Human Race Detection from Facial Images

Project Proposal

COMP3359 – Artificial Intelligence Applications

Deni Susanto (3035361635)

1 Background

Human species can be divided into several groups based on their visual traits, e.g. skin tone, facial features, hair type, etc [1]. These groups are called the "*racial groups*". The terms and criteria used to identify someone belonging to a group have always been changing from time to time [2] [3]. Even the existence of *"race"* itself is still a debatable topic as some researchers even claimed that race does not actually exist and has no scientific basis [4]. Regardless of the true existence of race, the majority of our society still accept the concept of it, put themselves into a racial group, and able to visually distinguish human into a major racial group. With the advent of artificial intelligence (AI), especially in the field of computer vision, it is definitely compelling to teach a machine classifying us, human species. This application can be useful in public demographic study.

2 Objectives

There are three objectives in this project:

2.1 Build a Model to Classify Images into Racial Groups

The main objective of this project is to build a model that is able to accurately classify human into 5 major racial groups, namely White race (Caucasian), Black race (Ethiopian), Asian race (Mongolian), and Indian race (Caucasoid-Australoid), and others.

2.2 Infer the Relation between Facial Features and Race

To investigate the impact of race to human facial features, e.g. facial structure, ratios, and skin tone. Since deep learning models are likely to act like a black-box, i.e. hard to infer features relation, this project will also explore shallow learning algorithms which, hopefully, will be able to measure facial features significance in determining race.

2.3 Exploring, Experimenting, and Learning

This project will also serve as a tool to explore, experiment, and learn the various technology in artificial intelligence, especially in the field of computer vision. Different approaches will be used to build the model. With the challenges faced in implementing various methods, this project should help overcoming the steep learning curve of understanding AI techniques and algorithms.

3 About the Data

The dataset is retrieved from *UTKFace*, a large-scale face dataset with labelled information such as race, age, and gender [5]. The data consists of both male and female faces with age range from 0 to 116. The race information, which the project is interested in, is split into 5 groups, i.e. *White, Black, Asian, Indian,* and *Others* (including Hispanic, Latino, Middle Eastern, etc.). In total, the dataset consists of 24,106 images with different dimensions which contain faces in it. Unfortunately, there is data imbalanced between the *race* labels, with *White:* ~10.2k, *Black:* ~4.6k, *Asian:* ~3.6k, *Indian:* ~4k, and *Others:* ~1.7k. Additionally, there appear to be some mislabelled data as well, i.e. missing information or wrong label formatting.

4 Methodology

4.1 Data Pre-processing

The images retrieved from the *UTKFace* dataset is far from clean. Almost all the images have noise from the background or unnecessary objects for the model to train on. Additionally, the images are in different sizes and imbalanced.

4.1.1 Face Detection

For every image in the dataset, it will go through face detection to check if the face is visible on the image, and the number of faces in the picture. The purpose is to filter out any images that does not have face in it or in a poor quality such that the face is not detected by the face detection algorithm. This project will utilize the *Dlib*'s pretrained face detection algorithm which will return bounding boxes indicating the location of the faces in the image (refer to Figure 4.1).

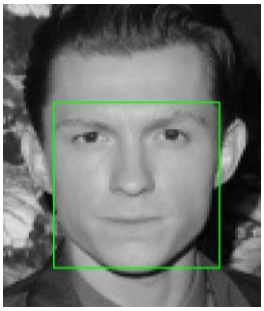

Figure 4.1 Dlib's Face Detector

4.1.2 Facial Landmark Detection

This step is done after the bounding boxes, representing face location, has been obtained. The bounding box will serve as the region of interest (RoI) for the facial landmark detection algorithm to work on. This project will utilize the pretrained 68-points facial landmark detection [6] to detect 68 coordinates of critical points on the face (refer to Figure 4.2). These points will be used later to extract facial structure features and for face alignment.

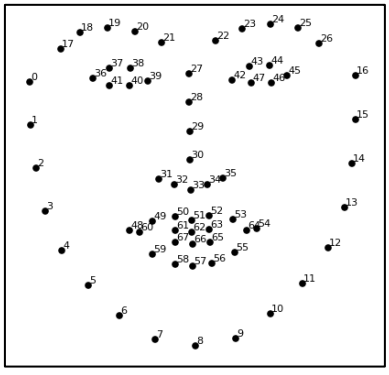

Figure 4.2 Facial Landmark Detection 68 Points

4.1.3 Face Alignment

Aligning the face for all the images is intuitively a good approach since it will help reducing the model capturing unnecessary features due to different faces orientation. Using the coordinates from the *facial landmark detector*, the orientation of the face (slope) can be calculated. For example, the coordinates of the 2 eyes can be used to calculate the slope between them. Then, using image processing techniques, the image can be rotated to such that the face will be aligned perpendicularly to the horizontal axis. See Figure 4.3 as an illustration.

Figure 4.3 Original image (left) and aligned image (right)

4.1.4 Crop to Face-only Images

The aim of this step is to clear out any noise, e.g. background, cloth, or other objects, so that only the face appears on the image. This is because the project aims to train a model that could predict race based on facial image only, so by reducing the image to face-only, it should help reducing the chance of the model learning unnecessary features.

Figure 4.4 Expected result of face-only image

To crop the image, the facial landmark coordinates will be used to determine where to crop. However, the facial landmark detection does not give the location of the forehead. Therefore, the golden ratio of human face could be useful in this situation, i.e. the average height of human forehead is 0.5 of the distance from eyebrow to chin. See Figure 4.4 for expected result.

