Age-Gender Identification and DCGANS on Face Regeneration and Completion

Yuan Liang
MSU ECE
liangy11@msu.edu

Yu Zheng
MSU ECE
zhengy30@msu.edu

Biyi Fang
MSU ECE
fangbiyi@msu.edu

1. Introduction

Deep learning has promised various possibilities of smartness in image processing, such as face recognition and corrupted picture completion. These amazing capabilities that our computer, mobile phone or even smartwatch have can help provide human being with new horizons of the world.

We propose a brand-new perspective of using deep learning CNNs to solve a real-world problem – facial missing part completion. Human faces share many similarities. For instance, an old man could have similar wrinkled canthus beyond limitation of races across the world. An infant baby probably could have a chubby and pink cheek no matter it is a she or he. Identifying those figures could be beneficial for various fields. For instance, the commercial company might leverage such gender and age classification to push advertisement respectively.

On the other hand, it is a matter of fact that images with completed faces is more meaningful in various contexts. For example, in police station, the witness of a crime will have much more confidence in identifying a picture captured by CCTV belongs to the robber (or not) when it contains a full face. So we asked ourselves, can we leverage those humongous amount of face image data, labeled or unlabeled, lying on the Internet to complete those with missing parts? In this work, we propose to first train and classify the age and gender group of labeled. As a secondary task, we train a model that is able to both generate faces as well as do image completion in Figure 1 and Figure 2. We select age and gender as our first step as it is representative of a much larger feature set (e.g., color tone of face skin).

To accomplish the first task, we train a traditional CNNs to do classification. In addition, we introduce Multi-Task Learning (MTL) into our model, under the rationale that age and gender might have relationship at some level as shown in the face features. As such, we will compare both traditional and MTL embeded CNNs to accomplish the same goal.

To accomplish the second task, we use deep convolution generative adversarial networks (DCGANs) where we define two neural networks (NN) to compete with each other in order to find the best figures. The intuition behind the scence is simple – training a discriminator (D) to evaluate how real the image looks; and a generator (G) to generate images which is optimally as close to reality as possible. By training them alternatively, D and G will learn properly of how to do these tasks. Meanwhile, DCGANs is enjoying as a hot topic among researchers, who are actively applying this technology to other parts of the world. However, as one of the uniquenesses of our dataset, our data not only contains faces of different gender but different groups of ages. As such, the difference between figures are drastically larger if the dataset contains toddlers, youngsters and teenagers who do not share much in common in faces. Therefore, as an assistance factor, we introduce MTL to our D to boost the performance of face generation and completion.

2. Dataset

The dataset we use in this project is the Adience benchmark [2] for age and gender classification. Examples are shown in Figure 4 The Adience dataset consists of face images fetched from Flickr. Even though the images have been cropped and aligned to locate the approximate position of face, there are still significant variances in pose, background, light and noise, which reflect the challenges of real-world imaging conditions.

The whole dataset includes more than 26,000 images from 2,284 different subjects. All the subjects are divided into 8 age groups, namely (0-2), (4-6), (8-13), (15-20),

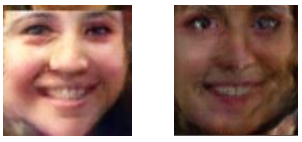

Figure 1. An example of facial generation.

Figure 2. An example of facial completion.

