Deep Learning Hand Poses

Author: CHEN Zihao
Supervisor: Dr. Dirk Schnieders
Group Members: CHEN Zihao, DA Yujia, ZHANG Yuqian

April 14, 2019
Abstract

In recent years, hand pose research has brought new potential in Human-Computer Interaction as well as Augmented Reality. However, existing models usually fail to recognize sequential hand actions and categorize the poses, because of the high complexity of hand poses as well as human’s labeling inaccuracy. The whole project, separated into three stages, revises the models from the literature as well as implements new ideas. This report focuses on the individual part, Stage 3, pose estimation. Currently, the model is capable of classifying hand poses from image, video and live-demo efficiently. Future studies may focus on extending the scope of this model to implement applications that brings convenience to both human-to-human communication as well as human-computer interaction.
Acknowledgements

I thank Dr. Schnieders, for his supervision in the project, his comprehensive suggestions and the computational resources he offers. I would also like to show our gratitude to Julia Da and Ivy Zhang, my project group mates, who kindly offer their helping hand when I am in need, and give strong technical support throughout the project. I also want to express my gratefulness to Samson Dai and Vera Kang, who voluntarily participated in creating the Joint Pose Dataset.
# Table of Contents

Abstract i

Acknowledgements ii

List of figures vi

List of tables vii

1 Introduction 1
   1.1 Background ......................................................... 1
   1.2 Purpose of the project .............................................. 1
   1.3 Report outline ....................................................... 4

2 Literature review 5
   2.1 Theoretical background .............................................. 5
   2.2 Existing Model Review ............................................... 5
   2.3 Review of datasets .................................................. 7
   2.4 Summary ............................................................... 9

3 Methodology and Approach 10
   3.1 Project pipelines .................................................... 10
   3.2 Environment setting ................................................ 11
   3.3 Data preparation ..................................................... 11
   3.4 Model implementation .............................................. 12
   3.5 Evaluation and metrics ............................................. 13
   3.6 Summary ............................................................... 14

4 Experiments and Results 15
   4.1 Classification on Existing Cropped Datasets ................. 15
      4.1.1 Data Preprocessing and Augmentation ................... 15
      4.1.2 Model Construction and Evaluation ..................... 15
4.1.3 Obstacles and Solutions ................................................ 17
4.2 Segmentation Dataset from Background Masking .................. 18
  4.2.1 Dataset Generation ...................................................... 18
  4.2.2 Classification with Grey Level Image ............................... 20
  4.2.3 Evaluation and Analysis .............................................. 20
4.3 Extending Application on Segmented Hands ......................... 21
  4.3.1 Training on Segmented Hand .......................................... 21
  4.3.2 Training on Hand Masks .............................................. 22
  4.3.3 Potential of Mask Application ....................................... 23
4.4 Classification Based on Joint Recognition ........................... 24
  4.4.1 Intuition .............................................................. 24
  4.4.2 Joint Pose Database .................................................. 25
  4.4.3 Data Modeling ......................................................... 26
4.5 Joint Threshold and Kalman Filter .................................... 29
4.6 Experiment Summary ...................................................... 29
5 Future Works ................................................................. 31
  5.1 Future plans ............................................................... 31
  5.2 Schedule ................................................................. 32
6 Conclusion ................................................................. 33
## List of Figures

1. The 21 joint and fingertips on the hand ........................................ 2
2. the static postures of this project ........................................ 2
3. An example of the pipeline with hand pose victory, from a raw input image to the posing type. Stage 1 hand detection takes in input data and crops out the hand image. Stage 2 joint recognition labels all the 21 joints on the output image from Stage 1. Stage 3 Pose Estimation takes in both outputs from Stage 1 and Stage 2, and predicts the pose Victory. ........................................ 3
4. (a) shows the pose definition in ASL. (b) demonstrates the definition of each digit (c) is an example in HGD, which has a pure black background, convenient for data augmentation. ........................................ 9
   4.a ASL ........................................ 9
   4.b digit definition ........................................ 9
   4.c HGD ........................................ 9
5. In the ASL Alphabet dataset, this kind of poor illumination condition often takes place. They have a negative influence on the data quality ............................ 9
6. In the Finger Spelling Dataset, some images have damaged pixels ............. 10
7. (a) sample data from Hand Gesture Dataset (b) augmented data with background replacement and random direction (c) augmented data with Gaussian noise ........ 16
8. Train and validation accuracy of ResNet50 ..................................... 16
9. These images all almost identical. Splitting these data to train and validation set may result in exaggerated high validation accuracy .............................. 18
10. This is an attempt to use RGB threshold. It is difficult to filter background without affecting the hand .................................................. 18
11. Generating segmented hands with a precise mask ................................ 19
12. This mask is not so successful since some background noises are not excluded. However, as the hand is complete, this data is acceptable .......................... 19
This is a screen shot from Stanford CS231N. From left to right, higher and higher dimensional features are shown. The features on the left can be understood by human vision, while the columns on the right side are not.

This is the labeling tool. We use mouse to drag joints to the correct position and they will be saved in a json file.

The left image is a 3D demonstration of Gaussian heatmaps. The right image shows the Gaussian maps that Convolutional Pose Machine generates at each of the six stages.

This is a binary map, representing the joint skeleton in the current experiment.

This is a hand mask generated in Chapter 4.2.1.

Both train and validation loss converge in fifty epochs.

The widely applied Kalman filter values.
## List of Tables

1. Milestones and their descriptions .................................................. 2
2. Overview of present hand classification datasets .......................... 8
3. The responsibility of each stage, as well as input and output flows .... 10
4. Although the input size decreases at each phase, the accuracy actually rises. This is because higher percent of useless information in pixels are discarded . . . 23
5. Schedule ..................................................................................... 32
1 Introduction

1.1 Background

Hand Pose Estimation, a relatively new topic in the field of Artificial Intelligence and Computer Vision, aims at recognizing hand poses in visual materials, including images and videos. More and more modern research achievements are reached in this Computer Vision task, which already shows great potential in human’s interaction with computers, robust medical diagnosis approaches, as well as enhancing the experience of 3D entertainment. Therefore, it is now one of the most popular topics and has aroused considerable attention from the academic and the industrial. Present approaches frequently depend on both RGB Images and Depth Images, which requires a more expensive Depth Camera, compared with regular cameras on laptops or phones. Therefore, doing hand pose estimation task without a depth camera will considerably reduce the cost and promote its practicability.

