Gender Prediction Using Convolutional Neural Network

Team: Junming Cui(jc8135) , Yuhao Ding(yd1158) , Yukun Jiang(jy2363)

1. Introduction

When we first encounter a person in real-life, our first instinct would be to predict the gender of that person, then it would be easier and more efficient to conduct a conversation afterwards. Similarly, predicting the person's gender is of vital significance for an algorithm in order to better interact with the person. The rise of social platforms and media has increased the need of facial recognition. Yet in most fields, the accuracy of existing models is far from satisfactory and a single mis-prediction could result in tremendous financial loss or security issue.

After researching online and reading papers in the area, we found that since the facial images are rather complicated, most of the simple machine learning algorithms discussed in class might not be applicable for the prediction of gender. Thus, as suggested in most researches, we propose to use the convolutional neural network as our model for the project.

We would first crop the pictures containing faces to the same size with faces in the center. Then we would use a portion of the pictures to train our CNN model, and tune the model until it gives satisfactory predictions. We eventually came up with a VGG 19 convolutional neural network trained and tuned properly, which could reach a prediction accuracy of for our validation set. In summary, we have obtained a vgg-19 convolutional neural network that gives approximately 75% accuracy in gender prediction.

2. Related Work

In the paper [1], Rothe et al. have developed some efficient CNN architectures for gender and age classification. However, since the paper was published several years ago and did not achieve high accuracy, we would like to make changes to the model using more recent machine learning methods, hoping to increase the accuracy of the model. Our attempts include enlarging the dataset by cropping the same picture from different positions, increasing the depth of the network by using the newest 19-layers convolutional neural network and utilizing momentum and weight decay for quicker convergence and less overfitting. First, by cropping the same image from different positions, we actually enlarge the dataset, thus cancelling some of the background noise in the images. Secondly, using the newest vgg-19 convolutional neural network would enable the model to give far more precise output. Thirdly, the introduction of momentum avoids wasting loops in the Pathological Curvature in neural networks, making it converge much faster.

3.Problem Definition and Algorithm

3.1 Task

We want our algorithm to perform the task below:
Input: Pictures with a clear face of any size
Output: Gender prediction (Male or Female)

3.2 Network Architecture

The detailed architure of our network is demonstrated below. In the network, there are three types of layer we use:

•Conv2d: the convolutional layer whose kernel winds with extracted features and helps to give tensor of outputs.

•ReLU: rectified linear unit which is the most commonly used activation function in CNNs.$f(x) = \max(0, x)$. A soft differentiable version of this function is called Softplus function $f(x) = \ln(1 + e^x)$

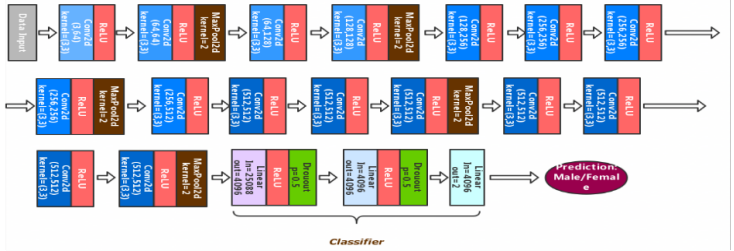
•MaxPool2d: A sample-based discretization process by using a max filter to help reduce dimensionality.

A typical pack of convolutional layers are: Conv2d ----> ReLU ----> MaxPool2d

And in the classifier there are another two types of layer we use:

•Linear: connecting every input features to every output features for preservation of information

•Dropout: with parameter p=0.5 we drop 50% of input units.



During the training process, we use cross-entropy loss as our loss criteria, and use Adam optimizer with momentum and weight decay as optimizers.

Momentum:

In our convolutional neural network model, we use gradient descent optimization algorithm to minimize the error function to reach a global minima. However, error surfaces are complex in the problem for gender classification, and may more resemble the situation where there are numerous local minima, and the gradient is trapped in one such minimum. Progress here is only possible by climbing higher before descending to the global minimum. So after searching and learning, we decided to use the momentum term to avoid such a situation. With momentum m, the weight update at a given time t becomes $\Delta wij(t)=\mu i\delta iyi+m\Delta wij(t-1)$, where 0 < m < 1 is the parameter Momentum which adds a fraction of the previous weight update to the current one. When the gradient keeps pointing in the same direction, this will increase the size of the steps taken towards the minimum.

