Facial Feature Recognition Project

1. Wenwei Wu 2. Weiyi Chen 3. Xinyuan Wang

Member 1 Contribution: Random Forest, SVM

Member 2 Contribution: Logistic Regression, Feed Forward NN

Member 3 Contribution: CNN

Introduction

How can we determine the age of an individual? The fundamental causes of facial aging are bone movement and development and skin deformations associated with wrinkles, and a loss of muscle strength. Bone development usually occurs during adolescence, while texture changes are connected to adults' most severe age-related deformations. Humans can guess the age of other people simply by looking at their faces and observing aging-related features. However, this human-made guess may be biased and deviated mainly due to racial differences and gender differences.

In this project, we propose to use several supervised learning algorithms, such as SVM, to predict the gender, age, and ethnicity of given human facial pictures and find out the optimal algorithm using a cross-validation method. We will also seek if gender or ethnicity could add difficulties to age prediction by running an age prediction algorithm conditioning on people's age or ethnicity. We would first use 70% of the pictures as a training data set to train the parameters of several models and use the other 30% to test the prediction.

The results we obtained show that our model demonstrates high accuracy in predicting gender and ethnicity but relatively low on age prediction. Though we have tried to group our ages by the states of life, the accuracy is still not high. However, prediction usually does not fall far from the actual age stage, and if we add conditions of ethnicity or gender to the prediction, we may see some improvements.

Dataset

https://www.kaggle.com/nipunarora8/age-gender-and-ethnicity-face-data-csv

The dataset we used is from Kaggle, which includes the facial images of different ages, genders, and ethnic groups. It has a total of 27305 rows and 5 columns. There is no more need for data cleaning since the author has already simplified the data.

The histograms in Appendix(data distribution) show the distributions of gender and ethnicity. There are 2 categories, male and female (denoted as "0" and "1"), in gender distribution. The number of observations for males is slightly greater than that for females. Ethnicity has 5 categories, White, Black, Indian, Asian, and Hispanic (denoted as "0" to "4" respectively). The number of observations for the white ethnic group is much greater than the rest. On the other hand, the numbers of observations among the black, Indian, and Asian ethnic groups are close.

The age data is not normally distributed; the number of observations around the age of 1 and 25 is much larger than the rest. The median age is 29, the 25th quartile age is 23, and the 75th quartile age is 45. Some outliers are also shown in the box plot. We decide to separate the age into 8 age groups based on the biological human development stages. Human biological development can affect human facial features; some specific facial features correspond to a particular age interval.

There are sample images in the Appendix. The pixel of each picture is 48×48. We flatten each image from 2d array into 1d array and find the dimension of the image data, which is 2304.

Methodologies

In general, we use 5 machine learning methods to build 3 models respectively to predict human age, gender, and ethnicity, based on given facial images. We aim to find the optimal model to identify human age, gender, and ethnicity by detecting human facial features.

Random Forest

Random Forest is an ensemble learning method for classification and regression by constructing a multitude of decision trees at training time. The prediction is based on the average results from all the trees. We choose Random forest since it is good at handle data with high dimensions. The dimension of our data is

2304, which is very high. The random forest can also avoid overfitting and improve the stability of the model's accuracy. Since we have such a large dataset, we need a stable model to make predictions.

Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that can be used for both classification and regression tasks. It can differentiate the entire dataset into different classes and find an optimal boundary between them. Since we perform classification tasks, SVM is actually a common choice. It is also effective in high-dimensional space, which fits our situation.

Logistic Regression

Logistic regression is a supervised learning classification algorithm used to model binary dependent variables (0 and 1). It is efficient to train, and it works well if we have binary dependent variables, such as the data in gender case. It's also commonly used when the data has categorical variables. There are types of logistics regression, including binary LR and multinomial LR. Multinomial logistic regression has more than 2 unordered variables. Ethnicity (5 classes) and age (8 classes) data both have multiple classes, and we can perform multinomial logistic regression on ethnicity and age.

Feedforward Neural Network (FFNN)

Feedforward neural network is an artificial neural network wherein the information only moves forward from the input nodes to the output nodes. It is simple to implement since it only has a single layer. Besides, a common advantage of neural networks is self-learning without being limited to the input provided to them. Since there is not much limitation, I choose to apply it to our three groups.