1.2 Purpose of the project

The purpose of the whole project will be introduced to provide an overview. After that, the individual part will be discussed, to verify the subgoal and its value.

This project aims to achieve highly accurate hand pose estimation with Deep Learning, focusing on several specific hand gestures, both dynamic and static ones, and constructing a deep learning model capable of successful identification on these poses. The model’s capability is limited to detect only full hands shown in the image. More specifically, occluded joints are allowed from the view of the camera, but the physical location of each joint should be within the boundary of the image; we assume that hands are not to from the disabled, so that, each hand in the image has five fingers and 21 joints, as shown in figure 1; the hands in the image should be at least clear enough for human vision to understand. With these restrictions, given an input visual resource, our model will return the location of hands, each of its joints, and categorize each hand pose...
into a predefined type. Figure 2 illustrates the static postures that this project propose to use, and their dynamic movements are also included.

![Figure 1: The 21 joint and fingertips on the hand](image1)

Figure 1: The 21 joint and fingertips on the hand

![Figure 2: the static postures of this project](image2)

Figure 2: the static postures of this project

Since there are several types of visual resources that we plan to train and test on, including images, videos, and live demo, we determine to set three milestones for the project step by step. Remarkably, a video with 20 FPS (Frame Per Second), can be considered as 20 sequential images per second; live demo, meaning a video in real time, can as well be decomposed to an image sequence. Considering this logical dependency and sorting level of intricacy in ascending order, we define the milestones in 1. M2 and M3 are based on M1, and they share almost identical algorithm as M1. We regard M1 as the trunk, while M2 and M3 are like the branches extended from M1. Hence, as a significant part, more specific objectives in M1 will be explained next.

Table 1: Milestones and their descriptions

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Accurately identify a static hand pose and its joints in a single RGB Image</td>
</tr>
<tr>
<td>M2</td>
<td>Reduce the oscillation in joint labelling on the sequential images and identify poses on each static image with accordant accuracy from M1</td>
</tr>
<tr>
<td>M3</td>
<td>Improve the prediction speed so that it can support live demo with equivalent accuracy from M2</td>
</tr>
</tbody>
</table>
To reduce the complexity of the M1, we break it down further into three pieces, defined as Stage 1, 2 and 3. In Stage 1, hand detection, accurate location of hands on the image will be found. In Stage 2, joint recognition, starting with cropped hand-centered images output from Stage 1, predicts the position of joints. In Stage 3, pose estimation, outputs from both Stage 1 and Stage 2 are utilized to determine the exact meaning of the pose. Figure 3 shows a sample pipeline on M1 with a victory pose. The reason for Stage 3 to take inputs from both side is to mitigate the risk accumulated at each stage: as each stage has some possibility of failure, along a single forward linkedlist, errors from earliest stages may be enlarged in latter stages.

I am responsible for Stage 3, pose estimation. Therefore, in my individual project and this report, Stage 3 is the focus. Utilizing different types of input resources produced from Stage 1 and Stage 2, my purpose is to categorize the poses according to predefined rules. In the whole process, efficiency and accuracy are the two essential factors.

Figure 3: An example of the pipeline with hand pose victory, from a raw input image to the posing type. Stage 1 hand detection takes in input data and crops out the hand image. Stage 2 joint recognition labels all the 21 joints on the output image from Stage 1. Stage 3 Pose Estimation takes in both outputs from Stage 1 and Stage 2, and predicts the pose Victory.

1.3 Report outline
All the essential information will be covered in the following chapters. In Chapter 2, a literature review will be given, which delivers the theoretical background of our project as well as available resources including datasets and existing models. In Chapter 3, the methodologies and approaches of the project will be explained in details. Experiments and results will be introduced about in Chapter 4, followed by future work in Chapter 5. At last, a conclusion will be given.
2 Literature Review

This chapter will introduce the backgrounds of theories, as well as the datasets and the models I have reviewed.

2.1 Theoretical background

This project applies deep learning models. Models are usually constructed with Convolutional Neural Network (CNN), which keeps a series of large matrices in particular structure called layers, and stores a large amount of information. With matrix operations and self-refining functions in CNN, the model can learn and adjust by himself. Data and label are the fundamental concepts in deep learning. In our case, data refer to images, just like some multiple-choice exercises, while label, indicates the correct result to this data, is similar to the standard answers to the corresponding “exercises”. Training means giving these practice questions and answers (training data and label) to the model and teach it to learn by itself; testing means to provide the model with a test to evaluate its performance. Training loss calculates how far away the prediction is from the label. It can be regarded as punishment on incorrect predictions and thus forces the model to modify itself to a direction that reduces the loss most.

In Deep Learning, training process helps to enhance a model’s performance on a specific task, and the testing process verifies how well the model is doing. With different approaches to define the model and prepare the training data, the model will show dramatically difference in task accuracy and time consumption. Therefore, during the training process, parameters, for example, the input image size, are fine-tuned and retrained repeatedly until achieving acceptable performance in the testing.

2.2 Existing Model Review

The literature includes three parts, i) typical classification models, ii) gesture-related works, iii) image processing approaches.
I researched on ResNet[10], who applied residual blocks and ranked the first in ImageNet Large Scale Visual Recognition Competition(ILSVRC)[6] in 2015. The residual block adds a short-cut as shown in . When networks extend deeply, vanishing gradient becomes a serious problem. ResNet uses the residual blocks to replenish gradient and therefore solves the problem. VGG net[11] is another widely adopted feature extraction network, which uses smaller kernels and deeper networks, and ranked second in ILSVRC 2014. They have been proved to be the most competent in classification models and will be evaluated. MobileNet[2] is another network that adapted for mobile vision applications and therefore aroused my attention. It splits the traditional convolutional layer architecture into two parts, depthwise convolution and pointwise convolution, to avoid intensive calculate and therefore promotes efficiency.

With regard to hand pose, I review Hand3d[5], which infers 3D joint location from a single RGB image. It is feasible to analyze joint features with their 3D coordinates with geometrical calculation. Additionally, this work is also a practical example of defining and integrating three stages.

Additionally, I haven’t found any effective works on pose classification on RGB image by deep learning. Researchers once preferred to use depth camera with traditional computer vision approach, marking spacial features to mark fingers. Now, with the increasing popularity of deep learning, many inexperienced researchers attempt to use classification model directly onto hand dataset, which could produce promising validation accuracy, but all fail in real-life cases. I have faced immense difficulties in finding useful sources. Without previous works in the related area, I have to conduct several experiments to find the correct direction.

