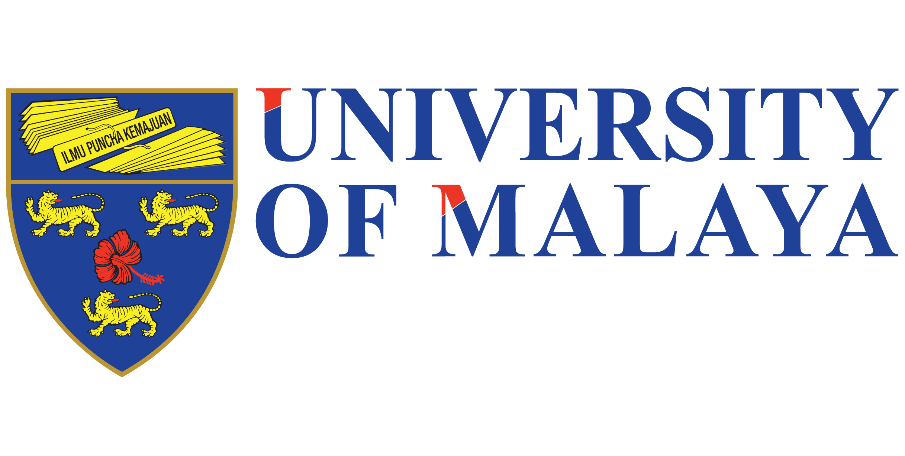
6r

WQD 7004 - PROGRAMMING FOR DATA SCIENCE

SEMESTER II - SESSION 2018 / 2019

GROUP PROJECT TOPIC

TRAINING A CONVOLUTIONAL NEURAL NETWORK TO DO FACIAL RECOGNITION USING BIG DATA

GROUP MEMBERS

CHEAH JUN YITT WQD180107

TAN YIN YEN WQD180108

CHOO JIAN WEI WQD180124

INSTRUCTORS

DR HAMID TAHAEI

SUBMITTED BY

26th MAY 2019

Table of Content

1. Introduction
   1. Facial Recognition
   2. Convolutional Neural Network
2. Objective
3. Scenario
4. Methodology
5. Results
6. Conclusion

# Introduction

## Facial Recognition

Facial recognition system is a technology used to identify or verify a person from a digital image of the person. The application of facial recognition system is immense. It has been used in various systems such as access control in security systems, commercial identification and marketing tool, video surveillance, indexing of images, and various social media platforms related functionalities.

On November 2017, Apple Inc. released iPhone X with a facial recognition system called Face ID installed. Face ID allows user to unlock the phone by looking at the front camera of the phone. (<https://www.theverge.com/2017/9/12/16288806/apple-iphone-x-price-release-date-features-announced>)

On December 2016, Amazon launched a chain of convenience store named Amazon Go in the United States. The store concept uses facial recognition and various technologies to automate the purchase, checkout and payment steps associated with a retail transaction. Customers can purchase products without checking out at a cashier or self-checkout station. (<https://www.theverge.com/2016/12/5/13842592/amazon-go-new-cashier-less-convenience-store>)

Add 1 more application on security

There are many more applications of facial recognition on improving business product, consumer experience and even advancing a nation’s interest.

## Challenge

The challenge of doing facial recognition on digital image is the difficulty of extracting facial features from pixels value. The recognition problem is made difficult by the variability of a face in the image. This includes variability in head rotation and tilt, light intensity and angle, facial expression, aging and other factors**. (Woody Bledsoe, 1966)**

Traditional methods:

https://www.researchgate.net/publication/271584966\_Face\_Recognition\_Challenges\_Achievements\_and\_Future\_Directions

Neural Network advantage

CNN models have been the state-of-the-art models for image recognition since the Imagenet breakthrough (source).

(The basic of CNN) image padding, convolution, pooling,averaging etc.

In this project, we do not design the CNN layers, instead we have selected the CNN models from a list of established CNN architectures such as VGG16, ….

These architectures were among the state-of-the-art CNN architectures on predicting the ImageNet labels at their times. However, the best CNN architectures in terms of accuracy are often difficult to train due to large number of parameters to train. For example, VGG16 has an accuracy of … but parameters …. The information per parameters (information density) is very low. In this project, we are using the MobileNetV2, which have a very good accuracy versus number of parameters tradeoffs (source). This project attempts to examine the performance of MobileNetV2 on doing facial recognition instead of general image recognition. The accuracy and amount of time spent on training will also be recorded as a form of benchmark.