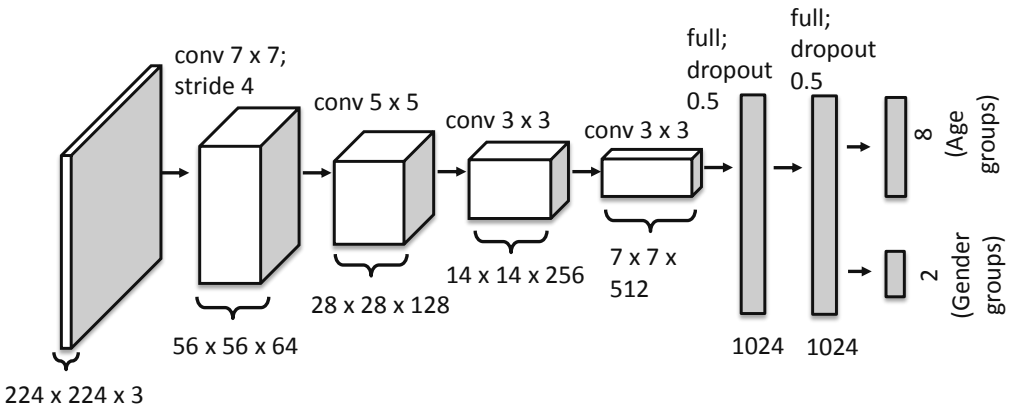

Figure 3. The structural view of our traditional CNNs.

(25-32), (38-43), (48-53), (60-). Each image is labeled by the corresponding age group and gender. We will use a subjective-exclusive cross-validation method to verify the validity of our architecture in recognition.

3. Age and Gender Classification on faces

In this section, we describe two models that aims to classify age and gender, with or without MTL.

3.1. Traditional CNNs Approach

Images are fist rescaled to 256×256 and a crop of 224×224 is fed to the network in Figure 3.

The four convolution layers are defined as follows:

1. The input picture is fed into the layer of 64 filters of size $3 \times 7 \times 7$ and stride 4, which results in maps of size $56 \times 56 \times 96$. Then a ReLU activation and max pooling of 3×3 are applied. The output maps are of size $19 \times 19 \times 64$.

2. In the second convolution layer, 128 filters of size $64 \times 5 \times 5$ and stride 1 are then applied, followed also by ReLU activation and max pooling with the same parameters as before. The output maps are of size $7 \times 7 \times 128$.

3. In the third convolution layer, 256 filters of size $256 \times 3 \times 3$ are then used, followed by ReLU and max pooling of 3×3 and stride 2. The output maps are of size $4 \times 4 \times 256$.

4. In the forth convolution layer, 512 filters of size $256 \times 3 \times 3$ are applied, followed by a ReLU activation. No max pooling is used in this layer. The output maps are of size $4 \times 4 \times 512$

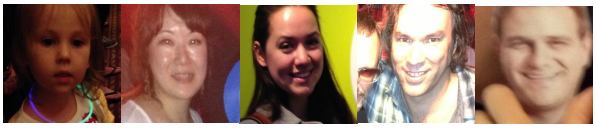

Figure 4. Examples of our dataset.

Following the convolution layers are three fully connected layers, which are defined by:

1. The first fully connected layers are constructed with 1024 neurons followed by ReLU activation and a dropout layer with a dropout probability of 0.5 in training.

2. The second fully connected layer is the same as the first one.

3. The last fully connected maps its input to the final classed through softmax activation.

The final classification is made by assigning the class with the highest probability to a given picture.

3.2. CNNs with Multi-Task Learning

In order to further improve the classification accuracy on age and gender, we introduce Multi-Task Learning in the classifier. MTL are the techniques where multiple learning tasks are solved at the same time, while exploiting commonalities and differences across tasks. The intuition behind our approach is that face and gender are two related attributes of human being as shown in the faces. In practice, instead of training two independent classifiers which classify age and gender separately, we train a classifier which can classify age and gender simultaneously.

One learning task may be able to provide useful information to another learning task on the same data. For example, by identifying the gender of the face in the image, the classifier can track more specific features on the face to infer the age, like the beard on a men's face. We expect that the MTL can improve accuracy on both age and gender classification.

The network structure of MTL is the same as the one previously described in the single-task learning except for the read out layer, where there are two types of outputs, one of a single output for the gender classification and the other of 8 outputs for the age classification. These two types of outputs are used to calculate the sigmoid loss of gender L_{gender} and softmax loss of age L_{age}, respectively, for each