Traditional computer vision approaches are also involved in this project. Although deep learning methods are the focus, traditional cv can be effective on image processing. A color space named HSV[19] has attracted my attention. In RGB color space, the three color components contribute equally to the overall color, while in HSV color space, color is only one aspect. H stands for hue, the color attribute, originated from one or two of RGB colors; S is the saturation of a pixel, related to chroma attribute; V represents the largest color component. This color space represents a more comprehensive aspect of color and is widely applied in filter background noise and generating masks.
2.3 Review of datasets

There are two types of relevant datasets. i) joint labeled dataset, which provides joints coordinates for each hand in the image data. ii) pose classification dataset, mainly related to sign languages, and defines meaning for a series of postures.

i) Rendered Handpose Dataset[8] and NYU Dataset[7] are widely used in the hand pose research field, proved to be well-diversified and comprehensive, with high image resolution and labeling quality. However, after a careful review, these datasets focus too much on joints labeling and do not stick to some fixed poses. In this classification problem, having joints on random poses is not enough; instead, joints should be labeled on predefined and stationary poses to help training. Therefore, after a careful review of the data, these datasets are not considered in pose estimation.

ii) Although we will work on a limited number of poses in this project, we collect a wide range of hand pose datasets to evaluate the competence of different models. The datasets are mainly related to American Sign Language(ASL), which defines a hand pose for each of the 26 English letters. The first contribution in this project is collecting, evaluating and utilizing existing hand classification datasets, including (1)Sign Language Mnist[13], (2)Static American Sign Language[14], (3)Columbia Gesture [9], (4)Sign Language Digits Dataset [12], (5)Hand Gesture Dataset[1], (6)ASL Alphabet[3] and (7)Finger Spelling Dataset[17].

The definition of ASL and sign language digits are illustrated in . The following table 2 is a summary of the seven datasets. (28, 28, 1) indicates the image has width and height equal to 28. 1 stands for the grey level image, and 3 for RGB image.

Although (1) is a representative and well-known dataset with over 30000 images on Kaggle, it was wiped out after I researched on Kaggle competition. This[6] simple network with a series of Conv2D and MaxPooling layers achieved over 99% validation accuracy on Sign Language Mnist within 50 epochs, but it only has 5% test accuracy when I randomly select 100 images from another dataset. The overfitting problem resulted from the low resolution of images, which are 28 * 28 grey level, and easy for models to memorize. (2) was dismissed for a similar reason (5) includes quality data with segmented hands, where backgrounds are set to pure black, with RGB value [0, 0, 0], as shown in 4. Although each class has less than 80 images, changing
Table 2: Overview of present hand classification datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>Quantity and Quality</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign Language Mnist</td>
<td>3000 for 24 ASL classes (28,28,1)</td>
<td>Large in quantity</td>
<td>Low resolution. Similar backgrounds.</td>
</tr>
<tr>
<td>Static American Sign Language</td>
<td>1680 for 24 ASL classes (64, 64, 3)</td>
<td></td>
<td>Less than 100 data per class. Low resolution.</td>
</tr>
<tr>
<td>Columbia Gesture</td>
<td>3400 for five vowel classes (4000, 2000, 3)</td>
<td>High quality data Each datum comes from different person</td>
<td>Contain invalid poses. Only have five classes</td>
</tr>
<tr>
<td>Sign Language Digits Dataset</td>
<td>2000 for 10 digit classes (100, 100, 3)</td>
<td>Data come from 200 students, high variety</td>
<td>Same white background</td>
</tr>
<tr>
<td>Hand Gesture Dataset</td>
<td>2517 for 36 ASL+digit classes (300, 300, 3)</td>
<td>The hands are segmented, easy to do augmentation</td>
<td>Each class only have 70 images</td>
</tr>
<tr>
<td>ASL Alphabet</td>
<td>3000 each for 36 ASL+digit classes (200, 200, 3)</td>
<td>Large amount of data. High resolution.</td>
<td>Some photos taken in extreme illumination condition</td>
</tr>
<tr>
<td>Finger Spelling Dataset</td>
<td>2500 each for 36 ASL+digit classes</td>
<td>Large amount of data</td>
<td>The pixels of image are damaged</td>
</tr>
</tbody>
</table>

background as data augmentation is a significant advantage. (6) has 3000 images for each pose, but some photos have extreme illumination condition, i.e., too dim or too bright to see anything from human eyes, shown in figure 5, which might lower the overall quality of datasets. (7) has an acceptable resolution, but the pixels are damaged as figure 6. After careful evaluation, (4)(5)(6) are selected as preliminary datasets.

Using multiple data sources together will improve the comprehensiveness and complexity of dataset, and enhance the competence of our model to do a real-world analysis. Current hand classes can combine to associate with dynamic poses: an ok posture can be regarded as a combination of pose ‘S’, and ‘F’; a grabbing pose as ‘5’ followed by pose ‘S’. With this approach, even if the final goal is about dynamic gestures, I can still decompose the dynamic poses to work with static poses first.
Figure 4: (a) shows the pose definition in ASL. (b) demonstrates the definition of each digit (c) is an example in HGD, which has a pure black background, convenient for data augmentation.

2.4 Summary

This chapter presented our literature review in different aspects. Essential technical terms are explained, followed by a brief introduction of the reviewed papers. At last, I reviewed several datasets and discussed their features. In the next chapter, some papers and datasets from the review will be selected and utilized in our methodology.

Figure 5: In the ASL Alphabet dataset, this kind of poor illumination condition often takes place. They have a negative influence on the data quality.
3 Methodology and Approach

This chapter will mainly discuss details of our methods, including how the group project is divided into individual parts, how the data is prepared, as well as model construction and evaluation in this individual topic.

3.1 Project pipelines

As aforementioned in Chapter 1.2, we define three stages to reduce the complexity problem. Each stage can be regarded as an independent block, which, with prepared training data and label, can be trained independently and simultaneously, and hence reduce the workload and enhance efficiency. Input and output of each stage are illustrated in table 3.