(Talk about bottleneck in mobilenetv2)

# Objective

The objective of this project is to predict the identity, gender, and age group of a person from a given digital image of the person using convolutional neural network (CNN) model. This can be achieved by training three separate CNN models using large number of images data to predict the age group, gender, and identity of a person.

The first CNN model will be trained on the Labelled Faces in the Wild (LFW) dataset. (lfw source) The dataset contains 13,233 images of 5,749 people detected and centered by the Viola Jones face detector and collected from the web. Each image is labelled. This allows a model to extract important facial features to correctly identify the person in the image.

The second CNN model will be trained on the UTKFace dataset (https://susanqq.github.io/UTKFace/) to predict the gender of a person inside an image. This dataset is a large-scale face dataset containing 23,708 images, with each image labelled by age, gender and ethnicity.

The third CNN model will be trained on the same UTKFace dataset to predict the age group of a person inside an image. The labelled age for each image will be categorized into 10 age groups, namely age of 1 to 3, 4 to 6, 7 to 12, 13 to 18, 19 to 25, 26 to 35, 36 to 45, 46 to 60, 61 to 75, and 75 and above.

Finally, all these 3 CNN models will be used to classify data unseen by the model during training. There will also be an attempt to use the pre-trained identity model to do person re-identification by classifying the identity of a person using images of classes or identities not known to the model via feature extractions on CNN layer and clustering technique, without re-training the model.

# Scenario and Methodology

## Dependencies

This project was run on the following software and package dependencies:

1. Windows
2. RStudio v?
3. R base 3.5.
4. R packages:
5. dplyr
6. As
7. As
8. keras
9. Anaconda
10. Cuda
11. Cudnn

This project was run on the following hardware:

1. I5 7400
2. gtx1070
3. 8gb ram

a) crop face using opencv (reticulate and python)

b) pre-processing, filtering etc

flow image from directory (ram constraint)

image augmentation

mobilenetv2 (source: why use - efficient:)

## 1. Identity Prediction

### Data Acquisition

The ‘faces\_data\_new’ (FDN) and the ‘Labeled Faces in the Wild’ (LFW) were downloaded at <https://www.kaggle.com/gasgallo/faces-data-new> and link here

For the LFW dataset, the deep funneled dataset is used over better data quality reason. The images went through a “deep funneled algo, and is aligned”

### Data Pre-processing

The LFW dataset contains 13,233 images of 5,749 people or classes. However, most of the class contains only 1 image. Having more images per class allow the model to generalize better on extracting the features of a specific person.

The LFW Dataset was filtered such that there are at least 15 images per class. Images were copied to separate folders, where each folder represents each class. All filtered images were cropped to include only the face using OpenCV Haar Cascade face detection algorithm. Then, for each class, 10% of the images were extracted as an unseen test dataset.

Similarly, the FDN dataset was cropped and split into train and test dataset. All classes were included since most classes have at least 10 images. (PreProcess\_Data\_faces\_data\_new)

(Appendix on running script Preprocess\_Data\_LFW.R)

Both datasets were combined as a bigger dataset with K images and N classes.

### Data Loading and Image Augmentation

For scalability purposes, data were not loaded into RAM at the beginning. Instead, using Keras built-in functionalities, images can be loaded and trained by batches by specifying the directory of the training dataset.

Images loaded were passed through an image augmentation function which randomly rotates the image by up to 45 degrees, shifts the image vertically or horizontally, or does a horizontal flip of the image.

(Examples of image augmentation)

Image augmentation is important as an augmented image has the same label. Whether the face was rotated, shifted or flipped, the augmented face still belongs to the same person. Therefore, image augmentation allows the model to generalize and extract important features of the face instead of being overfitted to features that are irrelevant in predicting the identity of a person, such as light intensity of the image, specific pose of the face (e.g. Person A having a lot of images with the head tilted to the left, or exhibit a particular facial expression) or background of the image etc. Besides, by doing image augmentation, the same image could go through infinitely many augmentation configurations. This significantly increase the amount of data we have and improve the accuracy of the model.