Weight decay:

When training our neural networks, we would like to prevent the weights from growing too large, since large weights may lead to overfitting. Therefore, we learned to use the "weight decay" method, in which after each update, the weights are multiplied by a factor slightly less than 1.

References:

[1] Rothe , R.,Timofte , R., &Van Gool, L. (2016, July). Deep expectation of real and apparent age from a single image without facial landmarks. Retrieved December 2019, from http://www.vision.ee.ethz.ch/~timofter/publications/Rothe-IJCV-2016.pdf.
[2] Chih-Min Ma, Wei-Shui Yang and Bor-Wen Cheng, 2014. How the Parameters of K-nearest Neighbor Algorithm Impact on the Best Classification Accuracy: In Case of Parkinson Dataset. Journal of Applied Sciences, 14: 171-176.
[3] Lemley, Sami Abdul-Wahid, Dipayan Banik, Razvan Andonie "Comparison of Recent Machine Learning Techniques for Gender Recognition from Facial Images" presented at MAICS 2016
[4] SVM with Local Binary Patterns (LBP) Emon Kumar Dey, Mohsin Khan & Md Haider Ali "Computer Vision-Based Gender Detection from Facial Image" presented at International Journal of Advanced Computer Science. Vol. 3, No. 8, Aug 201

During the validation process, if the validation error doesn't change for three loops, the algorithm automatically do early stopping. Also, the model will be saved with the best validation error.

Here on the right is simple pseudo-code for the training Process:

```
def PseudoCode():
    # Early stopping details
    n_epochs_stop = 5
    min_val_loss = np.Inf
    epochs_no_improve = 0

    # Main loop
    for epoch in range(n_epochs):
    # Initialize validation loss for epoch
    val_loss = 0

    # Training loop
    for data, targets in trainloader:
        # Generate predictions
        out = model(data)
        # Calculate loss
        loss = criterion(out, targets)
        # Backpropagation
        loss.backward()
        # Update model parameters
        optimizer.step()

    # Validation loop
    for data, targets in validloader:
        # Generate predictions
        out = model(data)
        # Calculate loss
        loss = criterion(out, targets)
        val_loss += loss

        # Average validation loss
        val_loss = val_loss / len(trainloader)

        # Initialize validation loss for epoch
        val_loss = 0

        # If the validation loss is at a minimum
        if val_loss < min_val_loss:
        # Save the model
        torch.save(model, checkpoint_path)
        epochs_no_improve = 0
        min_val_loss = val_loss

        else:
        epochs_no_improve += 1
        # Check early stopping condition
        if epochs_no_improve == n_epochs_stop:
            print('Early stopping!')

            # Load in the best model
            model = torch.load(checkpoint_path)
```

4. Experimental Evaluation

4.1 Data Description

For training data, we are using the dataset extracted from IMDB-WIKI, consisting of roughly 38000 clear facial images, each labelled with gender and age. Then before we proceed to training, we do data pre-processing to strengthen the clarity of the image and thus the robustness of our model. The data pre-processing procedure will be detailed explained in next section.

For testing data, we choose to use another dataset from APPA-REAL DATABASE. One thing in particular, this test dataset does not come with gender labels. So in order to test our gender-prediction model, our team (3 people) each manually label the gender feature of this test dataset(around 2000 images). Our manual label error range would be around 5%.

