Egocentric RGB Hand Detection

- **組別** 小魯與四位大神
- **組員**
  - 楊宗翰, r05922062
  - 林瑋柔, r05922128
  - 祝子軒, r05922131
  - 蔡佳文, r05922132
  - 黃博政, r05922136

Requirements

- Your report should be written in **one-column or two-column conference paper style**.

Introduction

Hand detection task includes two subtasks which are bounded box positioning and hand classification(left hand or right hand).
HTC provide two datasets which are DeepQ-Synth-Hand Dataset including 10,0000 labeled images and DeepQ-VivePaper Dataset including 464 labeled images and 6197 unlabeled images.
We using flipping and color analysis for data preprocessing. We apply four different methods for the task. 1.SIFT+YCbCr skin detection for capturing hands. 2.Cycle-GAN for domain adaptation and prediction. 3.SSD and 4.RetinaNet for hands detection.

Detect

SIFT + Skin Color Detector

- SIFT Feature Collection
  Collect hand segments from training data, and then extract feature using SIFT feature detector. We collect left/right hand description points separately, which could be used for later discrimination.
- Skin Detection
  We project images from RGB to YCbCr color space, then apply well-crafted color threshold to find posssible skin segments.
- Joint Inference
  Consider skin segments as possible hand positions, and then apply a KNN-based

matcher to their surrounding SIFT features with collected left/right hand description points as to determine if the hand belong to left or right.

Single Shot Multibox Detector (SSD)

- Tensorflow Object Detection API
- Using the MobileNets-SSD model to train a hand detector on synthesis data, and fine-tune on real data. This SSD model is an accuracy and real-time detector, achieve 0.788 mAP@IOU0.5 on deepq testing data.

RetinaNet

RetinaNet is a neural network purposed by Tsung-Yi Lin et al. This network is a powerful single stage detector, which inference bounding box and class directly from the input image. It aims to do dense object detection with scale-invariant. By means of focal loss, we are able to train on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training.

Data Augmentation

Apply data argumentation on labeled real data is an important issue, since labeled real data is rare compared to synthetic data.

Flipping

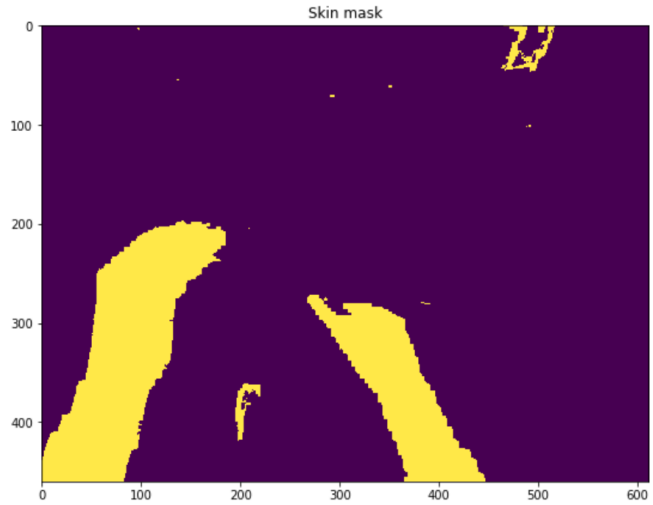
We noticed that data augmentation can simply achived by left-right flipping pixels and label of the real image.