4.1.5 Data Up-sampling and Augmentation

To address the imbalanced in the dataset, data up-sampling and augmentation will be used on the training set before feeding it for model training.

4.2 Modelling

As mentioned in the objective, this project will explore and experiment on different approaches. The next subsections will list the approaches that are likely to be used.

4.2.1 Convolutional Neural Network (CNN) Model

The first approach is to train a CNN model. Currently, the exact model architecture is still being considered. It is likely that the architecture will be based on some other popular architecture

that have been proven to work well, e.g. VGG-16, AlexNet, ResNet50, etc. However, since the computational resource is limited (architecture like VGG-16 has over 140 million parameters [7], which is too complex), this project might build a customized version of the mentioned architecture instead, while still applying the unique essence of the architecture. Figure 4.5 is an example of a custom VGG net architecture with less complexity [7]. It has less convolution layers and less nodes in the fully connected layer compared to any VGG nets version.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 256, 256, 32)      896
_________________________________________________________________
dropout_6 (Dropout)          (None, 256, 256, 32)      0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 256, 256, 32)      9248
_________________________________________________________________
dropout_7 (Dropout)          (None, 256, 256, 32)      0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 128, 128, 32)      0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 128, 128, 64)      18496
_________________________________________________________________
dropout_8 (Dropout)          (None, 128, 128, 64)      0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 128, 128, 64)      36928
_________________________________________________________________
dropout_9 (Dropout)          (None, 128, 128, 64)      0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 64, 64, 128)       73856
_________________________________________________________________
dropout_10 (Dropout)         (None, 64, 64, 128)       0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 64, 64, 128)       147584
_________________________________________________________________
dropout_11 (Dropout)         (None, 64, 64, 128)       0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 32, 32, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 131072)            0
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 655365
=================================================================
Total params: 942,373
Trainable params: 942,373
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.5 Custom VGG net [7]

4.2.2 Transfer Learning

Another approach is transfer learning. The transfer learning will use a pretrained model to extract features from the images and the only model being trained is multi-layer perceptron (MLP) model to classify the image from these extracted features. This approach will require less computational resources. And finally, its performance can be compared with the fully trained CNN model.

4.2.3 Shallow Learning Algorithms

Since both CNN model and transfer learning approach can be considered deep learning models, it is hard to infer which features are significant in determining one's race. Therefore, less complex models will also be considered, e.g. SVM, Random Forest, Multinomial Logistic Regression, etc. The features for the model input will be extracted from the facial landmark

coordinates (refer to section 4.1.2). One concern is that the landmark information might not be sufficient to predict one's race.

4.3 Evaluation and Analysis

Evaluation will be done for every model built in this project. If there are some issues with the model, e.g. overfitting or underfitting, then counter measures will be taken to fix the issue. However, there is indeed some concerns for CNN models evaluation. Since training CNN model from scratch can be time consuming, there might be a limited trial-and-error attempts in remodifying and retraining the model, e.g. change data input, adding layers, or even completely change the model architecture.

4.4 Platform, Tools, and Libraries

Due to the high resource requirement in training deep learning models, this project will be developed under the HKU's GPU Farm server, which is equipped with a powerful GPU (NVIDIA GeForce GTX 1080 Ti) running on Linux OS. The project will be fully coded in Python using mainly machine learning libraries from Scikit-learn, Dlib, and Keras with TensorFlow-GPU as the backend. The image processing will be mainly utilizing the OpenCV2 library.

5 Project Timeline

Date	Milestone
April 2	Done doing various research on the topic and assess the methodology feasibility
April 6	Finished GPU Farm setup, developed algorithms for data pre-processing, facial features extraction, and done data cleaning
April 10	Done research for CNN model architecture, done research on pretrained model to be used for transfer learning, developed code for CNN model training.
April 17	Finished at least 1 CNN model training, model evaluation, and analysis
April 18	Submitted interim report and video
April 25	Done transfer learning, model comparison, evaluation, and analysis
April 30	Done shallow learning models, tuning, evaluation and inference
May 5	All approach comparisons, identify limitations, and conclude work
May 9	Submitted final report and video

6 References

[1] M. J. Bamshad and S. E. Olson, “Does Race Exist?,” Scientific American, 2004.

[2] J. F. Blumenbach, T. Bendyshe, P. Flourens, R. Wagner, J. Hunter and K. F. H. Marx, The anthropological treatises of Johann Friedrich Blumenbach, London: Longman, Green, Roberts, & Green, 1865.

[3] C. S. Coon, The Races of Europe, New York: Macmillan Co., 1939.

[4] S. Worrall, “Why Race Is Not a Thing, According to Genetics,” National Geographic, 14 10 2017. [Online]. Available: https://www.nationalgeographic.com/news/2017/10/genetics-history-race-neanderthal-rutherford/.

[5] Y. Song and Z. Zhang, “UTKFace - Large Scale Face Dataset,” GitHub Pages, 2017. [Online]. Available: https://susanqq.github.io/UTKFace/.

[6] Z. Zhang, P. Luo, C. C. Loy and X. Tang, “Facial Landmark Detection by Deep Multi-task Learning,” ECCV, 2004.

[7] S. K. Basaveswara, “CNN Architectures, a Deep-dive,” Toward Data Science, 27 08 2019. [Online]. Available: https://towardsdatascience.com/cnn-architectures-a-deep-dive-a99441d18049.

[8] E. Kolbert, “There’s No Scientific Basis for Race—It's a Made-Up Label,” National Geographic, 2017. [Online]. Available: https://www.nationalgeographic.com/magazine/2018/04/race-genetics-science-africa/.