Table 3: The responsibility of each stage, as well as input and output flows

<table>
<thead>
<tr>
<th>Stage</th>
<th>task</th>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1 Hand Detection</td>
<td>locate all human hands in the images and crop the hands out</td>
<td>Full size images containing hands</td>
<td>Cropped hand-centered images</td>
</tr>
<tr>
<td>S2 Joint Recognition</td>
<td>Mark the 21 joints in each hand image</td>
<td>Output of Stage 1</td>
<td>2D coordinates of hand joints</td>
</tr>
<tr>
<td>S3 Pose Estimation</td>
<td>Associate the input data with a predefined hand pose</td>
<td>Output of Stage 1 and Stage 2</td>
<td>Hand Pose Categories</td>
</tr>
</tbody>
</table>

3.2 Environment setting
After careful consideration, I choose to work on TensorFlow. TensorFlow is an open source Deep Learning API with a wide range of efficiently implemented library functions. Notwithstanding PyTorch, another deep learning framework, is often used for research projects, I still choose TensorFlow. The reasons are as followed: Firstly, TensorFlow is currently the most popular Deep Learning framework, with the best available resources both from official support and non-official forums. Additionally, TensorFlow is such a versatile framework that, many models, though initially implemented in Caffe or Caffe2, are often re-implemented with TensorFlow by researchers, thanks to its provided convenient and comprehensive functions. This offers convenient access to compare and evaluate existing models in this platform than in PyTorch; Thirdly, TensorFlow has better parallel computing functionalities, and thus in our speed-pursuing project, it is a more prudent choice. I also use Keras, which uses TensorFlow as backend, and is more user-friendly.

Apart from the above software part, I also finish the hardware preparation. In this computation-intensive project, Graphics Processing Units (GPU) are required, otherwise with only Central Processing Units (CPU) in regular computers, all the resources and the time will be occupied by the training process. Google Cloud Platform provides us with 1 NVIDIA® Tesla® K8 GPU, which will be sufficient for me to finish this project.

### 3.3 Data preparation

Taking the limited number of data into account, I apply data augmentation approaches to gain more data based on the existing dataset. There already exist many approaches, such as global illumination, lossy image compression, specular reflection on the skin, as well as random rotation. Take random rotation as an example, I select one-tenth of the image data randomly, create new data by rotating these images in arbitrary degree from clockwise 0 to 20 degree. These approaches strengthen our model, make it adaptable to different illumination conditions, hand directions and hand sizes. I also use the Keras function ImageDataGenerator, which provides API for width and height shift, zoom, rotate and brightness[2].

The validation accuracy within a dataset alone might not be enough to prove the capability of the model. Therefore, apart from using existing dataset, I also create real-world test dataset with
less than 100 images, which are taken randomly from daily life, from different people, illumination condition, background. This dataset will be a reliable indicator of the performance of networks in the real-life analysis.

Additionally, we also build a new dataset, Joint Pose Dataset on our own, which boasts its feature of labeling poses and joints at the same time. The hand images are preprocessed by traditional computer vision approach to be segmented hand, which is easier to do data augmentation by switching backgrounds. The joints are mainly yielded by Stage 2, joint recognition. The details of how the dataset is accomplished will be discussed in Section 4.4.2.

### 3.4 Model implementation

Instead of building everything from scratch, I first search for available models, to avoid unnecessary redundancy. I look through open source websites including arXiv and GitHub for present papers and codes, evaluate the models and select one that fits in the stage best.

My pose estimation project can be categorized into a classification problem because it takes some hand images like the ones in Figure 1.1 and classifies each of them according to predefined hand pose, or an undefined hand pose, which will be classified to a background. I evaluate some top classification models, as introduced in Chapter 2.2 Literature Review, including ResNet50, VGG-16, and MobileNet. Normally a computer vision classification problem only takes images as input, in my case I evaluate on multiple data. I want to utilize both outputs from Stage 1 and Stage 2. Stage 1 provides hand-centralized images as well as segmented images, and Stage 2 offers their corresponding predicted joint locations. In my expectation, this new approach may outperform traditional classification models because more features are provided.

Additionally, this combined approach in Stage 3 has another advantage that, in video or live demo, when dynamic pose estimation is required, hand-centralized images(output of S1) helps the analysis of pose on each image, while the sequence of joints locations(output of S2) provides great assistance on detecting the change of pose. The latter part, I propose, may include pure mathematical calculations to compute the angles between fingers and the distance of movement, which is more stable than pure deep learning approach. With incorporation between deep
learning and mathematical computing, I believe that the model will perform more steadily and robustly.

After optimizing inference on static images, using multiple sources of input has another advantage that, in video or live demo, when dynamic pose estimation is required, hand-centralized images (output of S1) helps the analysis of pose on each image, while the sequence of joints locations(output of S2) provides great assistance on detecting the change of pose. The latter part, we propose, may include pure mathematical calculations to compute the angles between fingers and the distance of movement. With this approach, in the waving pose, we can detect the ‘5’ pose consistently, and observing the waving movements by analyzing coordinates changes.

Incorporating deep learning and mathematical computing, we aim at constructing a steadier and more robust model. I construct the ResNet50 according to the instructions in coursera[15], and import a vgg model from keras.application API. I utilize and modify the MobileNet v2 implemented in GitHub open source project[16].

### 3.5 Evaluation and metrics

Since this project focuses on a classification problem, we use a standard Softmax Cross-Entropy method to calculate the loss. The formula is shown below.

\[
\begin{align*}
    p_{o,c} &= \frac{e^{x_c}}{\sum_{d=1}^{C} e^{x_d}} \quad \text{for } c = 1 \cdots C \\
    CE &= -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})
\end{align*}
\]

We add the Dense layer at the end of each model with Softmax as activation function and calculate the categorical cross entropy loss on validation datasets. Before we construct the final model and move to work with specific poses, we will evaluate the models based on the datasets mentioned in Literature Review related to American Sign Language and digits, because they contain a wide range of complicated poses, which will powerfully demonstrate the classification capability of models.
Aiming at video analysis, the efficiency of models is as important as accuracy. Therefore, we will also evaluate the inference speed of models by Frame per Second (FPS).

In this computer vision problem, apart from the quantitative approach, we will also use visual judgement on the model’s test results, since this will filter out serious mistakes that yield small loss.

### 3.6 Summary

In this chapter, the methodologies and approaches were introduced in details. I started with the project workflow, hardware, and software setting, and then data preparation followed the model construction Chapter. Last but not least, the metrics were explained. In the next chapter, experiments and results will be presented, along with the challenges encountered when carrying out the methodologies, as well as the corresponding solutions.
4 Experiments and Results

In this chapter, a series of experiments will be covered. It evaluates the advantages and weaknesses of the series of experiments and the progress made in each phase. Existing datasets are collected and evaluated on models implemented in Keras, TensorFlow. After filtering out flawed datasets, models with competent validation accuracy are chosen. Upon selecting the models, we create a Joint Pose Dataset with four categories, which includes information of hand masks, joints, and meaning of gestures. The accuracy and the effectiveness are both taken into consideration. With settling down the final methodologies, the Joint Pose Dataset is extended to seven poses. The quality dataset and the optimized methodology altogether contributes to the success and contribution of this project.

