DATS 6203: Machine Learning II

Instructor: Dr. Amir Jafari

Individual report

Date: 4/27/2021

1 Introduction

This project is predicted age, gender, and ethnicity by face recognition. Since we have 2 models, so Rayna is responsible for the age model, and I responsible for the ethnicity and gender model. For the report and PowerPoint part, we work it together.

1.1 Individual work

When I look up at this project first, I notice that most people train three models to classify 3 features. At first time, I thought it just a simple multi-label classification, which can be predicted together. Later I figure out that all three features are not at the same level, which means I could not put all these features together as age is a quantitative variable. However, I notice that gender and ethnicity all belong to the categorical variable, so I begin to think about: if I put those two features together, we might need a more complex model. But the benefit is that, as long as this model train well, the whole data can improve the accuracy. Thus, I began to think about training 2 models. As long as those 2 models train well, the accuracy of the whole data set should improve.
When I try to do the Image augmentation, I try several simple way such as rotation, flapping and zoom in or zoom out the image. However, after I do those simple method, my model accurate is reduce. On possible reason may due to the zoom out the image. What's more, I also notice that if I want to significantly improve the accuracy, I need to cut off my image randomly and mix up together to predict. Unfortunately, I did not figure out how implement in code. Thus, I did not do it.

1.2 Portion of work

- **Data processing**: For data processing, I just first do the normalization of the image and reshape to the target. When I reshape to the target, since feature gender and ethnicity has it own value, so I decided to use one-hot-encoded first in ethnicity, and concate the gender value to the encoded ethnicity. Thus, our target reshpae to 7 columns.

- **Description of dataset**:
  In the ethnicity dataset, 0 represent as White, 1 represent as Black, 2 represent as Asian, 3 represent as Indian, 4 represent Others(like Hispanic, Latino, Middle Eastern).We can also general look at the distribution of Ethnicity:

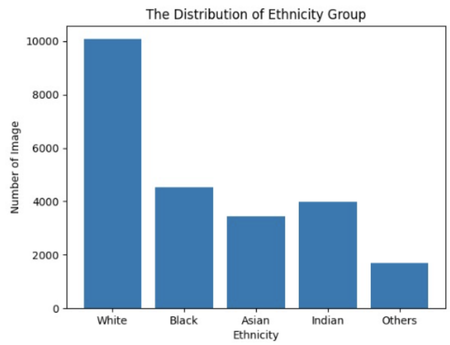


Figure 1: The distribution of ethnicity group

In Gender value, 0 represent as male, and 1 is female. The distribution is shows below:

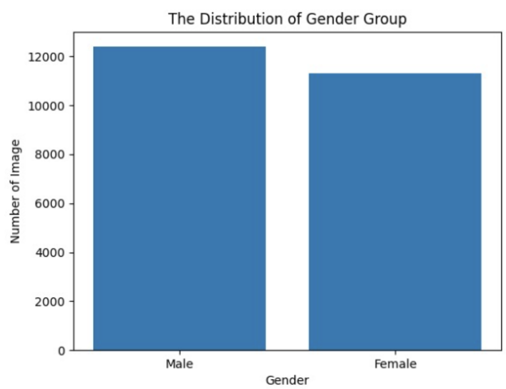


Figure 2: The distribution of gender group

- **Construct and train model**
  After I split the data to training set 70%, 15% validation set, 15% test set, I begin to do construct the model. For Gender and Ethnicity model, I use Keras to constructed the cnn model. Since the size of the photo is 48 x 48, and only has 1 color chanel, so the input shape is (48,48,1).
  Besides, In the architecture of this model, I use 6 convoluted layer as well as batch-normalization(). When adding the pooling layer, I only adding 2 pooling layer, which are Max pooling and Average pooling. The reason I only adding 2 layer is that: pooling layer basically use for reduced the calculate complexity. Since I want to improve my model, and AWS has ability to support it, so I just

