Child-Face Prediction using DCGAN

Kenny Tony Chakola kchakola@sfu.ca
Kunal Niranjan Desai kdesai@sfu.ca
Mayur Mallya mmallya@sfu.ca
Vipin Das vdas@sfu.ca

Abstract

The past couple of years have seen a significant increase in the application of Generative Adversarial Networks (GAN) in solving different types of challenges. Motivated by such ingenious applications of this concept in a variety of domains, we try to exploit the power of GAN by trying to predict the faces of the child, given the faces of parents. We make use of the Deep Convolutional Generative Adversarial Network (DCGAN) to generate the image of the prospective child. We measure the performance of our GAN model – both qualitatively and quantitatively – by comparing the generated child face with the ground-truth image. We also train a Convolutional Neural Network (CNN) model to generate the child-faces and show that GANs are better at generating faces that are more realistic and close to the ground-truth face image.

1 Introduction

The goal of child-face prediction is to synthesize realistic faces of children given only the images of the parents. To our knowledge, there has not been any published work in this area. However, there are some related works as mentioned below.

Alec Radford and Luke Metz et al. [1] proposed a study on the usage of DCGAN for unsupervised learning. In this paper, they were able to show that a deep convolutional adversarial pair learns a hierarchy of representations from object parts to scenes in both the generator and discriminator.

Tero Karras et al. [2] proposed an alternative generator architecture for generative adversarial neural networks, where the use of a new generator improved the state-of-the-art in terms of traditional distribution quality metrics. Two new automated methods were also employed to quantify interpolation quality and entanglement of the latent factors of variation.

Qin et al. [3] proposed an RBSM (relative symmetric bi-linear model) to verify kinship when the information about both parents are available. Datasets for our project are acquired from this work

Robert Gordan et al. [4] proposed a much closer solution to the domain of kinship by applying deep conditional generative adversarial networks to generate child faces using images of parents. While the approach looks promising, qualitative results seem to be poor. Additionally, their report does not provide quantitative metrics to measure the performance of the model. Much of our work in the following sections are concentrated on improving this model to meet the objective.

2 Data

The data used for this work is the Tri-subject Kinship Face Database (TSKinFace) [3]. Each group in the database consists of a family triple of a father, a mother and a child. All these images are harvested from public photo sharing platforms like flickr.com. The images are of the size 64 x 64 and the groups are of diverse races. There are a total of 1015 groups of families in the database.

CMPT726 - Machine Learning (Fall 2019), Simon Fraser University, Canada.

3 Approach

The current model [4] accepts images of parents as input and then generates the image of a child using deep convolutional generative network (DCGAN). Each input image is of size 64 X 64. Strided convolutions are applied to input images (i.e. parents) allowing network to learn its own down-sampling. Input noise distribution Z is a fully connected layer as it is just a matrix multiplication.

To learn the generator's distribution p_g over data x, a prior on input noise variables $p_z(z)$ is defined thus representing mapping to data space as $G(z; \theta_g)$ where G is a differentiable function with parameters θ_g [5]. The parameters we use in this model are individual image of each parent. Hence reframing the general equation of generator $G(z)$ shall look as below:

$$G(Z, f_1(p_1), f_2(p_2))$$

where p_1 and p_2 represent each parent and Z is a random noise applied on to the space of input data x. The generated image are then passed through a discriminator $D(x; \theta_d)$ to produce a single scalar boolean value that indicates whether the image produced is fake or real. The discriminator function can be expressed as:

$$D(x, f_3(p_1), f_4(p_2))$$

We then train D to maximize the probability of assigning true labels to both ground truth and generated images from G. Simultaneously we also train G to minimize $log(1-D(G(Z)))$. Hence the loss function applied as per the original GAN implementation is as below:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data(x)}}[logD(x)] + \mathbb{E}_{z \sim p_{data(z)}}[log(1-D(G(Z)))]$$