4.1 Classification on Existing Cropped Datasets

4.1.1 Data Preprocessing and Augmentation

As introduced in Chapter 2 Literature Review, (4) Sign Language Digits Dataset, (5) Hand Gesture Dataset, (6) ASL Alphabet are chosen. (5) has the highest quality but the least number of data for each class. Background replacing and Gaussian noise are applied to (5) as data augmentation, shown in figure 7, so that the amount of data in each class from each dataset will be balanced. We resized and padded all datasets into 200*200 RGB images, preprocessed with other data augmentation approaches mentioned in the Methodology Chapter, and split them into train and validation partition with ratio 4:1.

In this Chapter, all the validation dataset refer to a one-fifth partition of the original dataset, while test dataset usually comes from other datasets, or my Real-World Test Dataset.

4.1.2 Model Construction and Evaluation

In Stage 3, we implemented a ResNet50 with identity blocks and convolutional blocks. In the first attempt to train combined dataset, the training and validation accuracy didn’t change
Figure 7: (a) sample data from Hand Gesture Dataset (b) augmented data with background replacement and random direction (c) augmented data with Gaussian noise

through epochs. After testing the correlation between input size and learning rate, we found that with a higher input resolution, the learning rate must be adjusted to a lower level. We chose Adam optimizer with learning rate 0.0001, which made the loss reduce gradually and finally converge to around 0.5. With a batch of size 8, the training and validation accuracy reached over 0.9 in 26 epochs, as shown in figure 8.

![Figure 8](image)

Figure 8: Train and validation accuracy of ResNet50

We also adopted the VGG-16, which was developed from AlexNet but with smaller kernels and deeper network. It has similar performance on the validation dataset. Nevertheless, FPS of ResNet inference on my local computer without GPU is 4.2 frame per second while VGG is 2.7, and therefore ResNet50 is selected between the two cutting-edge models.

Concerned with the speed for real-time analysis, we also tried to construct a smaller net which might yield a higher FPS. We built a random forest with 5 identical networks, each consisting of less than 10 convolutional 2D layers with batch normalization, and also ended by a softmax
activation function. Each network was trained in parallel and a network with the highest likelihood is selected when predicting on valid data. This model converges slower than ResNet50 and VGG-16, taking on average 40 epochs to achieve a 0.7 validation accuracy. However, its inference FPS is around 5.5 on the same machine, better than ResNet50 in real-time analysis. As a summary, ResNet50 and VGG-16 have relatively high validation accuracy, but their efficiency might not be capable of real-time analysis; a random forest takes longer to train and has a lower validation accuracy, but it has the best efficiency.

4.1.3 Obstacles and Solutions

The high validation accuracies on ResNet50 and VGG-16 have proved the potential capability of them in handling the hand classification problem. With enough epochs and large batch size, these models will yield reliable result inside the dataset. However, when I use (8)Real-World Test Dataset, the result is very poor, with test accuracy roughly 55%. Since the models are reliable, the problem may lie in the quality of dataset. After a second review on the selected dataset, several major issues are found:

(i) Both dataset (4) and (6) are collected from some fixed backgrounds. After training, models can easily distinguish between these hands and these monotonous backgrounds, but the performance cannot be accustomed to random background.

(ii) Dataset (6) is collected through a video. Although this is an efficient way to collect large amounts of data, the data collected in a sequence appear to be identical. As shown in figure 9, this dataset claims to have three thousand data for each pose, but the time interval between each two continuous frames taken is so small, that the data are visually duplicate. If these identical data are split to both training and validation set, models easily get high validation accuracy. Dataset (5) avoids the aforementioned drawbacks because segmented data can be efficiently augmented, and this methodology is adopted to the next experiment. We decide generate our own dataset on segmented hands, and replace (4) and (6).

4.2 Segmentation Dataset from Background Masking
Figure 9: These images all almost identical. Splitting these data to train and validation set may result in exaggerated high validation accuracy

4.2.1 Dataset Generation

Realizing the insufficiency in the datasets evaluated from the above Chapter, I research on how to clean the background of images.

The first idea is to use upper and lower thresholds for (Red, Green, Blue) values. Only when a pixel locates in the range bounded by the thresholds, will this pixel be kept. Under this approach, each pixel has a corresponding boolean value, and for a data with (200, 200, 3) shape, it will have a mask with shape (200, 200). A series of threshold tuples were manually entered, but the resulting mask was not promising as in figure 10. It is difficult to filter out the background without corroding the hand. HSV is an alternative color space of RGB. HSV aligns closely with how color attributes are perceived by human vision. The idea of using HSV comes from this github project[16]. Additionally, a console is implemented to adjust the thresholds of HSV instead of fixed tuples in each execution. Figure 10 also shows an example of the interface, where each of the thresholds can be adjusted in the process. A boundary box is painted on the interface, and after a fixed background is chosen, it is accessible to complete a mask, as shown

Figure 10: This is an attempt to use RGB threshold. It is difficult to filter background without affecting the hand
in figure 11 and 12. They are both satisfactory cases.

Figure 11: Generating segmented hands with a precise mask

Figure 12: This mask is not so successful since some background noises are not excluded. However, as the hand is complete, this data is acceptable

Several volunteers are asked to strictly follow the pose definition, and make some variations at the same time. They rotate their hands slightly, and move back and forth in front of the camera, to promote the variation in hand locations, angles, and size, in the image.

On the first phase of generating this dataset, ‘5’ ‘F’ ‘S’ ‘A’ are collected from five volunteers, altogether 150 data per pose. This quality dataset is the second contribution of this project, with the following advantages:

i) The hand poses in data strictly follow the definition of American Sign Language

ii) Variety of gesture attributes, including location, facing direction, are captured inside the dataset

iii) Background augmentation is straightforward with precisely segmented data

4.2.2 Classification with Grey Level Image
The training process of this experiment is similar to Chapter 4.1.2. As the data background are set to zero, intuitively, this could be handled by a faster but less accurate model. I attempt both ResNet50 and MobileNet v2. Compared with ResNet50, MobileNet nearly doubles the speed and achieves FPS 7.

Remarkably, during the first project presentation, our FYP supervisor, Dr. Schnieders gave a suggestion on using grey level images instead of RGB ones. With number of data layers reduced to one, the FPS improves significantly, reaching an FPS of 8.5, and reducing the layers does not affect the accuracy of the models. The accuracy on validation set is 95% and on test set it is 73%. The fact has proved that Dr. Schnieders’s suggestion is considerably helpful.

