Department of Computer Science
University of Hong Kong
Final Year Project Plan

Deep Learning Hand Poses

Supervisor: Dr. Dirk Schnieders

Group members: Chen Zihao 3035232640
Da Yujia 3035234038
Zhang Yuqian 3035233565

FYP ID: fyp18034

1. Problem Statement

In the past few years, smart devices such as smart phones, smart watches and virtual reality headsets have been widely used in many occasions. It is, however, sometimes not very convenient to interact with these devices with buttons or touch screens. Users are able to see the effects, but not to touch them with their own hands. To illustrate, the screen of a smart watch may be too small for users to easily press the exact button, and they would probably prefer clearing messages with a wave of the hand. To achieve this goal, a model that can accurately recognize human hand poses is in great demand, and more attention needs to be attached to specific hand gestures for practical applications.

2. Objective

This project aims to achieve highly accurate hand pose recognition with deep learning, particularly on several specific hand gestures. The project will be divided into three stages, further information can be found in Table 1.

In the first stage, our goal is to build a deep learning model that can detect or segment human hands from the input picture. Accuracy is of great importance to this stage since it is the base of all following stages.

In the second stage, we intend to build a model that takes in the output of stage 1 and mark down the joints and fingertips. We expect that our model can achieve better performance than the state-of-the-art in terms of time and accuracy.

Our final objective is to develop an application that can timely recognize and respond to certain specific hand gestures and carry out the requested tasks. For example, one of the hand gestures that we target at is the “thumbs-up” gesture, which can be recognized by our model and contribute one like to our final year project page. The images of the hands should be able to be captured by an ordinary camera that is available on modern smart devices, and the analyzed results can be used to enhance user experience in interacting with these smart devices.

Classified by difficulty of successful implementation, our general objectives are: a) recognize still poses in a static image; b) recognize dynamic poses in a video stream; c) recognize dynamic poses in real time. In our expectation, a) will be accomplished with the preliminary model, while b) and c) will involve additional implementation on dynamic algorithms. A video or live demo can be regarded as a time sequence of images, and therefore we consider b), c) as a further step from a).

3. Background

There are many existing models that are able to achieve hand detection and recognition. In recent years, progress has been made in this area in terms of time performance and accuracy. There are two choices when building up a deep learning model. One is to build an end-to-end model that takes in a picture and output the same picture with marked joints and fingertips on it all in one. The other is to separate this process into two stages, the first detecting the position of the hand and the second marking down the 21 joints and fingertips on the cropped hand-centered images. The benefits of multiple stages is that models can be built independently for different stages and be trained separately and simultaneously. Therefore, most existing works chose to separate the process into multiple stages. [16] introduces a method that takes advantage of depth information obtained from passive stereo to perform better hand detection. A hierarchical PointNet with 3 point set abstraction levels was implemented in [17] and outperformed the state-of-the-art methods through optimizing stage 2. Both stage 1, detecting hands, and stage 2, marking joints and fingertips, were discussed in [8], where 3D keypoint is gained by using a sub-network, a transformation matrix learnt through training.

During the development of hand pose recognition, people are faced with several issues that are particularly difficult to tackle with. These problems include hand occlusion, high similarity between fingers, the impact of illumination, large variations of hand gestures, etc. Works have been done in these related fields as well. Methods that can reduce the influence of self and object occlusion as well as cluttered scenes are discussed in [18]. [22] uses a network that has two branches, B1 recognizing all the joints, and B2 recognizing how the joints are appropriately connected, where B2 is based on Part Affinity Fields (PAFs), a sub-network to learn to associate body parts with individuals in the image. Combining B1 and B2, the goal to detect multi person pose at the same time can be reached at the same time. This piece of work can be transferred to the field of multi-hand recognition as well.

New ideas that aimed at enhancing performance or reducing the required input information were proposed by different researchers. Among the existing works, some used pure rgb values, others developed their models based on depth information as well. Both [19] and [21] came up with the idea of capturing the hand simultaneously from different viewpoints, which can enhance the performance in occlusion cases. An interesting model that takes “real” RGB images generated from synthetic images together with 3D locations of joints and fingertips as training data to achieve the goal of generating 3D outputs without depth information is proposed in [20].