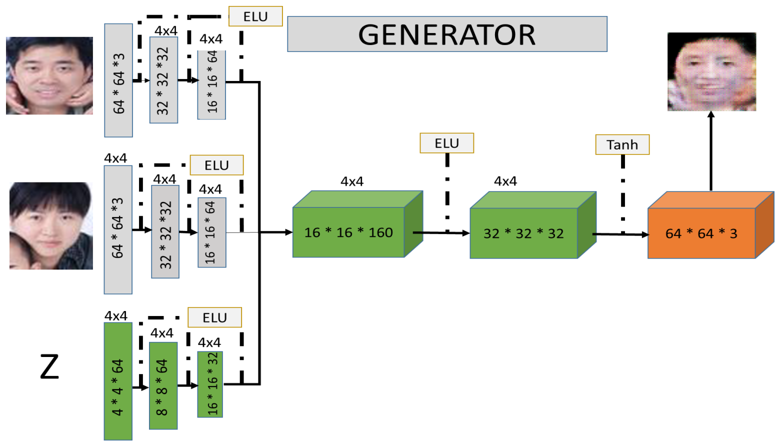

Figure 1: Architectural overview of generator in baseline model. Grey boxes refer to convolution layers and green boxes refer to de-convolution layers. Orange box depicts the output. Batch Normalization is applied between each layer and Dropout is applied to the first layer.

Our approach to improvise the model is four fold. We continue to use the same parameters, optimizers (Adam's optimizer) and loss function for discriminator and generator. However we felt the need of

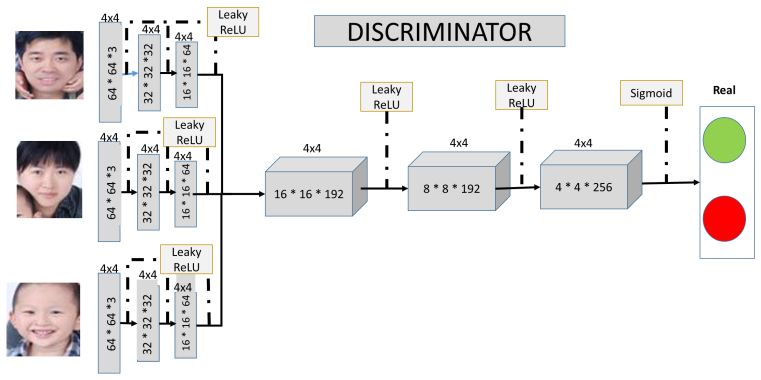

Figure 2: Architectural overview of discriminator in baseline model. Batch Normalization is applied between each layer.

modifying the convolutional layers to perform a deeper learning of pixels. Moreover,recent studies by Pedamonti et al. [5] and Nwankpa et al. [6] cites various flavours of ReLU and new activation functions in deep learning research. These papers motivated us to modify the activation functions that may substantially improve the learning. We also felt the need to experiment various weight initialisation techniques to prevent layer activation outputs from exploding. Finally we have also tried the traditional CNN model with same architecture to perform qualitative checks.

4 Experiments

The TSKin dataset has a total of 1015 families consisting of images of father, mother, and child. We made use of data augmentation techniques such as horizontal flipping and zooming to triple the size of our dataset. We use 80% of this data for training and reserve the rest for testing the model.

4.1 Evaluation metrics

Qualitatively, we evaluate the performance of the model by simply visualizing the predicted image and comparing it to the ground-truth child image. Quantitatively, we measure the similarity of the predicted image to ground-truth image using metrics such as Structural Similarity (SSIM) and Peak Signal to Noise Ratio (PSNR), which are the popular metrics used to quantify the similarity of images [9]. Higher the value of the metric, more is the similarity between the images for both the metrics.

The following modifications in the model are evaluated for 100 epochs due to limited resource power. The values for SSIM and PSNR are averaged over all the images in the testing set.

4.2 Layer Modifications

From the baseline model, we keep the noise vector unchanged. We tried making changes to the convolution layers of inputs in Generator and Discriminator to see if that has an impact on the efficiency of the model. The baseline model consisted of 2 convolution layers with a filter size of 4, a stride of 2 and a padding size of 1. The values for the evaluation metrics obtained are shown in table 1. Now, for the next model (model 2) architecture setup, we decided to reduce the kernel size of the convolution layers to 2 so that a detailed feature extraction was possible from the parents' images. The number of layers and stride remained unchanged from the previous architecture. To make the output of this CNN network to be 16x16x3, we set the padding of this architecture to be 0. The generated child image quality was low, which is why we moved for another setup (model 3) in which we decided to do in-depth feature extraction from each pixel by reducing the stride to 1. To accommodate this change, we made the filter size to 5 and kept padding size unchanged. As a result, the model now consists of 12 CNN layers to extract features from the parent images.