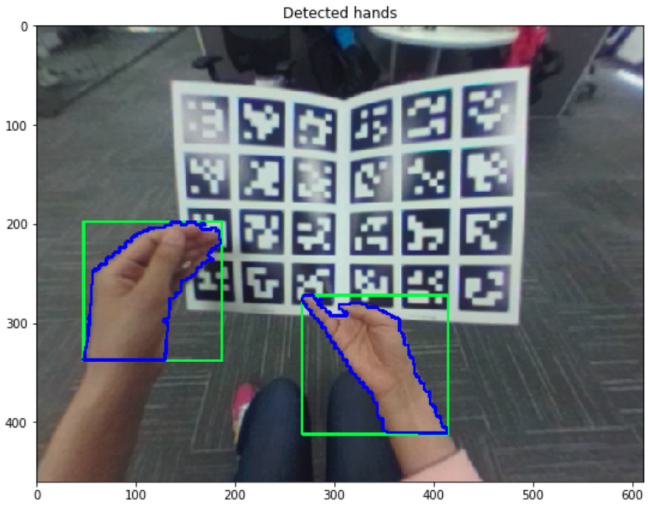
Domain Adaptation

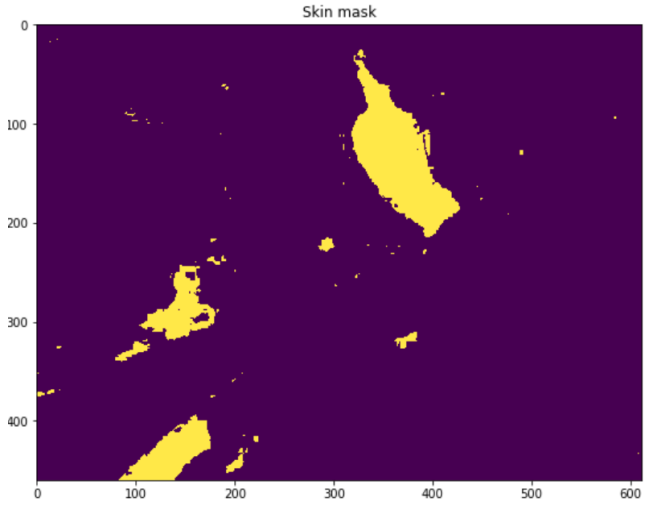
Data Augmentation can be accomplished by domain adaption from synthetic domain to real domain, that is, re-draw synthetic image in real style. We thus apply Cycle-GAN (Jun-Yan Zhu et al, 2017 (https://arxiv.org/abs/1703.10593)) for this task.