Here are the sizes of our dataset:
30000 train data ----> 8000 valid data ----> 2000 test data

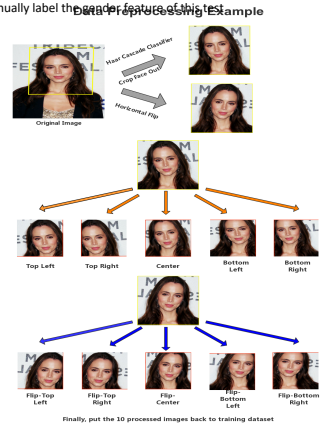
4.2 Data Pre-Processing

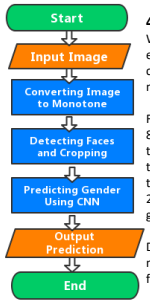
A careful inspection into the dataset gives the following features of the images:
• the face in the images might not be exactly located in the center i.e. there are a lot of noises in the images.
• the images are subject to different light conditions and people's skin colors.
To solve the above problems, we experiment and finally use the following data cleaning and preprocessing methods:

First, we utilize "Haar Cascade classifier" to detect faces in the image, and then crop the faces out. This method will also ensure that the image fed into the network is square-sized. Then we resize the image to (32 * 32), which is a bit larger than the required appropriate size of the input of the neural network (28 * 28).

Second, we would like to do a Ten-Crop on the image. We use a (28 * 28) frame to place on top left, top right, center, bottom left and bottom right to crop our the images. Then we will perform a horizontal flip on the image and implement the same procedures above again. In total we will have 10 copies of a single image and we put these 10 cropped-out images into dataset. This method will ensure biases that come from un-centered faces could be alleviated, for instance when someone is looking at a different angle or the front of face is not facing the camera. We believe such data pre-processing procedure would greatly increase the accuracy of our prediction model.

Here on the right is a vivid example of how we process an image:





4.3 Methodology

We propose to use the vgg-19 Convolutional Neural Network as our experimental methodology as explained in 3.2. We will evaluate our methodology based on three main criteria: the prediction accuracy, the robustness to noisy training data, and the computational power required. Ideally, we would like our methodology to be highly accurate in predicting gender, robust to noisy training data, and only requires a modest amount of computational power.

From the IMDB-WIKI dataset, we randomly downloaded 38123 labelled images, and split them into 30000 training images and 8123 validation images. Since the images in IMDB-WIKI dataset is randomly distributed, we assert with confidence that the training and validation sets are randomly selected. Furthermore, we downloaded 2000 images from the APPA-REAL dataset as testing set, and manually labeled them by male or female. The reason why we did not use images from IMDB-WIKI dataset for testing is that the faces in IMDB-WIKI dataset are mostly celebrities in their middle age, with a significant lack of people under 20 or above 50. By choosing testing images from APPA-REAL set, we can better test the performance of our algorithm for all age groups.

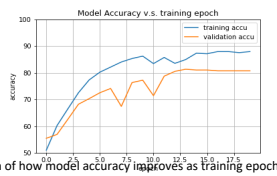
During testing period, the image is passed as input to the code. And the image is converted to Monotone mode from color mode for better face detection. This image is passed to the face detector which returns the cropped image centered around the face. The cropped face is then input to the convolutional neural network which classifies the face as a male or a female face.

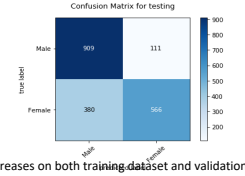
Other models can be used for gender classification are:

The K-Nearest Neighbor (KNN) classifier is one of the most heavily usage and benchmark in classification.[2] The k-NN algorithm classifies unknown data points by finding the most common class among the k-closest examples. By literacy review, we found that the advantages of KNN algorithm are: robust to noisy training data, and pretty accurate when the dataset is large. However, to use KNN we need to determine the parameter K (number of interest neighbors), and we don't know which type of distance to use and which attribute to use for best results. Plus KNN requires very high computational power since we need to compute the distances from every sample to all k-training samples around it, thus we will not use KNN-model for our project.

Emon et al. proposed a method in which they used Support Vector Machine(SVM) for gender classification [4] based on facial images. The use of Nonlinear Support Vector Machines (SVMs) are investigated for appearance-based gender classification with low resolution "thumbnail" faces processed from 1,755 images from the FERET face database. The performance of SVMs is shown to remain unsatisfactory at 78.91%. Since the accuracy is so low even when the model is particularly tuned in the same dataset, we will not consider SVM for our project.