Inferring from the output images, we decided to experiment a model architecture (Model 4 in table 2) which is a mixture of above models. The input image of 64x64 was fed into the first stage of this model that extracts features from a higher level. This stage has a single convolution layer with a filter

Models	SSIM	PSNR
Baseline Model	0.1187	3.5450
Model 2	0.1020	3.8876
Model 3	0.0933	3.9813
Model 4	0.1154	3.5634

Table 1: Evaluation Metrics for different layer architectures

size 4, a stride of 2 and a padding of 1. The output of this layer was a 32x32. This 32x32 image was fed into second stage which consists of 5 convolution layers each with a filter size of 4, a stride of 1 and with no padding. This was then fed to the final stage to produce 16x16 images of parents which contains a single convolution layer of filter size 2, a stride of 2 and a padding of 1.

4.3 Activation functions

Activation functions used by baseline model are inspired from the work done by Radford et al.[4] with only exception to the usage of ELU (Exponential Linear Unit) in the hidden layers of generator. It was highlighted by Nwankpa et al [6] about the critical limitation of ELU where it does not center the values at zero. In-order to avoid this problem, we thought of applying two extensions to ELU commonly called as Continuous Differentiable Exponential Linear Unit (CELU) [7] and Scaled Exponential Linear Unit (SELU) [8]
Continuous Differentiable Exponential Linear Unit (CELU) proposed by Barron et al. had useful properties compared to ELU. This method introduces an alternative parameter method which is continuous for all values of α, making the rectifier easier to reason about and making α easier to tune. The formula looks as follows:

$$f(x) = \begin{cases} \text{x}, & if \quad x \geq 0 \\ \alpha(e^{x/\alpha} - 1), & \text{otherwise} \end{cases}$$

Scaled Exponential Linear Unit (SELU) is another variant of ELU proposed by Klambauer et al. This function has a peculiar property of inducing self-normalizing properties. The formula is as follows:

$$f(x) = \tau \begin{cases} \text{x}, & if \quad x \geq 0 \\ \alpha e^{x} - \alpha, & \text{otherwise} \end{cases}$$

where τ is the scale factor.

Apart from the above two important functions, we also tried with multiple variants of ReLU to evaluate the performance of generator. Evaluation metrics obtained from these experiments are mentioned below.

Activation Function	SSIM	PSNR
ELU	0.1187	3.5450
ReLU	0.1459	3.9669
LeakyReLU	0.1018	4.0116
CELU	0.1165	4.9286
SELU	0.1519	3.5337

Table 2: Performance comparison of different activation functions used in generator

We found that SELU did produce better results among all. The output of the hidden layer in generator is still passed through Tanh with no change. As far as discriminator is concerned, we still continue to use Leaky ReLU in hidden layer followed by Sigmoid at output because our modifications did not help in any improvements.

4.4 Weight Initializations

The sensitivity of the generated images on the weight initialization motivated us to experiment with several weight initializations. The original GAN model for child-face prediction by Gordan et al.

used a gaussian normal distribution for initial weights of the model. In our experiment we make use of Xavier/Glorot [10], Kaiming/He [11] initializations with uniform and normal distributions, and compare their effect on the performance of our GAN model. The following table summarizes the model performance for 100 epochs for different weight initializations.

Weight Initialization	SSIM	PSNR
Gaussian Normal	0.1187	3.5450
Xavier Normal	0.1259	5.6667
Xavier Uniform	0.1158	5.6314
Kaiming Normal	0.1277	5.0184
Kaiming Uniform	0.0919	4.8534

Table 3: Performance comparison of different weight initializations

4.5 CNN Implementation

We implemented a supervised CNN model in parallel to the experiments done with layer modifications. This step was done to recheck the quality of images produced from both DCGAN and CNN based approach. To measure the loss we used RMSE with respect to ground-truth child image.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i\in I}(y_i - t_i)^2}$$

5 Proposed Model

The changes in the previous experiments that gave the best results were put together to form the final model. We, therefore, change the architecture to model 4, activation functions to SELU, weight initializations to Xavier Normal as shown in the below figure.

