

# CS168 Report

---

**Zecheng Fan** 704822675

**Xilai Zhang** 804796478

## Introduction

---

Designed to convert all suitable input images to a suitable image for TED analysis and comparison.

## Procedure

---

```
$ python3 general_filter.py -i INPUT_FOLDER -o OUTPUT_FOLDER
```

```
$ python3 glass_blur_filter.py -i INPUT_FOLDER -o OUTPUT_FOLDER
```

## Details

---

`general_filter.py` takes in raw photos to give cropped images with standardization:

- Import images from the INPUT folder and use pretrained facial recognition model to recognize faces in these images and locate landmarks
- Compare positions of landmarks to the standard positions and compute the rotation matrix
- Calculate rotation angles and discard images with undesired rotation angles
- Process the remaining images, perform an affine transformation to standardize images
- Crop the upper half of the face based on preset points
- Output cropped images into the OUTPUT folder

`glass_blur_filter.py` takes in cropped images and output images that are sharp and with no glasses on:

- Read images from the INPUT folder and discard images that are less than 8KB
- Convert the remaining images into tensors and pass in the glasses model and the blurriness model
- Gather results from the 2 models and output images that pass both tests into the OUTPUT folder

## CNN models

---

RESNET34 is used to train both models - glasses filtration model and blurriness filtration model. RESNET34 has the best performance among all models including RESNET50, DENSENET and VGG16.

Each model is trained using hand labeled data from UMDFaces dataset after processing raw data using `general_filter.py`

Use `preprocess.py` to pack and label training and test sets.

At last, we use `resnet.py` to train each model. We use 40\*5 epochs since our training set is small. We keep the model with the best test score.

Size of each training set: 120

Size of each test set: 40

## Note

---

In our submitted code, we removed all trained models since the models are too large. The complete version can be found on Google Drive at

<https://drive.google.com/drive/folders/1BOcNzY65N6DotZ1EwMTm851xenshu4cT?usp=sharing>

## Reference

---

<https://www.pyimagesearch.com/2018/04/02/faster-facial-landmark-detector-with-dlib/>

<https://arxiv.org/abs/1611.01484v2>