4.2.3 Evaluation and Analysis

Accuracy and efficiency are the two main factors this project focuses on, and they are both improved in this experiment: self-generated segmentation hand dataset promote the comprehensiveness of dataset, boosting the accuracy; grey level data provides less information than their RGB form, trading off precision for large speed improvement.

Nevertheless, on real-life test cases, the performance verifies to be unpromising. Although big progress has been made compared with the first experiment, the gap between our validation performance and real-world application cannot be filled.

To find the reason for this gap, another literature review is conducted on papers on arXiv and open source projects on GitHub. Few papers and projects research on hand pose classification, and fewer uses only RGB image. Some relevant papers use the existing datasets we have analyzed in Chapter 4.1.1 and therefore not reliable. The others claim to achieve real-time demo, with Support Vector Machines and Inception Nets, but their accuracy are low in real-life test cases. In other words, in existing open source works, pose estimation is hardly accomplished with a classification model.

Intuitively, color is one of the most important features of an object from human vision. Different hand poses have the same color, and the textures of hands are also the same regardless of the gestures. Therefore, the complexity of a traditional classification model might not be high enough to handle different categories with large similarities.
This argument could be strengthened by the case of the most famous classification topics, ImageNet. GoogLeNet[4], also known as Inception v1, was the champion of ILSVRC 2014 (ImageNet Large Scale Visual Recognition Competition). It has an error rate of only 6% among thousands of classes with on average 700 data for each category. I inspect into a few mislabeled images of this model, and find out that color and texture are both critical factors. The trucks and cars with similar car shell cannot be labeled correctly, different species of birds with same feather color and pose are hardly distinguished. Therefore, when objects share too much common features among different categories, even the state-of-the-art models cannot perform precisely. In terms of hand, a polymorphic object, it will be as well hard to use purely classification model to solve the problem. In the next experiment, more informative inputs are adopted so that feature can be conveniently extracted from my classification model.

4.3 Extending Application on Segmented Hands

The previous two experiments are on cropped images because when we set the pipeline of the original project, Stage 1 outputs cropped image. Since now cropped-hand is too complex for our project, My group mate working on Stage 1 attempts to do hand segmentation instead of hand detection, which will reduce the complexity of the joint recognition task on Stage 2 as well as pose estimation task for my Stage 3 In this Chapter, segmented hand is the input of my project.

4.3.1 Training on Segmented Hand

In this experiment, ResNet and MobileNet are still the core model, and they will not be replaced unless the loss fails to drop and converge.

We use our own segmented hand dataset, which contains four labels, ‘A’, ‘S’, ‘F’, ‘S’. The image processing and training procedures are similar to the previous two experiments. The segmented images have pure black background, and they are converted into grey level image to improve efficiency. Stochastic Gradient Descent with initial learning rate $1e^{-3}$, decay rate $1e^{-5}$ is chosen after some warm-up tuning.
In this attempt, MobileNet reaches validation accuracy of 0.80 in 70 epochs. In Chapter 4.2.1, an interface tool is adopted to generate segmented data. In real-life test case, I use this interface to predict real-time analysis and yields positive results. With the continuous input from a live camera, the model can eventually predict most of gestures correctly. The FPS of this MobileNet model is around 6 on the personal laptop, without GPU.

This is the first time high accuracy is achieved in real-world test cases, and as a third contribution in this project, we proved that segmentation helps to form a solid prediction model in pose estimation, and outperforms cropped-hand image from hand detection intensively.

### 4.3.2 Training on Hand Masks

As the entire group project has three stages, where the previous two stages have relatively low FPS, the efficiency of this individual project shall be enhanced to reduce the whole burden and increase the overall FPS.

The intuition comes from the difference between cropped hand and segmented hand: the environment noise is mitigated to zero, indicating that the outline of a hand can be effortlessly obtained from feature extraction. Each pose itself is a hand, with similar colors, similar structure, similar ratios of finger size and palm size, but the most distinct feature the the outline of the poses. It might be sufficient to just provide the shape and shadow of this gesture as training data.

To verify the hypothesis, modifications are made to the Segment Hand Interface introduced in Chapter 4.2.1. Instead of the segmented hands, only masks of each pose are saved. As aforementioned, a mask consists of boolean values, which vectorizes as a square matrix with 0 and 1. To further scale down the computing intensity, masks are resized to 64 * 64 squares. At this stage, scale of input has reduced from (200, 200, 3) with each value ranging from 0 to 255, to (64, 64, 1) binary matrix.

I applied the random forest with 5 identical simple networks introduced in Chapter 4.1.2. Due to its small input size and lower complexity of the networks, the efficiency is improved triply, to 20 FPS, and the test accuracy also improved to 0.9 at the same time.

Here is my Noise Reduction Hypothesis:
Deep learning models extract from low level features like color, size, shape and textures, to high level features that human vision cannot comprehend, as shown in figure 13. The higher percentage of useful features the model can extract from, the more accuracy and efficiency the model is. From the following table 4, we can see the accuracy increases with the reduction of input data scale, more specifically the descent in noise. From segmented hand to hand mask, irrelevant skin color and palm-print are excluded, while effective features like outline of hand and finger length are kept. Therefore, the decreasing input size not only accelerates the training and the inference procedures, but also raise the accuracy.

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Model</th>
<th>Input Size</th>
<th>Test Accuracy</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropped hand</td>
<td>ResNet</td>
<td>(200,200,3)</td>
<td>55%</td>
<td>4.2</td>
</tr>
<tr>
<td>Segmented hand</td>
<td>MobileNet v2</td>
<td>(200,200,1)</td>
<td>73%</td>
<td>7</td>
</tr>
<tr>
<td>Hand Mask</td>
<td>RandomForest</td>
<td>(64,64,1)</td>
<td>around 90%</td>
<td>8.5</td>
</tr>
</tbody>
</table>

Figure 13: This is a screen shot from Stanford CS231N. From left to right, higher and higher dimensional features are shown. The features on the left can be understood by human vision, while the columns on the right side are not.