convoluted as much as I can to get better performance.
What's more, I add more feature map to get as much accuracy as I can. for activity function, I choose ReLU as our activation function. Since the target is multiple-label and the sigmoid function only returns between 0 to 1, we placed the sigmoid function in the output layer of the model, which serves to convert the model's output into a probability score, which can be easier to work with and interpret.
When choosing the flatten layer, less neuron might reduce the accuracy, so I choose the neuron to 4096 with 0.2 droup out rate to avoid over fitting.
below shows the part of the code:

```
model_eg = tf.keras.Sequential([
        InputLayer(input_shape=(48,48,1)),
        Conv2D(32,(3,3),activation="relu"),
        BatchNormalization(),
        MaxPooling2D((2,2)),
        Conv2D(64,(3,3), activation="relu"),  #64, 21,21
        BatchNormalization(),
        AveragePooling2D((2,2)),      #64, (10,10)
        Conv2D(128,(3,3), activation="relu"), # 128, (8,8)
        BatchNormalization(),
        Conv2D(200,(3,3), activation="relu"),  #200, (6,6)
        BatchNormalization(),
        Conv2D(400,(3,3),activation='relu'), #400,(4,4)
        BatchNormalization(),
        Conv2D(500,(3,3),activation="relu"), #500,(2,2)


#  MLP
        Flatten(),
        Dense(500, activation="relu"),
        Dropout(DROPOUT_eg),
        BatchNormalization(),
        Dense(4096,activation='relu'),
        Dropout(0.3),
        BatchNormalization(),
        Dense(4096,activation="relu"),
        Dropout(0.2),
        BatchNormalization(),
        Dense(4096,activation='relu'),
        Dropout(0.2),
        ........
        Dense(4096,activation='relu'),
        Dropout(0.2),
        BatchNormalization(),
```

```
        Dense(7, activation="sigmoid")
         ])
```

- **Accuracy**
  I also plot the accuracy of the model I train:

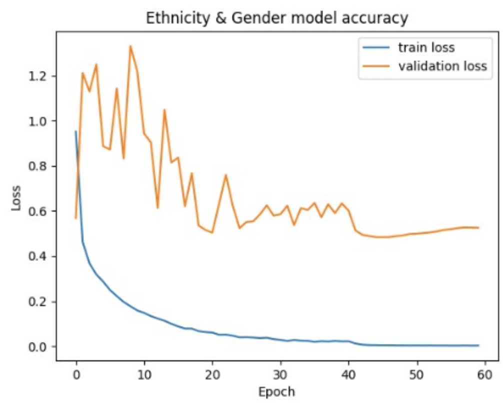


Figure 3: the accuracy of ethnicity and gender

1.3 Result

For the model I train, the best score I can get is 66%. Based on this model, the distribution of predict correctly in ethnicity and gender is shows below:

Ethnicity

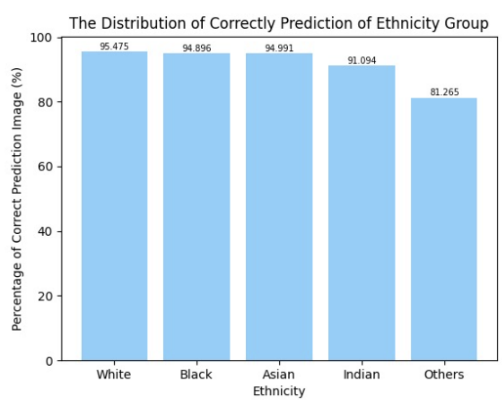
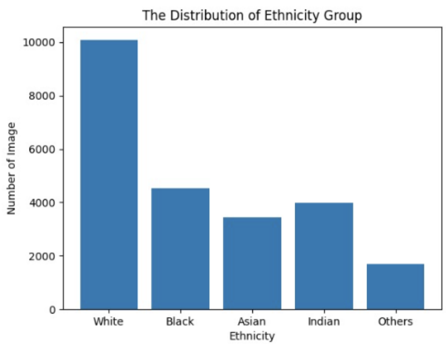
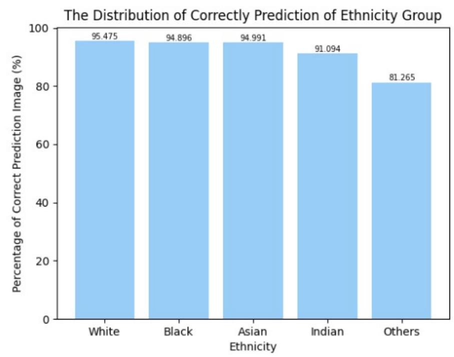


Figure 4: The distribution of prediction ethnicity

Based on figure above, White, Black, Asian, Indian have relatively high predict accuracy, as all these 4 types above 90% predict accuracy. In order to find some potential reason, I compare the distribution of Ethnicity and the prediction accuracy for each ethnic group. As shown in the graph, the racial distribution is very uneven, with