Convolutional Neural Network (CNN)

CNN is a type of deep neural network used to evaluate visual images. Shift invariant neural networks are built on the shared-weight architecture of convolution kernels or filters that slide along input features and translate equivariant outputs called feature maps. We use CNN in this analysis because the images we have are not exactly aligned, hence using CNN can help us find the important facial features and make the correct prediction.

Implementation details

Random Forest

We divide our data into 70% train data and 30% test data in random forest implementation. We perform random forest models on predicting human ethnicity, gender, and age-based on analyzing facial features. We set 500 trees in our forest because the more trees we have, the higher test accuracy is. 500 trees is a balance we made between efficiency and accuracy. We set the maximum depth of the tree to be 1000. Conventionally, even if we allow the tree to go up to depth 1000, we would get a much smaller max depth than 1000. We use the function of "Gini " to measure the quality of a split. The functionality between geni and entropy is similar, but the former is relatively more efficient.

SVM

In SVM implementation, we divide our data into 70% train data and 30% test data. We perform SVM models on predicting human ethnicity, gender, and age-based on analyzing facial features. We pick "rbf" as the kernel of my SVM models. RBF stands for radial basis function. Because of the complexity of our data, it isn't easy to find a linear boundary to classify our data. If we use the linear kernel, it would fall into an infinite loop and never see a classification boundary. RBF kernel is preferred because the complexity of the RBF kernel SVM grows with the size of the training set. We use the rbf kernel to build models to predict ethnicity, gender, and age.

Logistic regression:

In logistic regression, we import the data and separate it into three groups, respectively. Then we split the training and testing data (7:3) randomly. After that, we apply the sklearn package to perform logistic regression. The gender data is a binary logistic regression, so we use parameter penalty='l1', solver = 'liblinear.' ethnicity and age data should perform multiclass logistic regression, for which we can choose solver = 'sag.'

Feed Forward Neural Network

After importing the PyTorch package and data, we divide the training set and testing set. It is reasonable to set the parameters to be : input_size (48x48), num_epoch(5), hidden_size(500), learning_rate(0.001). Since gender, ethnicity, age all have different sizes, we then set num_classes to the corresponding ones. We obtain the information of the data loader. To perform Feed Forward NN, we input the neural network architecture by creating a class called NeuralNet. Inside the NeuralNet, we do the linear transformations (2 hidden layers) and specify the forward architecture. After that, we do the loss and optimizer, train, and test the model to get the result.

CNN

We first predict gender during implementation since it has the least amount of class number. Then, by adding additional convolutional layers to the architecture, we can determine the best architecture for gender. It turns out that the best prediction is from a five layers model, which includes 4 convolutional layers with channels 16, 32, 64, 128, and a linear layer, respectively. The last convolutional layer was able to increase the prediction accuracy by 1 percent. In each convolutional layer, we normalize our data so that the neural network will not lose performance due to significant data input before the "nn. Relu" activation code.

For the ethnicity prediction model, by a similar method, we can obtain the best prediction result from the model with 4 layers, this time with convolutional channels 16, 32, 64, respectively.

We will first create a model based on all ethnicities and genders using the life stages we have for the age prediction model. Then we will run age conditioning on different races and genders to see if we can improve the prediction. We use a similar 4 layer model as the ethnicity prediction, but with some slight change on the parameters for the best results.

Aside from changing the in-feature channels and out-feature channels in each layer, changing hyperparameters was also one attempt. The difficulty here is that we want high accuracy and want our running time to be efficient. So we have set different hyperparameters to different models for the best outcome.

During the implementation, though by adding layers shall we see performance improvements, we noticed that the loss showed significant volatility, which indicates that there are issues with model overfitting; that is, the model was fit too well for a particular sample. To deal with the initial fluctuation under the condition that a more complexed model was able to predict better, our strategy was to use a learning rate scheduler that set the learning rate after specific steps to be the product of its current rate and a gamma variable, which helps the loss converge to 0 quicker at the beginning, and therefore significantly reduces the fluctuation.