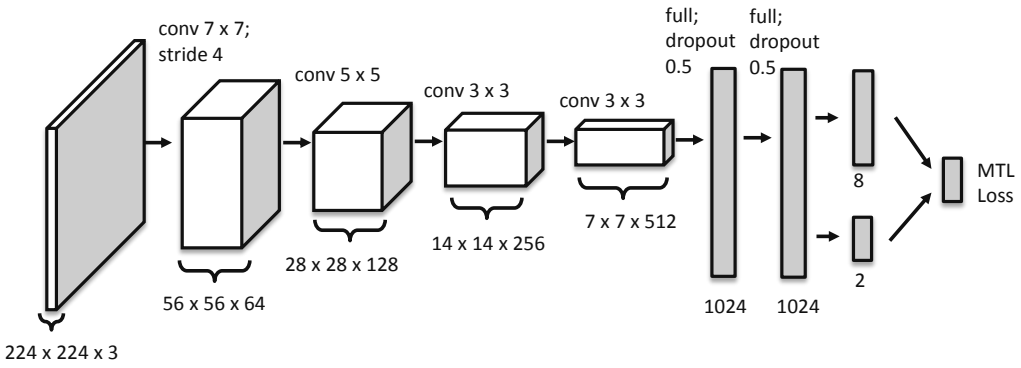

Figure 5. The structural view of our CNNs with Multi-Task Learning.

training sample. The network parameters are trained to minimize an overall loss which is a weighted average of the two losses, i.e.,

$$L_{total} = (1 - \lambda)L_{age} + \lambda L_{gender} \quad (1)$$

where λ is a hyper parameter that can be tunned. The structure of our model is shown in Figure 5.

4. Face Generation and Completion

In this section, we introduce the method of how we preprocess the data, as well as the DCGAN model that aims to generate or complete figures. We also covered our DCGANs with MTL which shares the same task as DCGANs.

4.1. Data Preprocessing

The images in our data set are collected from Flickr albums, where faces are often rotated and shifted against varied background. In order to remove those defects, we choose to use OpenFace to crop and align faces into 64×64 pixels [1]. The OpenFace takes two steps to do the alignment, fist localize faces by detecting the eyes and nose, then an affine transform is applied to output images reshaped and ready to be passed to neural network.

4.2. DCGANs

In the next step, using the images that are classified into certain age and gender group, we train a GAN to learn the features of human face, and to help produce missing part of a face image. Human faces have so many similarities, which implies that there is a lot of "reusable information." In order to fill the missing part of a face naturally, we exploit the "reusable information" in our database. As is shown that we have classified the training data into different categories. We will use data in a selected category as the training set in the face completion steps. The selection of category can be seen as assigning the training set some prior information like age and gender.

In this project we will train Deep Convolutional Generative Adversarial Networks (DCGANs) [3] to produce the missing part in a face image. This idea is to train two adversarial neural networks simultaneously. The first network is a discriminator that learns to determine whether the sample is from the data distribution. The other one is a generative model (generator) which will produce "fake" images that deceive the discriminator. After several steps of training, the optimization will reach a steady point where the discriminator is unable to distinguish the "fake" images.

Mathematically, we can view the training process of DCGANs as a minimax problem. Denote the generator as function G(z) with input z sampled from a simple distribution. Denote discriminator as D(·) with input being either images, say x, from selected data base or generator output G(z). In the training process, discriminator tries to discriminate between selected data base and G(z), that is, maximizing log(D(x))+log(1-D(z)). Simultaneously, generator tries to deceive the discriminator by minimizing log(1-D(z)). After several steps of training, the optimization will reach a steady point where the discriminator is unable to differentiate between x and G(z).

Shown in Figure 6 is the structure of the generator G(z).

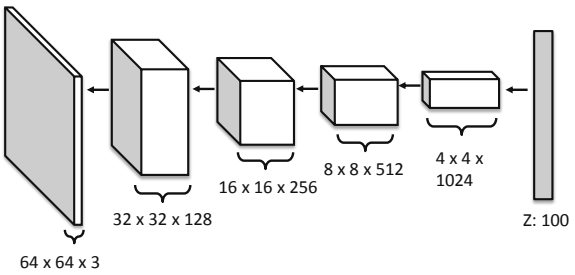

Figure 6. The structural view of generator G(z).