The output of an augmented image is a three-dimensional 150 by 150 by 3 (150, 150, 3) vector. The first dimension is the width, the second dimension is the height and the third dimension is the channels of the image.

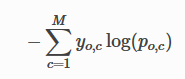
### Model Training

The dataset is split into training and validation dataset, where the proportion is 80% and 20% respectively. The augmented images in the form of (150, 150, 3) vectors were passed to the CNN model in batches of 32. The weights or parameters of the CNN models were randomly initialized by default. The input passed through the CNN layers.

The final layer of the MobileNetV2 CNN model is removed and replaced with 4 layers, namely a 2d global average pooling, a fully connected 128 neurons layer with activation function ReLu, follows by a fully connected 256 neurons layer with activation function ReLu and an output layer of X neurons with activation function softmax.

The purpose of the global average pooling 2d layer is to condense the output of the 156th layer of the MobileNetV2 CNN model to a 1d vector. The added two dense layers act as the final feature extraction layers before predicting the classes of an image using the output layer with softmax activation function.

Since, this is a classification problem with 487 classes, the loss in this case is set to categorical cross-entropy. The formula for categorical cross-entropy is:



Where *M* is the number of classes, *log* is the natural log, *y* is the binary indicator (0 or 1) if the class label *c* is the correct classification for observation *o* and *p* is the predicted probability observation *o* is of class *c*.

The model is trained iteratively for 100 epochs by adjusting the weights of the model through minimizing the loss using the RMSProp optimizer with learning rate of 0.0001.

### Model Selection

For each epoch of training, the model weights were captured as checkpoint. The best model is selected by picking the checkpoint with the lowest validation loss. At this checkpoint, the model has the best fit, where it does not underfit nor overfit on the training dataset and performs poorly on the validation dataset.

(Show loss graph)

In this case, the best model is at the “33rd” epoch. The accuracy of this trained model is then evaluated by comparing the predictions made on the unseen test dataset and the respective true label.

## 2. Gender Prediction

### Data Acquisition

The UTKFace dataset was downloaded at link.

### Data Pre-processing

The UTKFace dataset contains 23,708 images. The images were cropped to include the face only by default. The only pre-processing step was to copy the images to two different folders labelled 0 for male, and 1 for female.

### Data Loading and Image Augmentation

Images loaded were passed through an image augmentation function which randomly rotates the image by up to 30 degrees, shifts the image vertically or horizontally, or does a horizontal flip of the image.

The output of an augmented image is a three-dimensional 128 by 128 by 3 (128, 128, 3) vector. The first dimension is the width, the second dimension is the height and the third dimension is the channels of the image.

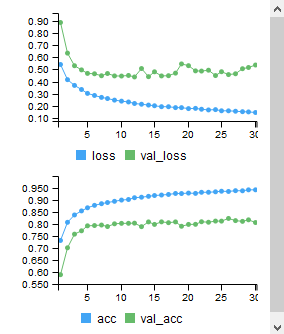
The dataset is split into training and validation dataset, where the proportion is 80% and 20% respectively. The augmented images in the form of (128, 128, 3) vectors were passed to the CNN model in batches of 32. The weights or parameters of the CNN models were randomly initialized by default. The input passed through the CNN layers.

The final layer of the MobileNetV2 CNN model is removed and replaced with 2 layers, namely a 2d global average pooling layer, and an output layer of 2 neurons with activation function softmax.

Since, this is a classification problem with just 2 classes, male or female, the loss in this case is set to categorical cross-entropy, which is equivalent to the binary cross-entropy.

The model is trained iteratively for 30 epochs by adjusting the weights of the model through minimizing the loss using the RMSProp optimizer with learning rate of 0.00002.

### Model Selection

****

The best model is selected on the lowest validation loss. In this case, the model achieved the best performance at the 12th epoch. The model is then evaluated on the test dataset.

## 3. Age Group Prediction

### Data Acquisition

The same UTKFace dataset used for gender prediction was used to do age group prediction. The dataset was downloaded at link.

### Data Pre-processing

The UTKFace dataset contains 23,708 images. The images were cropped to include the face only by default. For the entire dataset, the age spans from 1 to 116 years old. Due to functionality constraint in Keras for R, we could only do classification on the age target. Therefore, the age was divided into 10 age groups, namely age of 1 to 3, 4 to 6, 7 to 12, 13 to 18, 19 to 25, 26 to 35, 36 to 45, 46 to 60, 61 to 75, and 75 and above. 10 folders were created to represent each age group. Each image in the dataset was copied to the respective directory.