To further reduce the fluctuation, particularly the instability at the end of the training, we used "nn.Dropout" command in the fully connected layer. When the model is very complex, and the data is not enough, there tends to be some spike up at the end of the loss diagram. And by setting Dropout(0.5) indicates that at this layer of the network, the neural has a 50% chance to be abandoned and not be part of model training. This implementation decreased loss volatility and increased prediction accuracy by at most 3% in some models.

Results

Overall, CNN has the highest accuracy among all. We also observe that, in most cases, the test accuracy for gender is the highest in general. This high accuracy may be due to the low complexity of gender data. There are only 2 classes in gender, but 5 classes in ethnicity and 8 classes in age. The age data, though we have already placed ages into different stages, one may still argue that some stages can still have a tremendous amount of similar features, especially for children of a younger generation, it is difficult to estimate their age, gender, ethnicity even for a human. In addition to the age data, each class is unbalanced. For improvement, obtaining a more extensive data set on minorities, normalizing, or scaling data can be used in future studies.

SVM: The test accuracy for ethnicity, gender, and age is 0.72, 0.86, and 0.55, respectively. Overall, the result of SVM is decent, but it would be better if we can add more hyperparameters to the model. For example, "C", the Regularization parameter, can be added into the model, this can improve our prediction.

Random Forest: The test accuracy for ethnicity, gender, and age is 0.66, 0.82, and 0.52, respectively. We think there is some problem with overfitting since the training accuracy for ethnicity, gender, and age are very high, which is 0.997, close to 1. Therefore, we need to reset the maximum depth of the tree so that we can prune the tree, and solve the overfitting problem.

Logistic regression: The test accuracy for gender, ethnicity, and age is 0.83, 0.71, 0.57, respectively. Based on the result, gender has the highest accuracy compared to the other two. This might be because logistic regression performs well for binary data variables. In addition, the output has less accuracy because logistic regression may lead to overfitting on the training set, thus may not be able to perform accurate results on the test set. To improve the accuracy, we can try to use different solvers and increase the max_iter numbers in sklearn logistic regression function.

Feed Forward NN: The test accuracy for gender, ethnicity, and age are 0.87, 0.74, 0.58, respectively. This result is consistent with the general result. Age has the lowest accuracy among all three due to the complexity; to increase the accuracy, we may add more hidden layers in the network class. Or we can change the learning algorithm parameters, including batch sizes and epochs, to achieve optimum.

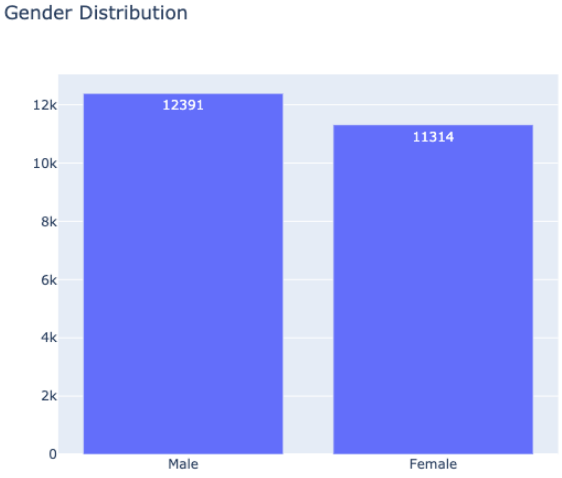
CNN: The test accuracy for gender, ethnicity, and age are 0.9, 0.77, 0.61, respectively. This result is consistent with the general effect. Age has the lowest accuracy among all three due to the complexity. But even if predicted wrong, it would not be far away from the correct interval. Once we condition age on race, we would immediately obtain an increase in the accuracy of age. For example, when we extract the data on Asians, the age prediction will improve by almost 5%. This demonstrated that different races, though mostly the same, do have unique features that show aging.