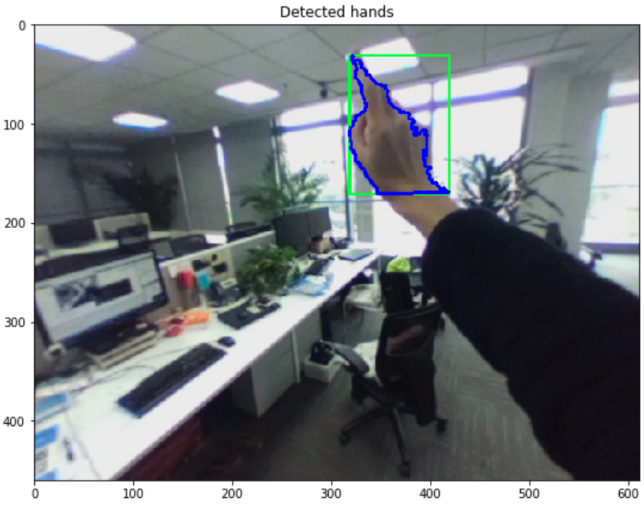
Experiment

SIFT + Skin Color Detector



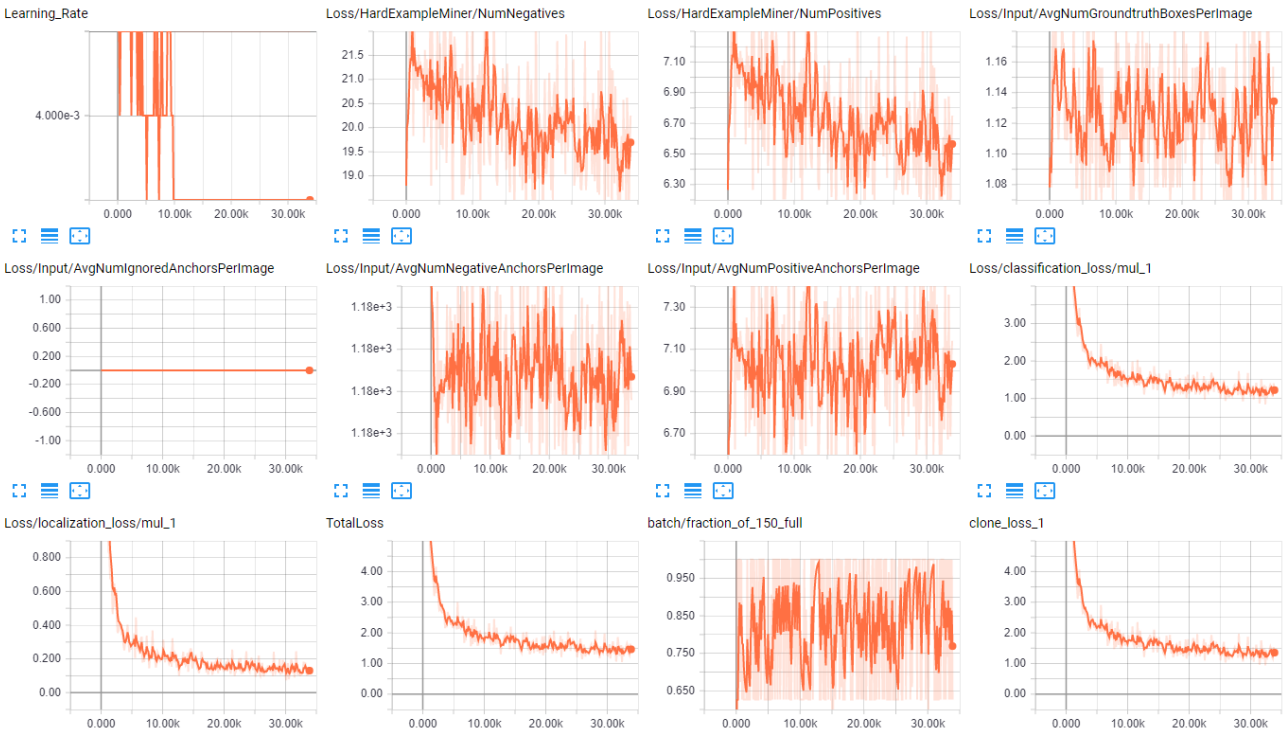




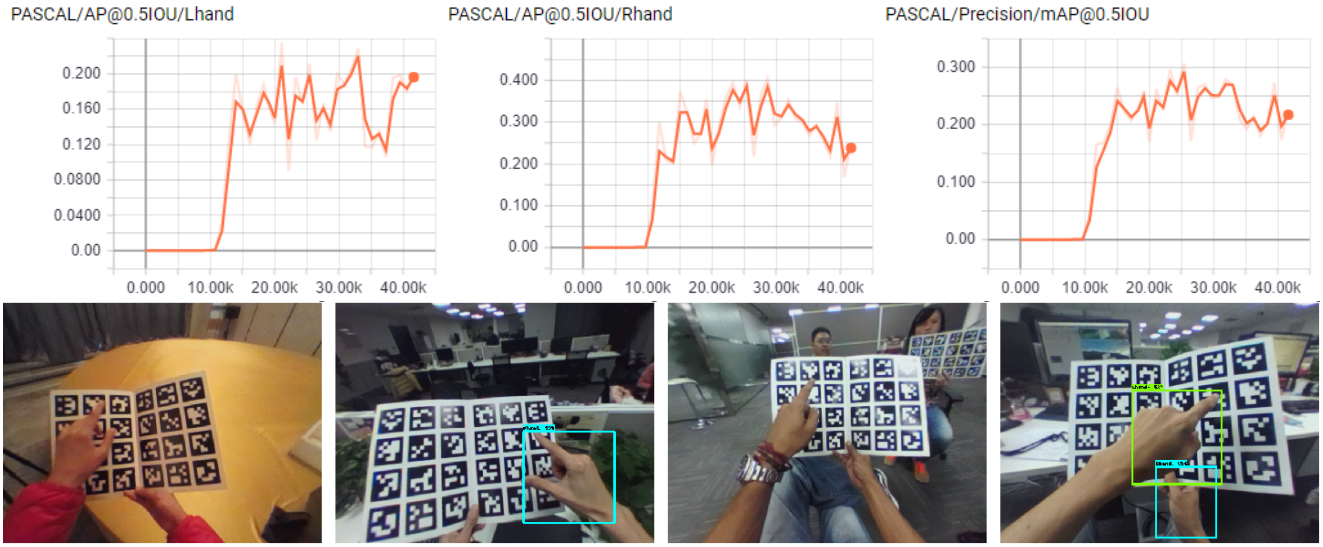


Single Stage Detector

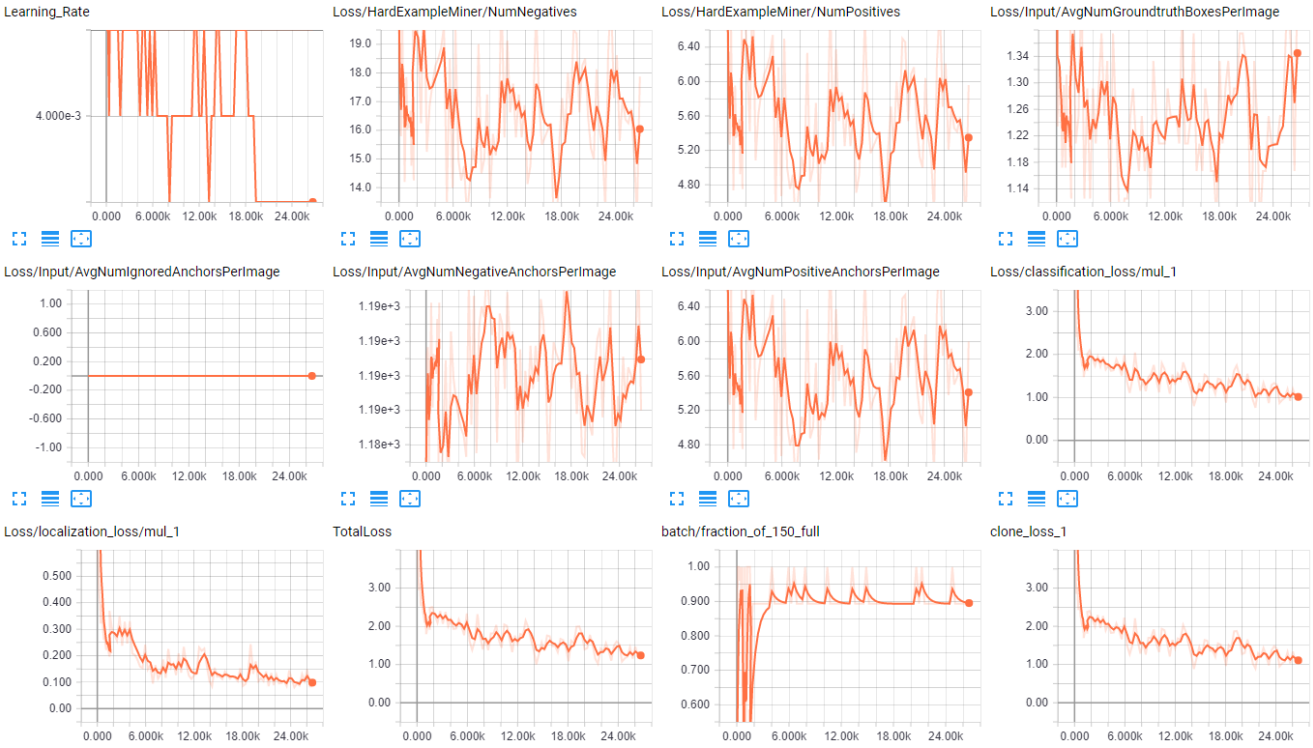
- **Training on synthesis data**



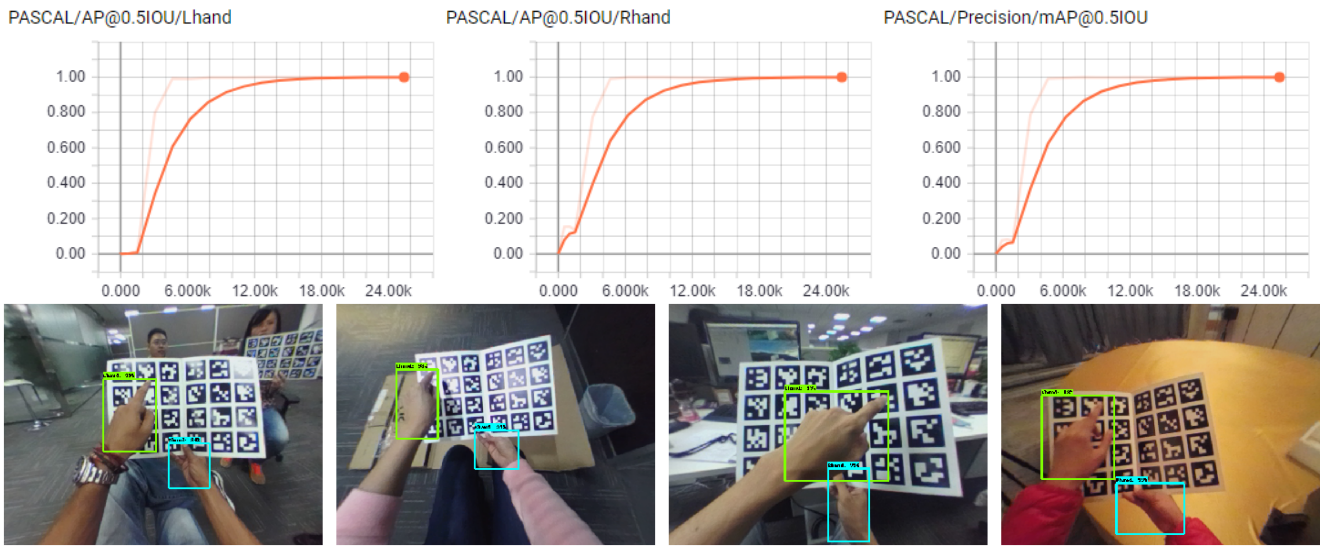
- **Testing on validation real data**



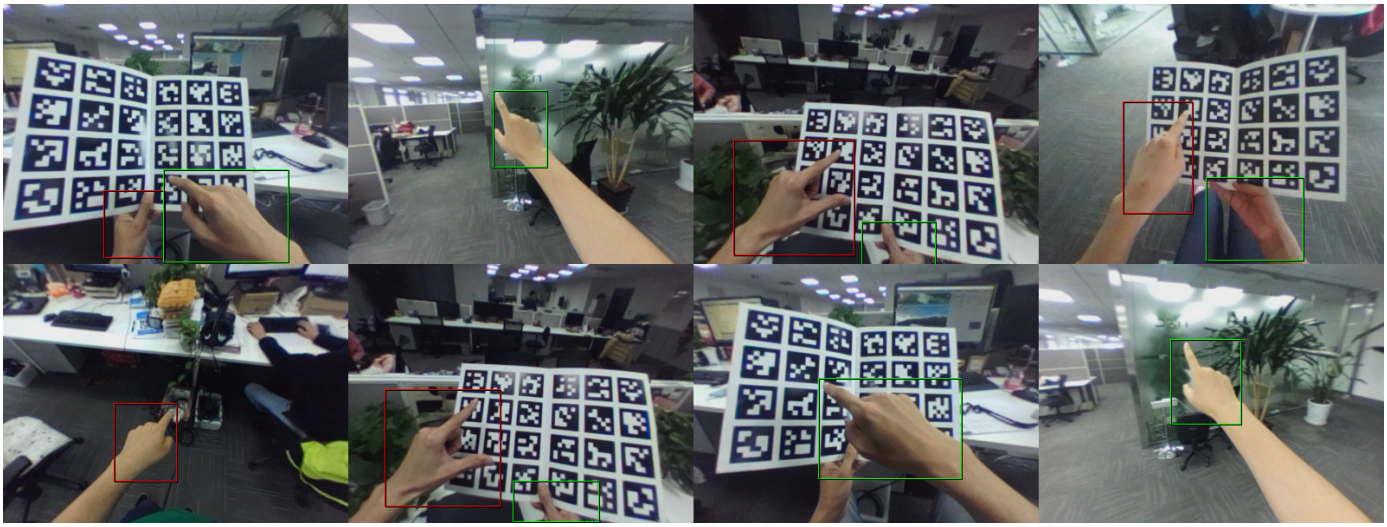
- **Training on real data (fine-tune)**



- **Testing on validation real data**



RetinaNet