Also, multiple datasets has been built up to facilitate the development of new models in the area of hand pose recognition. Popular hand pose datasets include NYU Hand Pose[24], ICVL, and EgoHand[25], all of which contains hand pose images that are real and with both RGB and depth information.

4. Scope

Considering the complex nature of this problem, the scope of our final year project will temporarily stay within the following boundary.

This project will only focus on input of pure 2D RGB values. Though putting depth information into consideration may significantly enhance the performance, our goal is to develop a model that can be used with daily accessible devices.

To reduce the complexity of the problem, this project will only consider hands that are not occluded by other objects or hands appearing at the edge of the image which are only partially visible. Besides, this project only considers normal human hands, while animal paws or deformed hands.

Moreover, to attach practical meanings to our project, while a fair amount of training will be carried out in all hand poses, only a few selected hand gestures will receive more attention in model training.

5. Prerequisites

Since this project will be mainly coded in Python 3.6, we consider the following platforms for this computer language to be relevant and helpful: a)PyCharm, a python platform supporting wide range of functionalities including debugging and formatting; b) Sublime Text 3 , which is an elegant and convenient text editor, easy to pick up with; c) Jupyter Notebook, an web application supporting block-by-block execution with corresponding results, provides great convenience in debugging and illustration.

With regard to hardware requirement, this project demands Graphics Process Units(GPU) to run deep learning network with. Google Cloud Platform, providing stable GPU services, will be our primary source to train the networks intensively, while another GPU offered by our supervisor, will act as long-term support for further tuning. Additionally, a digital camera is needed for video and real-time analysis.

Lastly, the essential library we will be working on is TensorFlow, an open source machine learning API with wide range of efficiently implemented library functions. OpenCV is an open source computer vision library that is supported by TensorFlow, which will help to process the images.

6. Methodology

6.1 Project Pipelines

This project constructs a Computer Vision model on TensorFlow, and aims at recognizing specific human hand poses from various visual sources, including images, videos and live demo.
The ultimate task, a top-down approach on hand pose recognition, will be divided into three stages, defined as followed:

Stage	task	input	output
1 hand detection/ segmentation	locate all human hands on images and crop the hands out	Full size images containing hands	Cropped hand-centered images
2 joint recognition	Mark the 21 joints on each hand image	Output of stage 1	2D coordinates of hand joints
3 pose estimation	Associate the input data with a predefined hand pose	Output of stage 2	Hand Pose Categorization

Table 1. The details responsibility of each stage, as well as input and output flows

6.2 Data Preparation

After our literature review, we select COCO 2017[23], NYU Dataset[24] and EgoHand[25] as our training and evaluation datasets. Firstly, as mentioned in the Background section, these are the most popular datasets used in hand pose estimation, which proved to be well-diversified and comprehensive dataset with high labeling quality, both from first-person and third-person perspective. Secondly, many state-of-the-art works set baselines according to these datasets, and therefore using the same datasets provides convenient platform for horizontal comparison and evaluation.

We preprocess all the datasets into format of stage 1, 2 and 3, providing input data for each of the three stages, so that each stage can be regarded as an independent task and can be trained simultaneously.

Data augmentation can increase the amount and enhance the comprehensiveness of input data. Currently, there already exist many approaches, such as global illumination, lossy image compression, specular reflection on skin[1], as well as random crop, random rotation[2]. We will select from these methods according to the performance, and thus strengthen our network.

6.3 Network Implementation and Training

In stage 1, Hand Detection/Segmentation, we will use a pre-trained object detector and train with our hand pose datasets. At present, many existed object detection models can perform well, including RetinaNet[3], R-FCN[4] and Mask-RCNN[5], according to the leaderboard of COCO 2017 Object Detection Task[6]. Apart from model itself, the innovative ideas to train hard examples will also be taken into consideration, for example, focal loss[3] and Online Hard Example Mining[7]. We will try a series of networks and ideas to figure out one that is the most suitable for our hand detection case.