Table 4: Although the input size decreases at each phase, the accuracy actually rises. This is because higher percent of useless information in pixels are discarded

4.3.3 Potential of Mask Application

Realizing the feasibility of hand mask, we intended to connect Stage 3 right after segmentation output from Stage 1, as introduced in Chapter 4.3. Nevertheless, although using segmentation hand input can triple the efficiency, from 0.10s per image to 0.03s per image on CPU, the actual
improvement is only 0.1s per second; in original plan, hand detection takes 0.13s per image on CPU, but the new proposal with hand segmentation dramatically increase the processing time to 5s on GPU. On personal laptop without GPU, segmentation is not practical, as it may take more than a minute to process a single image; the 0.07s improvement on Stage 3 cannot fill this huge gap.

Therefore, considering Milestone 3, this newly proposed pipeline is not feasible, but it does enhance the inference quality on image (Milestone 1) and video (Milestone 2).

Additionally, this experiment helps to develop a side project on fixed-background ASL translator. We obtain masks for all classes in ASL, and train the random forest model with the extended mask dataset. As shown in figure 12 in Chapter 4.2.1, one can adjust the threshold values in the toolbar to generate a precise mask within the fixed boundary box. The mask will be resized to (64, 64) and passed to the model to yield prediction. The time lapse for one inference is about 0.05s, so we have actually developed a live ASL translator.

4.4 Classification Based on Joint Recognition

4.4.1 Intuition

In Stage 2 (my group mate’s project), the determination of joints location determines the structure of a hand quantitatively and can be used to handle complicated human-computer interaction tasks. Since Stage 2 is indispensable, I decide to make advantage of these key points and construct model based on Stage 2. Intuitively, given only the twenty one pairs of joint coordinates, human can imagine the structure of the hand aligned on a 2D plain, and recognize this hand pose. Therefore, the joint coordinates themselves might be sufficient for pose estimation. According to my Noise Reduction Hypothesis: (i) 21 joints infer the 2D structure of hand, the length of each finger and their positions, which are all useful information (2) they discard unusable features like color and palm-print. In conclusion, it’s highly possible that this methodology could raise the accuracy and efficiency to a new level.

4.4.2 Joint Pose Database
To carry on the joint experiment, we create the Joint Pose Database based on segmented hand database generated in Chapter 4.2.1. We further extend the label from ['A', 'S', 'F', '5'] to ['A', 'S', 'F', '5', 'I', 'L', 'Y']. The latter three are collected because the model in Stage 2 (my group mate’s project) can label this poses with acceptable accuracy without training. Stage 2 and this individual project cooperate closely, in the following pipeline:

(i) Stage 2 predicts joints coordinates from segmented hand database
(ii) Manually corrects the mislabeled joints and create more joint labels
(iii) Stage 2 trains with the refined labels
(iv) Stage 2 predict more data than last time
(v) If still lack of data and label, goto ii)-iv)

After three iterations, Joint Pose Dataset is eventually finished. I implement a tool, as in figure 14 to label new data and adjust the joints predicted by Stage 2, which accelerates the iterations dramatically.

Although this procedure takes a week to finish, it is worthwhile: This Joint Pose Dataset,

![Figure 14: This is the labeling tool. We use mouse to drag joints to the correct position and they will be saved in a json file](image)

contributes as the first hand dataset, that has 7 pose classes, each containing 160 precisely segmented hand images, and accurate joint coordinates. Combining hand’s spatial information and gesture semanteme in this dataset is a big progress in the Human-Computer Interaction area.
4.4.3 Data Modeling

Four approaches on data formatting are taken into consideration: (i) Gaussian Map (ii) geometrical features (iii) a single vector (iv) boolean matrix. Not all the data models are implemented.

4.4.3.1 Gaussian Map

This idea originates from the labeling format of Convolutional Pose Machine\[18\]. 21 Gaussian maps of size (200, 200) are generated, where each map shows a Gaussian distribution, indicating the probability of each location to be a joint. A 3D figure, figure 15 can illustrate the idea: in the left image, the higher in the Gaussian Mountain, the higher the possibility that this point is joint location, and the volume of the mountain adds up to one. The figure on the right is a 2D version, and brighter pixel has higher possibility to be a joint. This is the most informative data format, but 21*(200, 200) has too much redundancy, and conflicts with the goal to build a light-weighted model. On personal laptop, the inference of this 21 layer Gaussian map takes more than 3 seconds to finish. Therefore, this approach ceases before training.

4.4.3.2 Geometrical Feature Analysis

As introduced in Chapter 4.4.2, the Joint Pose Dataset contains seven categories, ['A', 'S', 'F', '5', 'I', 'L', 'Y']. Researching on the features of each pose, I find that an acute angle between finger bones indicates that this finger is curly, while all blunt angles means this finger is almost straight; additionally, the distance between each pair of fingertips can further determine the pose. Theoretically, all the standardized poses from one category should have the same attributes. However, due to the noise in the environment, at real-world test cases, Stage 2 cannot
predict the joints as precisely as how data are labeled, and a standard rule may not be applicable on every data. As a consequence, this method only achieves 49% accuracy on a set of predicted joints (without manual correction).

Realizing the risk of real-world cases, we generate a larger dataset consisting of three parts: Joint Pose Dataset, predictions from Stage 2 on segmented hands without manual correction, predictions from Stage 2 on hands with various backgrounds (a result of data augmentation from segmented hands). The fraction of each part is roughly 1 : 1 : 1, and the new dataset has 4000 images. In (iii) and (iv), the “real-world test cases” are selected from the third part of dataset, predictions with various backgrounds.

4.4.3.3 Joint Vector

Considering input data as a (21, 2) shape vector can maximize the efficiency. I construct a random forest with five MobileNet v2, each trained in a different source of dataset as afore-mentioned. Since each model are trained in different environment, they are expected to handle different circumstances successfully. At each inference, each model yields a confidence score for each class, and the top score of all classes is selected. This approach reaches 95% validation accuracy in 100 epochs, and has 20 FPS. However, each data only has 42 value points, while the whole network is rather complicated. With only four thousand data, the overfitting problem cannot be avoided even if I add three dropout layers in each network. The validation dataset was split from original dataset, where the joint style are similar; on real-world test cases the joints structure are irregular and the model yields unstable results. In a real-world test case, the accuracy is 0.14, close to random guess result.

4.4.3.4 Binary Matrix

Just like hand mask, a (64, 64) joint mask is generated. the joints as well as the connecting bones are painted on this binary canvas. It inherits the informative presentation format, while reducing the range from 0 255 to binary. The joint sets are normalized by zooming and shifting, so that each pose is at the center and takes up entire 64*64 space. After that, the coordinates and their connections are painted as shown in figure 16. The intuition is that, although it is impractical to obtain segmented hand as input data, we can use the joints to paint a skeleton of the hand,
which has similar characteristics with hand outline mask, figure 17.