After careful research and scrutiny [3], we have found that Convolutional Neural Network to be the most promising model for gender and age prediction. In papers by Lemley et al., CNN produces a lot better results than any of the previous methods currently in use with an error margin of 0.5% using FERET dataset.





4.4 Results

Below is the graph of how model accuracy improves as training epoch increases on both training dataset and validation dataset. We could see at beginning both accuracy on training and validation dataset increase as training epoch increases. Yet after 12~13 epochs, the validation accuracy stays more or less the same. This is an indicator that the model might be prone to over-fitting. So we do early-stopping after 13 epochs.

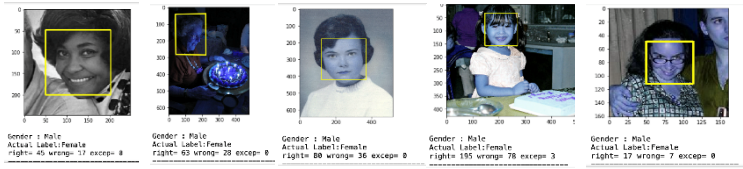
After we tune our prediction model from training, we test the model on the dataset "appa_real_release" with manually labelled gender labels, which consists of around 1200 male images and 700 female images. We get:

Prediction right= 1475
Prediction wrong= 491
Total Accuracy= 14751475+491 =75.02 %
To be more specific, we also calculate the Confusion Matrix for the testing. During the testing, Among the 1475 True Male images, we correctly classify 909 as Male and misclassify 111 as Female. Among the True 946 Female images, we correctly classify 566 as Female and misclassify 380 as Male.

4.5 Discussion

From the results above, there are two points worth discussing:

Compared with validation accuracy, the test accuracy drops by around 6%. This can be explained in term of difference of the quality and homogeneity of our validation and testing data. On one hand, the validation data is split out from the training data from "IMDB-WIKI", which is composed mostly of Male and female film celebrities in their prime. On the other hand, the testing data from "APPLE_REAL" contains people from all walks of life. In particular, it contains a lot images of children and the elder, whose gender features might not have been well learned during the training process in our model.
From the Confusion Matrix, we find that the model is around 4 times more likely to misclassify a Female as a Male than misclassify a Male as a Female (406 v.s. 136). This means the model is pretty inaccurate in predicting the gender of a Female. By a careful inspection into our training dataset, we find that while the images of Male celebrities are quite diversified, the images of Female celebrities are actually rigid and homogeneous. Then when we move to the testing dataset which contains all kinds of inhomogeneous Male & Female images, the particular celebrity-female features do not fully apply. The followings are a few typical mis-classified example:



5. Conclusions

In our project, we trained a network that could improve computer's prediction of gender based on facial images. For computational efficiency, we utilize a pre-trained VGG 19 network, and only train the final classification layer. This method, though saving us from a lot of time required for training, will affect model accuracy.

We realized that convolutional neural network is working properly even with substantial noises inside the dataset. In both our training and testing process, we use internet images rather than lab images, this leaves room for great image noises. In our project we used too few training data, thus the accuracy isn't good, however, given enough training data, the network have shown its capability of dealing with noises.

In our testing set, we deliberately choose data that cover a variety of ages, and it shows that the model isn't predicting well in some age groups, especially the elder and babies. Also, skin colors might be another factor. We think that these can be solved by a more concrete training set with less biases.

There are some things we can consider to improve:

•Models can be fully trained with better computational power. Possibly with more parameters tried.

•Data can be processed with other techniques. One such technique as we have seen in other projects is mix-up, which mix up both training images and labels by linear interpolation.

$x=\lambda*x1+(1-\lambda)*x2$,

$y=\lambda*y1+(1-\lambda)*y2$.

•Other data augmentation methods can also be used.

•The VGG19 network we used, is already a very deep neural network, but our data fed inside is 28*28, we think that the image size is too small, if we can have access to potentially bigger facial image, possibly more features can be used in the training process.