2] Tyagi, S., Upadhyay, P., Fatima, H., Jain, S., & Sharma, A. K. (2023) 'American Sign Language Detection using YOLOv5 and YOLOv8'. Available at: https://doi.org/10.21203/rs.3.rs-3126918/v1 (Accessed: 15/03/2024).

[3] MediaPipe (2024) *Pose landmark detection guide.* Available at: https://developers.google.com/mediapipe/solutions/vision/pose_landmarker (Accessed: 15/03/2024).

[4] MediaPipe Team (2023) *MediaPipe Hands.* Available at: https://github.com/google/mediapipe/blob/master/docs/solutions/hands.md (Accessed: 15/03/2024).

[5] IriniKlz. (2024) *AI-4-Media-23-24/Week-4-Sensing-bodies.* Available at: https://git.arts.ac.uk/tbroad/AI-4-Media-23-24/blob/main/Week-4-Sensing-bodies/00-collect-images-from-webcam.py (Accessed: 15/03/2024).

[6] wkentaro. (2024) *labelme.* Available at: https://github.com/labelmeai/labelme (Accessed: 15/03/2024).

[7] Non-destructive testing novice (2021) *LabelMe labeled json to txt format conversion tutorial.* Available at: https://blog.csdn.net/qq_42046837/article/details/120278209 (Accessed: 15/03/2024).

[8] Lee, D. (2020) *Using Computer Vision in Helping the Deaf and Hard of Hearing Communities with Yolov5..* Available at: https://daviddaeshinlee.medium.com/using-computer-vision-in-helping-the-deaf-and-hard-of-hearing-communities-with-yolov5-7d764c2eb614 (Accessed: 15/03/2024).

[9] Ultralytics (2024) *yolov5.* Available at: https://github.com/ultralytics/yolov5 (Accessed: 15/03/2024).

[10] Association for Computing Machinery (2018) *ACM Code of Ethics and Professional Conduct.* Available at: https://www.acm.org/code-of-ethics (Accessed: 15/03/2024).