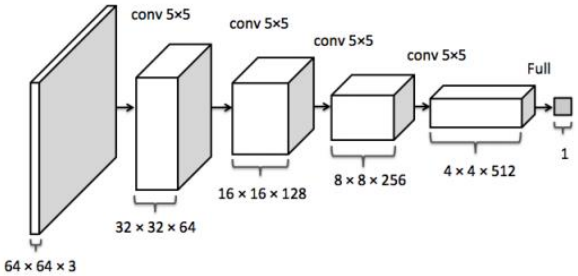

Figure 7. The structural view of discriminator D(z).

The input is a 100 dimensional random noise vector sampled from uniform distribution between [-1,1]. Following is a fully connected layer with 8192 neurons and reshaped to the size of $4 \times 4 \times 1024$. The subsequent are 4 transposed convolutional layers with stride 2 and padding, which reduce channels and up-sample features by factor of 2. Therefore the final output is an image of size $64 \times 64 \times 3$.

The discriminator D(x) is a reverse of generator as shown in Figure 7. The input image of dimension $64 \times 64 \times 3$ is passed through 4 consecutive convolutional layers with the final output of size $4 \times 4 \times 512$. The subsequent fully connected layer with softmax activation outputs the probability that x is sampled from the true distribution.

For training, we use the Adam Optimizer with initial learning rate 0.0002 and momentum term $\beta_1 = 0.5$. The batch size is set at 128 and weights are initialized from normal distribution with mean zero and standard deviation 0.02.

4.3. DCGANs with Multi-Task Learning

As mentioned earlier, MTL may be able to achieve better performance than the single-task learning on the same data. Here, we also introduce MTL in the DCGAN.

Instead of only judging whether a face image is fake or real, we design a multi-task discriminator that can also identify other information of the face image, e.g., the gender and age. We expect that the age and gender information can help the discriminator to improve its capability of distinguishing fake images. In this way, a stronger discriminator can be obtained. In order to cheat the discriminator, the generator needs to generate more realistic faces.

We add age classification as another learning task in the discriminator. The generator uses the same architecture as the previous one. The discriminator also inherits the previous architecture except for the readout layer, where 8 additional outputs are added to calculate the sigmoid loss of age for the real face images, denoted by $L_{age}(x)$. The loss function for the generator keeps the same, while the loss function for the discriminator is modified to

$$(1-\lambda)log(D(x)) + \lambda L_{age}(x) + log(1 - D(G(z))) \quad (2)$$

where λ is a hyper parameter that can be tuned.

4.4. Complete Faces with DCGANs

The quality of the completion is defined by two parts, i.e., contextual loss and perceptual loss as below [4]

$$L_{contextual}(z) = \| M \odot G(z) - M \odot y \| \quad (3)$$

$$L_{perceptual}(z) = log(1 - D(G(z))) \quad (4)$$

Where M is the mask of the missing part and $\odot$ is the element-wise product. The contextual loss $L_{contextual}$ is to make sure that the known pixels in the generated picture looks like the original picture. The perceptual loss is to make the recovered picture looks real, that is to deceive the discriminator. The overall goal is to minimize both contextual loss and perceptual loss, hence the following optimization problem is formulated

$$\hat{z} = \underset{z}{\operatorname{argmin}} L_{contextual}(z) + \lambda L_{perceptual}(z) \quad (5)$$

Where the tunning parameter λ is often set as 0.1. Finally we can find a z that can make the completed picture $G(z)$ looks natural.

5. Evaluation

In this section, we evaluate our models with respect to two tasks: age and gender classification and face generation and completion.

5.1. Age and Gender Classification

As an overview, for age classification, we have eight classes of age groups with both male and female consisting a 26,000 dataset. We do not use any data augmentation and preprocessing. Instead we directly load the figures into our model. We choose AdaDelta as our optimization method as it is widely used across the community and generally gives good results. In each iteration, we random select 200 images as our mini-batch. And in each 100 iteration, we random select 3000 figures to test the performance of our accuracy. We divide the dataset into training and testing set. The training set contains four times of number of images as the testing set. Model of traditional CNNs and CNNs with MTL is trained on the training set and validate using the testing set. We do classification on both age and gender and report the results in Table 1. As indicated in the table, the performance of the models are slightly better if we include MTL into. This could be because that age and gender share some related information as shown in the human faces. Thus, by combining both loss function into one, the shared information of both featured can be strengthened. We show the accuracy as a function of training step in Figure 8.