### Data Loading and Image Augmentation

Images loaded were passed through an image augmentation function which randomly rotates the image by up to 30 degrees, shifts the image vertically or horizontally, does a horizontal flip of the image, changes the brightness of the image, performs channel shift, or zooms the image by a random factor.

The output of an augmented image is a three-dimensional 128 by 128 by 3 (128, 128, 3) vector. The first dimension is the width, the second dimension is the height and the third dimension is the channels of the image.

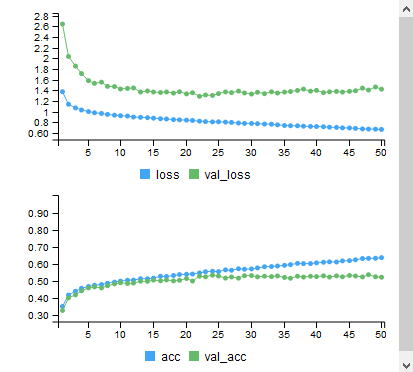
The dataset is split into training and validation dataset, where the proportion is 80% and 20% respectively. The augmented images in the form of (128, 128, 3) vectors were passed to the CNN model in batches of 32. The weights or parameters of the CNN models were randomly initialized by default. The input passed through the CNN layers.

The final layer of the MobileNetV2 CNN model is removed and replaced with 2 layers, namely a 2d global average pooling layer, and an output layer of 10 neurons with activation function softmax.

Since, this is a classification problem with just 10 classes, namely the 10 age groups, the loss in this case is set to categorical cross-entropy.

The model is trained iteratively for 50 epochs by adjusting the weights of the model through minimizing the loss using the RMSProp optimizer with learning rate of 0.00003.

### Model Selection

****The best model is selected on the lowest validation loss. In this case, the model achieved the best performance at the 22nd epoch. The model is then evaluated on the test dataset.

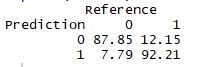
## 4. Person Re-Identification using Pre-trained Model

# Results

## Accuracy: Identity, Gender and Age Group

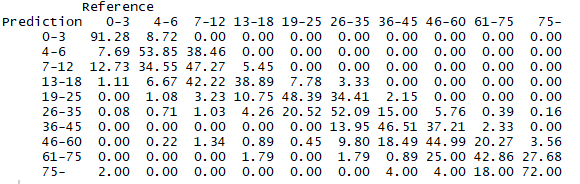
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Prediction Type | Training | Validation | Testing |
| 1 | Identity | 99.25% | 94.12% | 94.14% |
| 2 | Gender | 93.04% | 82.47% | 89.79% |
| 3 | Age Group | 61.00% | 53.55% | 53.10% |

## Confusion Matrix: Gender



The figure above shows the confusion matrix in percentage terms on the test dataset that describes the accuracy of the model on predicting the gender. The model seems to perform slightly worse on predicting male (class 0) than female (class 1).

## Confusion Matrix: Age Group



The figure above shows the confusion matrix in percentage terms on the test dataset that describes the accuracy of the model on predicting each age group. We can observe that the model has a difficulty predicting the exact age group but is able to predict into a nearer age group. It has some “broad sense” of age and rarely predict very inaccurate age group. For example, the model does not predict a person age 36 and above into the age group of “13-18”.

## Accuracy: Person Re-Identification on Unseen Test Data

The accuracy on unseen test data is 47.54%.

## Features Visualization: Train Data (sample of 7 classes)

## 

From the figure above, we can observe that the feature clusters formed are highly distinguishable by their respective class labels. It shows that our model can extract unique features on the training images to classify the respective classes or identities.

## Features Visualization: Unseen Test Data (sample of 7 classes)

## 

From the figure above, we can observe that some of the feature clusters formed can be distinguished by their respective class labels. It shows that our model could not extracts unique facial features to classify the respective classes or identities of unseen test images.

Unseen data - prediction on image (show image and show actual and prediction)

Visualization: TSNE features embedding

# Conclusion

Road map

accuracy might be low for identity, but could be improved by having more images of the unseen individual

Could be improved with re-id training (source: https://arxiv.org/abs/1611.05666)