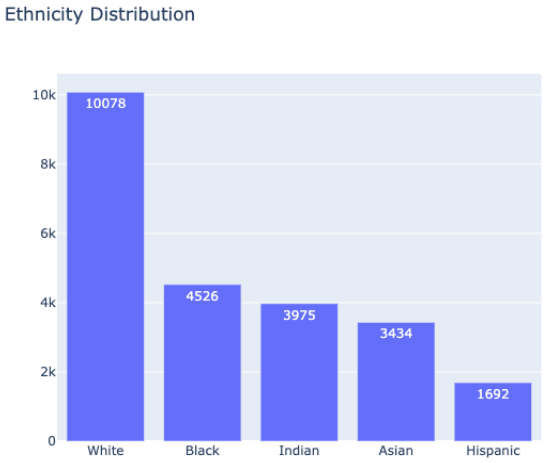
In this project, we learned that sometimes overfitting can be a significant problem during predictions. Therefore, we have learned and tried various methods to obtain a better prediction and resolve the overfitting issue. It also turns out that when we are trying to analyze images, when we are dealing with only a group of people, age prediction becomes more accurate.This implies that when dealing with less diverse groups, it is much easier to make correct predictions regarding specific features.

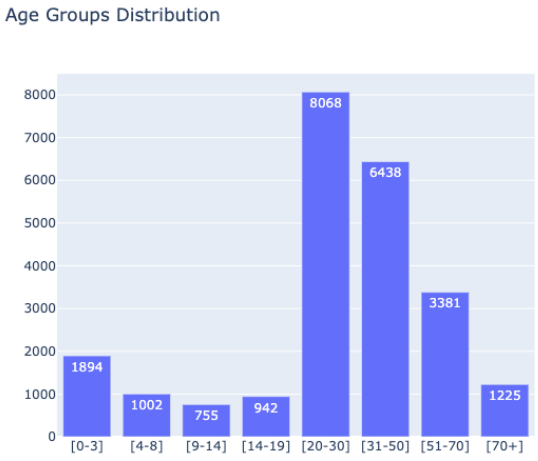
References/Citations/Appendix

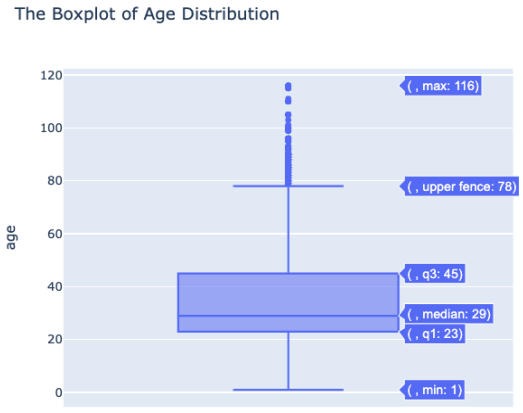