	CNNs	**CNNs with MTL**
Age Classification	43.1%	45.3%
Gender Classification	73.0%	75.0%

Table 1. Performance of CNNs without or with MTL

5.2. Face Generation and Completion

5.2.1 Face Generation

As mentioned earlier, we have trained two models that can generate human faces based on our dataset, one with and the other without MTL. The two models are trained on all data from our dataset. Training the generator happens twice as many as training discriminator. After 15000 iterations, we randomly selected 64 latent variables from a -1 to 1 uniform distributed samples, and generate 64 figures of human

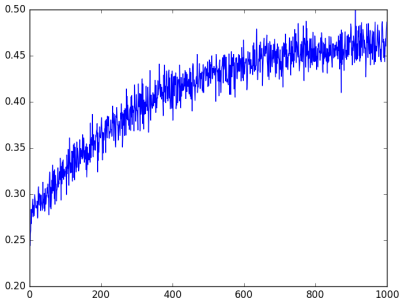

Figure 8. Accuracy as a function of 100 training steps.

faces. The results of both model without (left) and with MTL (right) is shown in Figure 9.

In order to compare the performance of both models, we recruited three volunteers who has no knowledge of the details of the projects. We told them that the two sets of 64 faces are generated by either computer programs or from real world data. The participants are asked to mark the ones they think are from the real world. In sum, participants give average 16.0 out of 64 faces that are sampled from the real world to the model without MTL, whereas they give 19.3 to the model with MTL. It is true that this is a very objective task for human being, but it could, to some levels, indicate that MTL may be beneficial to boost the performance of DCGANs in terms of generating faces on the basis of a dataset with large variance in face and from both genders.

5.2.2 Face Completion

We also do the face completion using our trained DCGANs model on pictures with missing parts. Specifically, we take pictures of our own faces, take out the center pixels sized 20 $\times$ 20 square and use DCGANs to complete the faces. As an example, the pictures of the team members are preprocessed and masked and the completed pictures $G(z)$ are shown in Figure 10. As we can see from the results, our model is not well-trained to complete missing part with appropriate colors. However, the location of missing noses and mouth are precisely located by DCGANs. We believe that with better tuning and longer training, our DCGANs model could perform much better. Hence, DCGANs works for our dataset.

6. Conclusion

In this tech report, we developed and tested a classification model for age and gender classification. We show that with proper embedding of Multi-Task Learning (MTL) into traditional CNNs, the performance of model can be boosted. On the second, we train and verify DCGANs models without or with MTL to face generation and completion. The results indicate that our model work reasonably well in both tasks.

Figure 9. Generated faces of DCGANs model without (left) and with MTL.

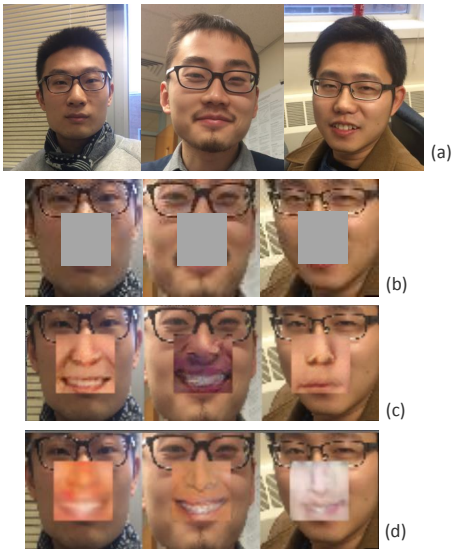

Figure 10. Source and results of face completion: (a) Sources of face completion; (b) Masked preprocessed figure; (c) Example 1 of face completion; (d) Example 2 of face completion.

References

[1] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. pages 34–42, 2015.

[3] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[4] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2016.