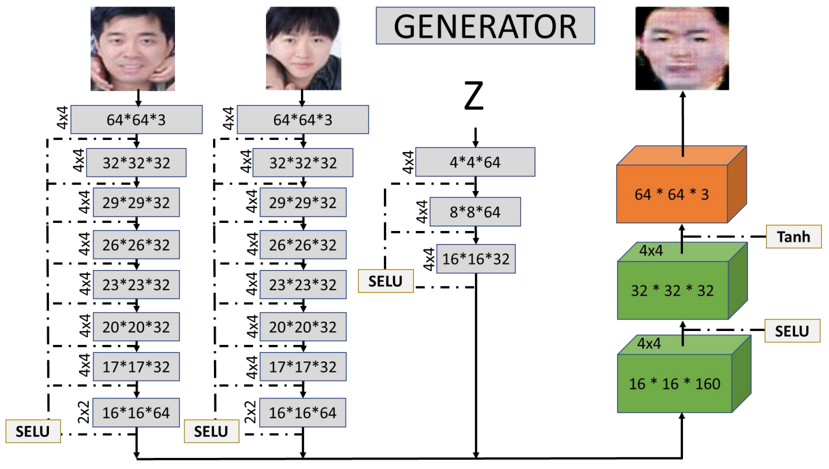

Figure 3: Architectural overview of generator in proposed model

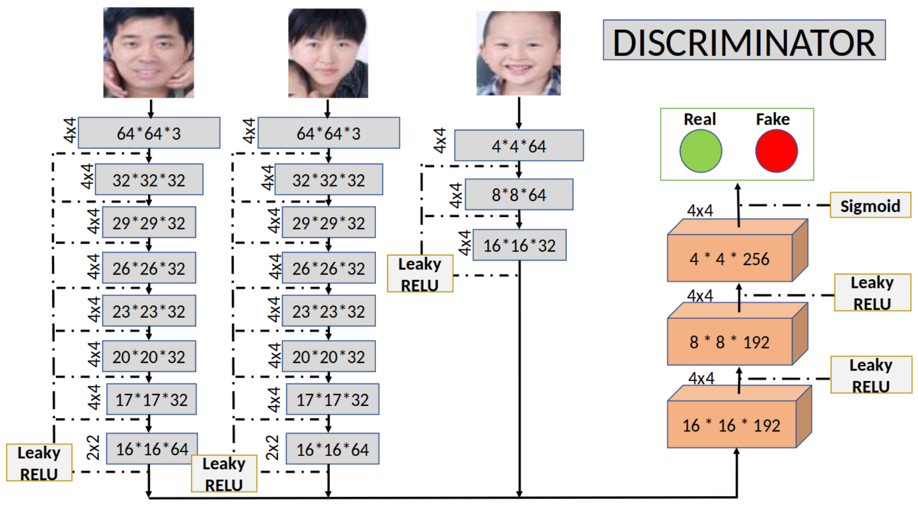

Figure 4: Architectural overview of discriminator in proposed model

6 Results and Conclusions

After training the model for sufficiently long time ($\sim$ 1000 epochs) we get the following results. The proposed model is able to generate better child-images that are close to the ground-truth image and have higher values for both the similarity metrics. The CNN model, trained for the same number of epochs, however, generates blurry images and consequently gives low scores for both the quantitative metrics.

Clearly, from the quality of generated images and the scores on the evaluation metrics, we can say that there is a good scope for further improvement of the model, at least when we compare to the state-of-the-art face-generation GAN models. The major step in the direction of improvement would be the acquisition of a larger and more consistent dataset.