Domain Adaptation

We tried three domain pairs, (a) synthetic image to real image, (b) hand mask to real image, (c) hand mask with simulated arm constructed by label to real image. The result is in figure 6. In (a), the domain adaptation from synthetic image to real image is only a color space transform that makes color of synthetic more realistic, but it didn't catch any feature of real data, such as screens, wood tables. In (b) and (c), the result is not satisfying. We think this is a consequence cauesed by scarce of variety of scene in real images, and lack of mask information (color represent table, screen … etc). We didn't apply domain adaptation to detection model

(a)

(b)

(c)

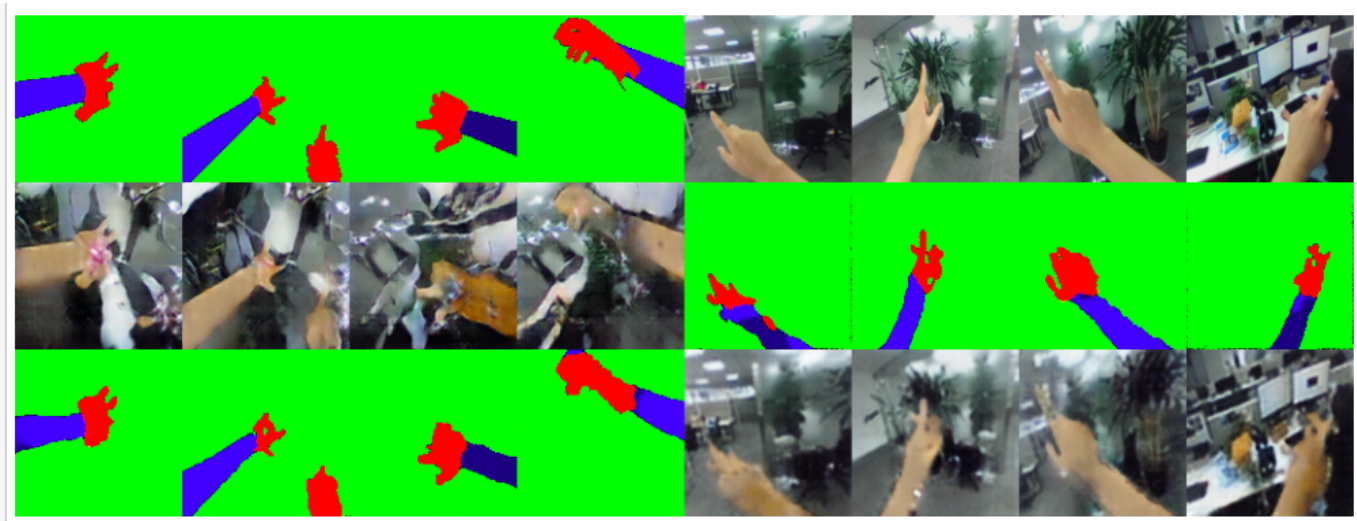


Figure 6. (a) synthetic image to real image, (b) hand mask to real image, (c) hand mask with simulated arm constructed by label to real image. Each image constisted by 3 rows, the rows from top to bottom are original image, adaptated image, reconstructed image respectively.

Conclusion

Obtaining training data is a big issue in object detection. In this task, since collecting/labeling real data is expensive, the tagged real data is only a little. The use of 1. synthetic image, 2. labeled image, and 3. unlabeled are equally important. Our experiment result shows our method can effective use 1. and 2. efficiently, while 3. can't. Since synthetic image is far from real image, fine tuning on real images is important to get better performance.