(a) 1



(b) 2

the major the white people and the minor others. However, the stratification in the split data set plays a crucial role in my model's ability to predict other races well and achieve high accuracy. On the other hand, the prediction ability of my model is weak in others.
Since ethnicity is a categorical variable, I decide to use a confusion matrix to analyze the model. Below is confusion matrix for ethnicity:

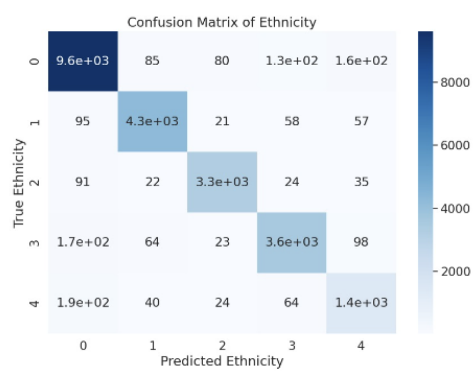


Figure 6: The confusion matrix of ethnicity

General look at the correct prediction images: Figure 6 shows that around 9600 images of white are predicted correctly, 4300 images of black predict correctly, 3300 images of Asian predict correctly, 3600 images of Indian predict correct, and 1400 images predict correctly. Continue to analyze the predicted data: Among all categorical of the correct prediction, the number of the white image is the heights. Looking back to the data distribution, the data sample of white is significantly higher than others. Thus, it may be explained that most classification of the correct image is white. Besides,even has 9600 images of white predict correctly, still has 170 images are classified as India, and 190 images predict as others. From those wrong classified example,we would say some features between White and Asian, Others are not distinguished well.

Gender

For gender value, I also visualize the distribution of correct prediction: looks like the

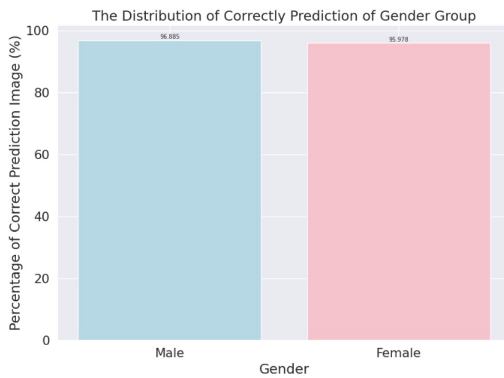
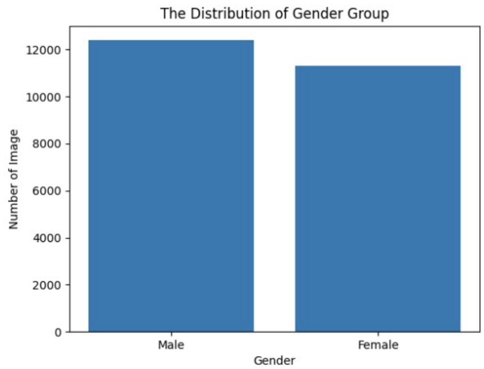
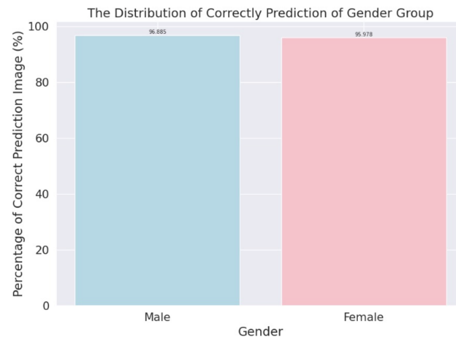


Figure 7: The distribution of correct prediction

accuracy for gender was more than 95 % both on male and female. I look back to the data distribution and compare it:



(a) 1



(b) 2

compare the distribution of gender and the prediction accuracy for each gender group. Looks like the importance of stratification is demonstrated here to allow both male and female colleagues to be accurately predicted.
Similar to the ethnicity, I use confusion matrix to analyses the model. From the