In stage 2, Joint Recognition, a series of existing networks both on hand pose and body pose will be chosen to compare the performance. *Learning to Estimate 3D Hand Pose from Single RGB*[8] is a good instance in this stage, while *Convolutional Pose Machine*[9], which initially focuses on body pose, has proved to show considerable accuracy in our preliminary attempt in adaptation to hand pose. We also look into *COCO 2017 Keypoint Detection Task*[10], to vertically compare with the models in the Object Detection Task[6], which provides insights to efficiently associate the first two stages altogether.

In stage 3, static image inference can be categorized in object classification, where abundant impressive models can be found. ResNet[11] and VGG[12] are both commonly used backbone network that once achieved great performance in *ImageNet Competition*[13]. We shall evaluate the performance of apply classification algorithm to the joints from stage 2, or apply to the hands from stage 1, and even from a concatenation of both. With regard to video inference, fewer excellent works in 2D dynamic pose have been done and thus more innovation will be involved. Our present intuition is to refer to the existing depth method based on traditional Computer Vision method[14], and then compromise with deep learning approach. Due to uncertainty on the feasibility, we might look for other approaches later on.

6.4 Evaluation and Metrics

The evaluation method we will be using on the first stage will be mean Average Precision(mAP). It calculates the .a widely accepted approach for horizontal comparisons. On the second stage, we plan to apply Object Keypoint Similarity(OKS)[15], with the formula

$$OKS = \Sigma_i[exp(-d_i^2/2s^2)\delta(v_i > 0)]/\Sigma_i[\delta(v_i > 0)]$$

where d_i is the L2 Distance between ground truth and prediction, s is an object scaled factor, v indicates whether the keypoint is labelled and visible(not occluded) or not, and i denotes the index of joint. The third stage is a classification problem, and we will use softmax cross-entropy method to calculate the loss.

Apart from calculating the loss, we may also do visual judgement to justify the accuracy of model's inference, since loss in same quantity can result from predictions with different levels of distortion.

7. Feasibility Analysis

The crux of this project is to build and train the models for detecting hands, marking joints and fingertips, and recognizing particular hand gestures. There are, however, a large number of existing works and models that we can use for reference, and in the past few years hand pose datasets focusing on different aspects of hand pose recognition have been built for public use. These existing resources can provide guidance as well as training data for our model. The third stage, which requires recognition of hand gestures, may be more challenging than the first two stages, but with the satisfactory performance of both stage 1 and stage 2, It is feasible to develop an original algorithm to perform accurate hand gesture recognition and further applications.

8. Risks and Challenges

- **Loss amplified among stages**
  Since the three stages we are presenting are in a step-by-step sequence, the failure on earlier stage will dramatically affect the performance of later stage. For example, even though the stage 2 model can mark the joints on a hand with high accuracy, if a stage 1 model cannot find hand, it will only lead to meaningless results, as the input of stage 2 are wrong inference of hand
  Mitigation: Mainly depend on existing state-of-the-art on the first and second stage, to ensure the stability and accuracy of the model.
- **Time consumption constraint**
  Each stage will add up to total time of inference on a single image, and thus, the process could overtime given too many hands in one image. If inference on some images needs 0.5s, then it's impossible to realize the live demo, which requires Frame Per Second(FPS) of at least 10
  Mitigation: (1)Try further optimization on the network efficiency.
  (2)Implement a parallel and concurrent logic on predicting joints of each hand in an image

9. Project Schedule and Milestones

Time Frame	Task
9.30	**Delivery of Phase 1**: · Detailed project plan · Project website
10.15	Research into relevant work · Get familiar with the chosen Framework, and related packages · Further research on relevant paper, and choose the models to implement
10.31	Data preparing · Choose suitable datasets · Determine ways for data augmentation
11.30	Model development · Preprocess data and label for each of the three model stages · Implement the deep learning models
12.31	Preliminary result · Model training and hyperparameter tuning · Compare with baseline and evaluate the results

1.7-1.11	**First Presentation**
1.15	Dynamic application · Apply model on analyzing pose from video · Apply model to analyze real time scenario
1.20	**Delivery of Phase 2** · Preliminary implementation · Detailed interim report
3.15	Optimization and Innovation · Attempt innovative ideas to enhance the model · Achieve appropriate balance from the accuracy-speed tradeoff
3.31	Finalization of the Project
4.14	**Delivery of Phase 3** · Finalized tested implementation · Final report
4.15-4.19	**Final Presentation**
4.29	**Project Exhibition**
5.29	**Project Competition**