Sample images

Data distribution

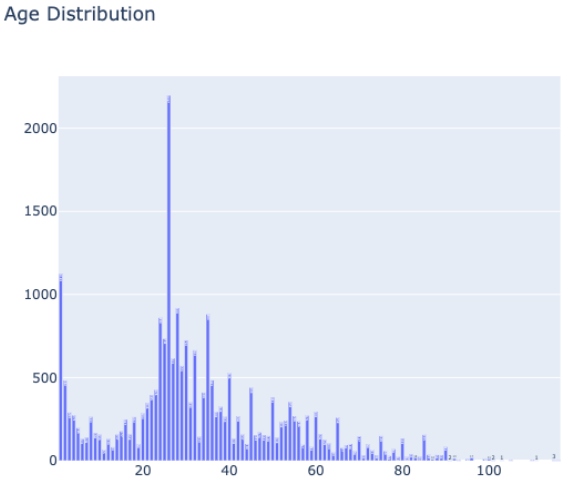






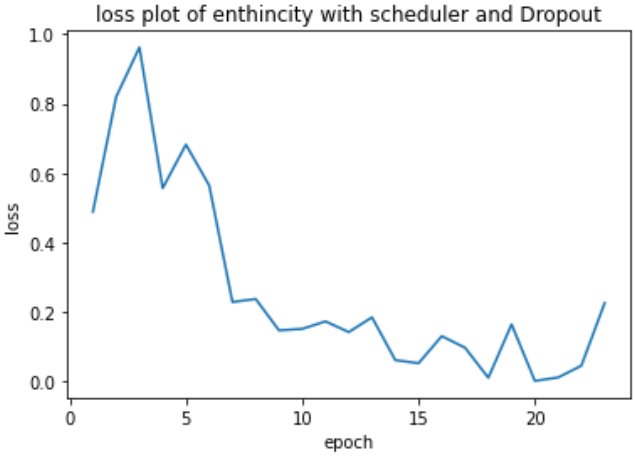


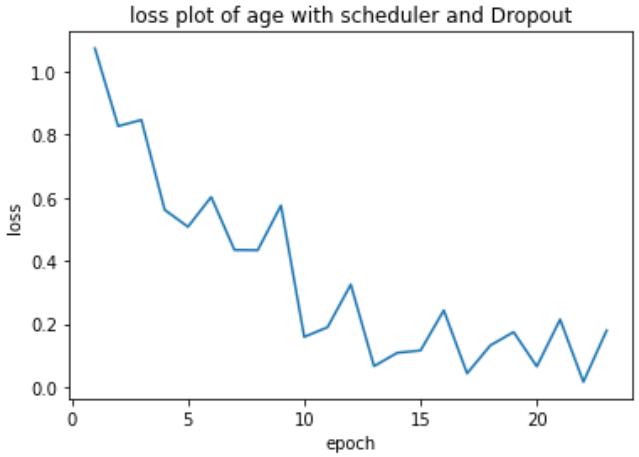
\

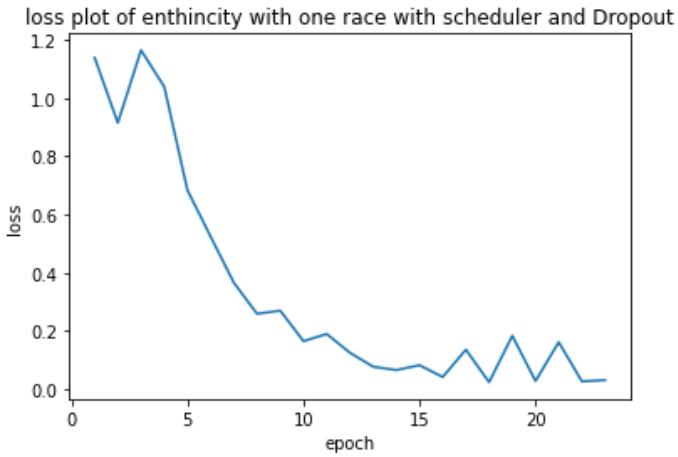

Age Distribution
2000
1500
1000
500
0
20
40
60
80
100

Loss of CNN

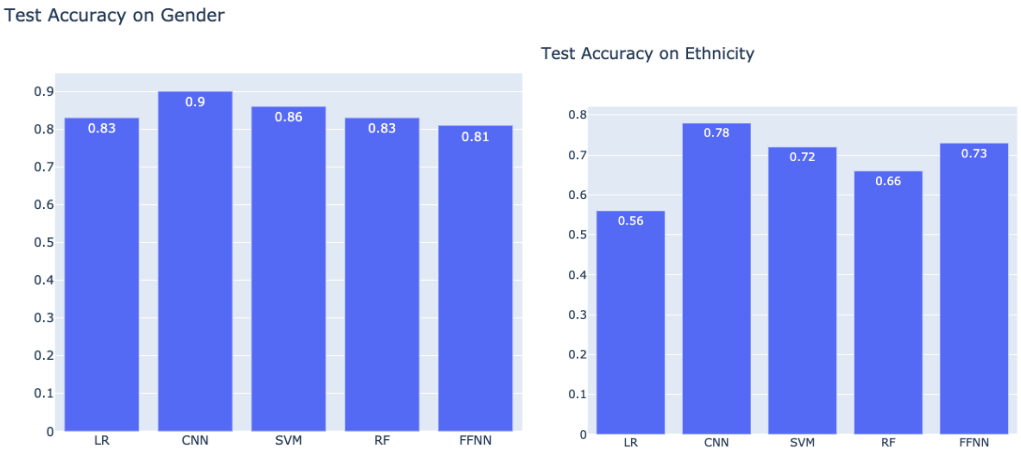

loss plot of gender with both scheduler and Dropout
loss
epoch
0.35
0.30
0.25
0.20
0.15
0.10
0.05
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


loss plot of enthincity with scheduler and Dropout
loss
epoch
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20





Test Accuracy



Test Accuracy on Age