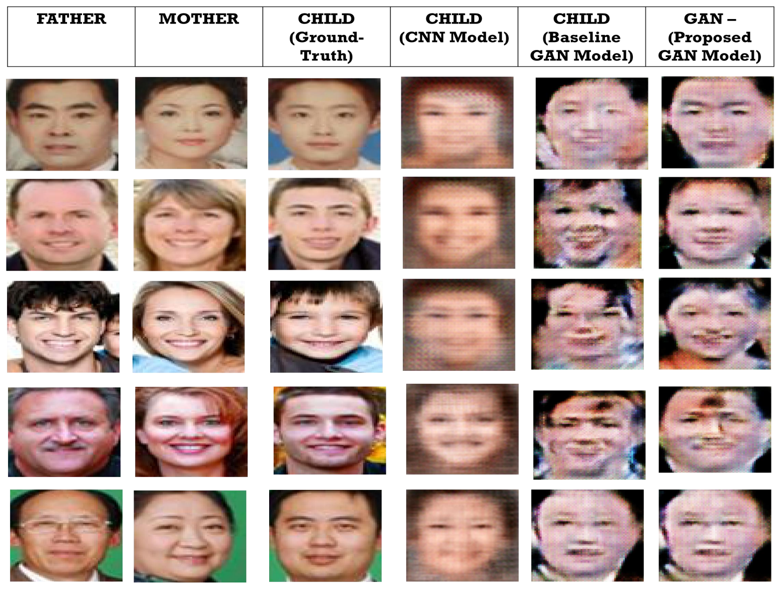

Figure 5: Comparison of qualitative results of CNN, baseline, and proposed model

Model	SSIM	PSNR
CNN	0.1143	3.005
Baseline GAN model	0.1452	5.996
Proposed GAN model	0.1715	6.152

Table 4: Quantitative comparison of different models

7 Contributions

All the members have contributed equally to the project. The breakdown, upon your request, is as below.

Task	Member
Ideation and Related work	Vipin Das and Kenny Tony Chakola
Data Extraction and Augmentation	Mayur Mallya and Kunal Niranjan Desai
Experimentation with Layer Changes	Kenny Tony Chakola and Vipin Das
Optimizing DCGAN using Activation Functions	Vipin Das and Kunal Niranjan Desai
Optimizing using Weight Initialization	Mayur Mallya and Kenny Tony Chakola
Incorporation of Evaluation Metrics	Kunal Niranjan Desai and Mayur Mallya
Poster and Report	All four

8 Acknowledgements

We thank Prof. Mori and his TAs for this course, the project presentation event, and their valuable time and advice. We also thank CSIL SFU for the resources.

References

[1] Radford A, Metz L, Chintala S. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434. 2015 Nov 19.

[2] Karras T, Laine S, Aila T. "A style-based generator architecture for generative adversarial networks." InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019 (pp. 4401-4410).

[3] Qin, Xiaoqian, Tan, Xiaoyang, and Chen, Songcan. "Tri-subject kinship verification: Understanding the core of a family." IEEE Transactions on Multimedia, 17(10):1855–1867, 2015.

[4] Robert Gordan, Mingu Kim, Alexander Muñoz. "Child Face Generation with Deep Conditional Generative Adversarial Networks." https://github.com/munozalexander/Child-Face-Generation/blob/master/writeup.pdf

[5] Dabal Pedamonti."Comparison of non-linear activation functions for deep neural networks on MNIST classication task." arXiv:1804.02763v1 [cs.LG] 8 Apr 2018

[6] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshal."Activation Functions: Comparison of Trends inPractice and Research for Deep Learning".arXiv:1811.03378v1 [cs.LG] 8 Nov 2018

[7] Jonathan T. Barron."Continuously Differentiable Exponential Linear Units".arXiv:1704.07483v1 [cs.LG] 24 Apr 2017

[8] Günter Klambauer,Thomas Unterthiner and Andreas Mayr.Self-Normalizing Neural Networks.arXiv:1706.02515v5 [cs.LG] 7 Sep 2017

[9] Søgaard, Jacob; Krasula, Lukáš; Shahid, Muhammad; Temel, Dogancan; Brunnström, Kjell; Razaak, Manzoor (2016-02-14). "Applicability of Existing Objective Metrics of Perceptual Quality for Adaptive Video Streaming" (PDF). Electronic Imaging. 2016 (13): 1–7. doi:10.2352/issn.2470-1173.2016.13.iqsp-206.

[10] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks." In AISTATS, 2010.

[11] Kaiming He,Xiangyu Zhang,Shaoqing Ren and Jian Sun."Delving Deep into Rectifiers:Surpassing Human-Level Performance on ImageNet Classification".arXiv:1502.01852v1 [cs.CV] 6 Feb 2015