*the **blackened** items are released in the timetable from COMP4801 official website

10. Project Management

The workload will be equally divided on to the three group mates. With regard to deliveries, Yujia is in charge of website maintenance, while Yuqian and Zihao are responsible for most of the writing tasks. All the three members are doing the literature review together, to search for existing state-of-the-art deep learning models. Since our project is mainly comprised of three stages, namely, detecting hands, marking down 21 joints and fingertips, and recognizing hand gestures, each group member will be responsible for one of the three stages. More specifically, Yujia will focus on stage 1, ensuring that the deep learning model of our project will be robust enough to pass meaningful cropped hand-centered images to the following stages. Meanwhile, Yuqian will be in charge of stage 2, taking the output of stage 1 as input to train the model to mark the 21 joints and fingertips accurately and speedily. And Zihao will be responsible for stage 3, where more attention will be attached to the recognition of a few particular hand gestures. We expect that the last stage will be more challenging, therefore Yujia and Yuqian will also make significant contributions to the last stage in literature review and model design. Eventually, all three members will contribute to the development of the final application that can achieve real time respond to the hand gestures in front of the camera.

11. Reference

[1] C.Zimmermann and T. Thomas Brox, Learning to Estimate 3D Hand Pose from Single RGB Images, ICCV 2017

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu and A.C. Berg, SSD: Single Shot MultiBox Detector, in ECCV 2016

[3] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, Focal Loss for Dense Object Detection, in ICCV 2017

[4] J. Dai, Y. Li, K. He and J. Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks, in CVPR 2016

[5] K. He, G. Gkioxari, P. Dollar and R. Girshick, Mask R-CNN, in ICCV 2017

[6] COCO 2017 Object Detection Task, http://cocodataset.org/#detection-leaderboard

[7] A. Shrivastava, A. Gupta and R. Girshick, Training Region-based Object Detectors with Online Hard Example Mining, in CVPR 2016

[8] C. Zimmermann and T. Brox, Learning to Estimate 3D Hand Pose from Single RGB, in ICCV 2017

[9] S. WEi, V. Ramakrishna, T. Kanade and Y. Sheikh, Convolutional Pose Machine,in CVPR 2016

[10] COCO 2017 Keypoint Detection Task, http://cocodataset.org/#keypoints-leaderboard

[11] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, in CVPR 2015

[12] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv technical report, in 2014

[13] ImageNet Competition, http://www.image-net.org/challenges/LSVRC/

[14] A. Kurakin, Z. Zhang and Z. Liu, A real time system for dynamic hand gesture recognition with a depth sensor, in EUSIPCO 2012

[15] Object Keypoint Similarity, http://cocodataset.org/#keypoints-eval

[16] J. Zhang, J, Jiao, M. Chen, L. Qu, X. Xu, A and Q. Yang, Hand Pose Tracking Benchmark from Stereo Matching, in ICIP 2017

[17] L. Ge and Y. Cai, Hand PointNet: 3D Hand Pose Estimation using Point Sets, in CVPR 2018

[18] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, C. Theobalt, Real-time Hand Tracking under Occlusion from an Egocentric RGB-D Sensor, in ICCV 2017

[19] G. Poier, D. Schinagl and H. Bischof, Learning Pose Specific Representations by Predicting Different Views, in CVPR 2018

[20] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D Casas and C. Theobalt, GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB, in CVPR 2018

[21] T. Simon H. Joo, L. Matthews and Y. Sheikh, Hand Keypoint Detection in Single Images using Multiview Bootstrapping, in CVPR 2017

[22] Z. Cao, T. Simon, S. Wei and Y. Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, in CVPR 2017

[23] COCO 2017, http://cocodataset.org/#overview

[24] NYU Hand Pose Dataset, https://cims.nyu.edu/~tompson/NYU_Hand_Pose_Dataset.htm

[25] EgoHand Dataset,http://vision.soic.indiana.edu/projects/egohands/

[26] ICVL Big Hand Dataset, https://labicvl.github.io/Datasets_Code.html