I reuse the light-weighted networks trained with hand masks in Chapter 4.3.2 and apply transfer learning. Given that the network already achieves a real-time-analysis standard accuracy with hand mask, I further train the net with these mask-like binary maps. After 50 epochs, the validation accuracy reaches 98%. The loss converges as quick as the training process Chapter 4.3.2., as shown in figure 18. Aware of the potential risk of overfitting, a video is taken to analyze the real-world performance. This light-weighted model has 30 FPS, and the whole project integrating three stages can as well reach 10 FPS, which achieve the standard of live demo. In terms of accuracy, in the 18-sec video with 538 frames, the inference accuracy is over 70%, regardless of the trembling joints.

This is the fourth contribution in this individual project: It proves the feasibility of transfer learning from segmentation problem to keypoint classification problem, and paves way for pose estimation with pure RGB image.

The binary matrix methodology is adopted as the final solution to this individual project, because of over 70% real-time analysis accuracy and 30 FPS. In the following Chapter, how this project (Stage 3) is integrated into the whole project will be discussed.
4.5 Joint Threshold and Kalman Filter

The difficulty of pose estimation mainly comes from the unstable prediction of Stage 2. To mitigate the risk, a threshold is added after output joint coordinates. Each joint is associated with a confidence score, and if the scores on fingertips are too low, it indicates that the pose and the direction of this finger cannot be determined. Among the seven poses we are dealing with, the location of fingertips of thumb, index and little are quite important. Under this circumstance, we agree on a sets of threshold that, once a vital fingertip is ambiguous, even if the overall score is promising, we will terminate the inference on this image before Stage 3.

Moreover, Kalman Filter is implemented after Stage 2. Kalman Filter is a statistical algorithm which helps to predict the next time state movement based on transition matrix, measurement matrix, and noise factors. It records past events and attempts to predict smooth movement of current stage. I adopt the widely accepted values, figure 19 with noise factor 0.03. Plotting the video output from kalman filter, the level of joints trembling is greatly reduced, and the continuity of joint movements is also improved.

With these two approaches, joints output from Stage 2 are carefully treated and then used to infer the class of pose.

4.6 Experiment Summary

The most difficult part in these experiments is the lack of existing quality resources. Only a few datasets can meet our standard, and we have to generate a new Joint Pose dataset. Few works solve this classification problem on RGB image level with random backgrounds, so I need to
do a large range of experiments to find an optimized approach. Cropped hand, segmented hand, binary mask, joints have all been experimented in the procedure. Their feasibility, accuracy and efficiency are all considered, and Chapter 4.4.3.4 Binary Matrix approach strikes a good balance among these factors, outperform the other methods. Throughout the experiments, my knowledge on traditional computer vision, deep learning, statistics, and programming skills are challenged and strengthened. Being a pioneer work on high accuracy in RGB hand classification, I hope this project can help others to think further based on it, and hopefully promote the development of Human Computer Interaction with bare hands.
5 Future Works

5.1 Future plans

The accuracy and the efficiency standard we set at at the beginning of the project has been reached, and the three Milestones have been accomplished. The future direction may lie in the higher variety of poses, i.e., to accept joint recognition and pose estimation on all the 26 letters in American Sign Language. Additionally, the joint trembling and mislabeling remain a major issue. If more reliable solutions can be adopted later on, stable joints will have higher probability to yield correct prediction result.

Combining joints information and traditional computer vision is another idea that is not within the initial scope of this project. But since binary joint masks can play an equivalent role as segmentation masks in this problem, the reverse direction may also be practical: use joints and local features obtained by traditional CV methods to yield hand segmentation. This might be more efficient than directly applying deep learning.

If the above problems can be solved, this work may be extended to many applications. For example, a real-time ASL translator that tracks hand in the webcam and offers the meaning of sign languages. Additionally, hand gestures translation may also provide user interface for people to control computer remotely with hands, and it improves the user experience.
### 5.2 Schedule

Table 5: Schedule

<table>
<thead>
<tr>
<th>Time period</th>
<th>Events</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.30</td>
<td>Delivery of Phase 1</td>
<td>Completed</td>
</tr>
<tr>
<td>10.15</td>
<td>Further research on papers</td>
<td>Completed</td>
</tr>
<tr>
<td>10.31</td>
<td>Data preparation</td>
<td>Completed</td>
</tr>
<tr>
<td>12.15</td>
<td>Model implementation</td>
<td>Completed</td>
</tr>
<tr>
<td>1.20</td>
<td>Model training and evaluation</td>
<td>Completed</td>
</tr>
<tr>
<td>1.20</td>
<td>Delivery of Phase 2</td>
<td>Completed</td>
</tr>
<tr>
<td>2.13 - 2.22</td>
<td>First Presentation</td>
<td>Completed</td>
</tr>
<tr>
<td>2.28</td>
<td>Performance optimization</td>
<td>Completed</td>
</tr>
<tr>
<td>3.15</td>
<td>Live demo</td>
<td>Completed</td>
</tr>
<tr>
<td>4.14</td>
<td>Final Delivery</td>
<td>Completed</td>
</tr>
</tbody>
</table>
6 Conclusion

All the relevant and important information of this individual project have been presented in this interim report. We have discussed the background and motivations of this topic, literature review and methodologies, experiments and future plans.

This is a sub-project of the whole group project on Deep Learning Hand Poses, and it serves as a sign language translator. Although many challenges and obstacles are encountered in the halfway, I manage to finish the project, reaching the accuracy and efficiency goal, and completing the three Milestones, from images, video to live demo. In this project, several contributions are made.

Firstly, a comprehensive dataset review is conducted. The research points out the insufficiency in those current datasets, and propose standard to form better ones. Secondly, a multi-functional quality dataset, Joint Pose Dataset, is created. It not only serves for our own usage, but could also benefit other researchers. Thirdly, this project finds a solution to pose estimation by classification on hand segmentation. As a result, at a fixed location, a live-demo ASL translator can be easily implemented. Last but not least, it applies a transfer learning approach and eventually manages to achieve the destination of pose estimation.

Therefore, overall speaking, this project is a success. It is a personal success, and might also push the hand research forward one step. Due to the time and resource limitations, I have faced lots of challenges. Fortunately, now this project is eventually finalized and completed. In terms of further studies, more poses can be trained with the model to consummate its functionalities, after which digital applications can be built based on this model, to invite a new mode of Human Computer Interaction.
References