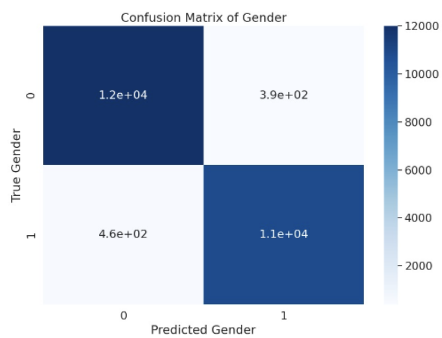


Figure 9: Confusion matrix of age

Figure, around 12000 male images predict correctly, and around11000 female images predict correctly. We can calculate some statistics to measure our confusion matrix:

1. Total $= 12000 + 460 + 390 + 11000 = 23850$
2. Accuracy : $(11000 + 12000)/23850 = 0.964$
3. Misclassification Rate (Error Rate): $(390 + 460)/23850 = 0.036 = 1 - 0.964$
4. Actural yes:$(460 + 11000) = 11460$

5. True positive rate (Recall) : 11000/114600 = 0.95986

Based on the statistic above, around 96.4% image classifier correctly, around 3.6% of image classifier wrong. Among those classified as true images, around 95.986% image indeed classified correctly.

Show incorrectly Ethnicity& Gender Image
I also use this model draw some incorrectly image: The ethnicity & gender model is

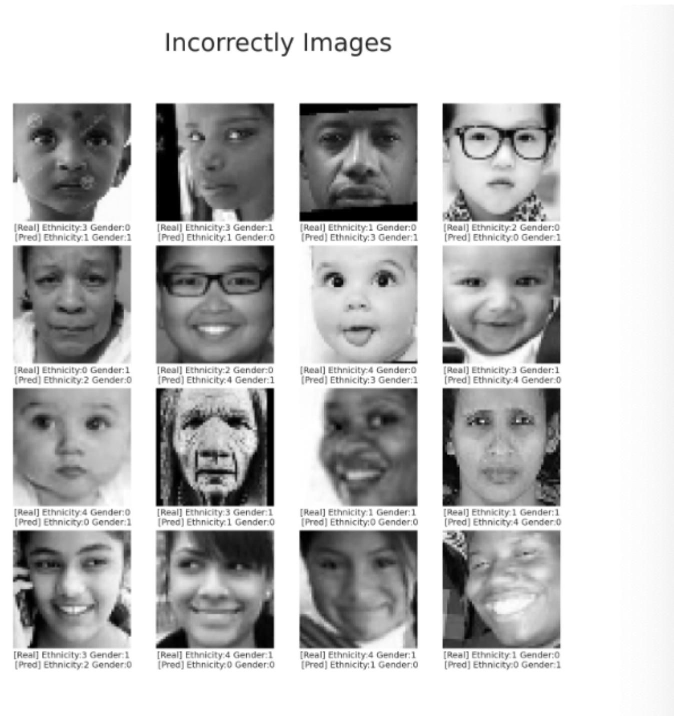


Figure 10: Some incorrectly image

inaccurate in predicting a child's ethnicity and gender. In fact, a boy is usually fully physically developed man at the age of 18 and girl is usually fully physically develop woman at the age of 16.

1.4 Summary and conclusion

Based on the model I train, most of image predict well on gender, which is meet my expectation as gender only has 2 categories. For ethnicity, we only has 4 main ethnicity group to classify it, so it may not predict accurate in real world. As in real world, if we want to divide more accurate, the ethnicity will divided beyond 4 types. Thus, if we only train the model for those 4 types, it probably not so accurate. On the other hand, some features between White and Asia is not distinguished well, probaby some detail between white and Asia did not learn well. Besides, only the simple image augmentation such as rotation, flapping, and zoom in,zoom out it not work for our image dataset, so we still need to use more stronger method such as randomly cut out the image and mix up together as input might improve the accurate.

1.5 percentage of the code

The code of model structure comes form class exercise,so the percentage of my own code is: $(11 - 9)/49 = 0.0408 = 4.08\%$

2 references

References

[1] Kaggle: Age, Gender & Ethnicity Prediction
`https://www.kaggle.com/rithikb24/age-gender-ethnicity-prediction`.

[2] Amir-jafari: Deep Learning,
`https://github.com/amir-jafari/Deep-Learning`

[3] Gentle Introduction to the Adam Optimization Algorithm for Deep Learning
`https://machinelearningmastery.com`
`/adam-optimization-algorithm-for-deep-learning/`

[4] Building powerful image classification models using very little data
`https://blog.keras.io`
`/building-powerful-image-classification-models-using-very-little-data.`
`html`

[5] Google image
`https://www.google.com/imghp?hl=en`