

**A Project Report On**  
**DETECTION AND CLASSIFICATION OF GAN GENERATED FAKE**  
**FACES**

*Submitted in partial fulfilment of the  
requirements for the award of the degree  
Of*

**BACHELOR OF TECHNOLOGY**  
*In*  
**ELECTRONICS AND COMMUNICATION ENGINEERING**

Submitted by:  
**Shubhanshu Agarwal**  
(16118078)

Under the guidance of:  
**Dr. VINOD PANKAJAKSHAN**



DEPARTMENT OF ELECTONICS AND COMMUNICATION ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE – 247667 (INDIA)  
June, 2020

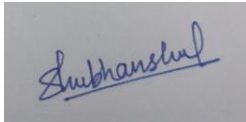
## CANDIDATE'S DECLARATION

I hereby certify that the work presented in this report entitled "DETECTION AND CLASSIFICATION OF GAN GENERATED FAKE FACES" in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology is carried out by me during the period of July 2019 to June 2020 under the supervision of Dr. **VINOD PANKAJAKSHAN**.

This work has not been submitted elsewhere for the award of a degree/diploma/certificate.

Date: May30,2020

Place: Roorkee



Shubhanshu Agarwal  
(16118078)

.....

Name of Student

# Detection and Classification of GAN Generated Fake Faces

Shubhanshu Agarwal  
Department of ECE  
Indian Institute of Technology  
Roorkee, India  
sagarwal@ec.iitr.ac.in

**Abstract**—Current development in the domain of computer vision, image, processing, and deep learning allows us to generate hyper-realistic images that are hardly distinguishable from the real ones. In particular, the generation of fake faces with the help of some deep learning architecture is increasing day by day. However, at the same time, these deep learning architecture are maliciously used. The so-called real faces become the source of manipulation and forgery in the real world. Therefore, it is essential to develop its counterpart so that it can be used to distinguish the fake images from the real ones. In the last few years, several models are proposed that are used for this purpose, but with evolution, in the fake images, the same model cannot be used to give the excellent result as given earlier. Hence, there is a need for a generalized model which can be used in each condition or can be modified with the minimum number of changes.

**Index Terms**—Generative Adversarial Networks, Image Forgery, Fake Faces

## I. INTRODUCTION

With the advances of Machine Learning, Deep Learning, and Artificial Intelligence, it is tremendously easy to create hyper-realistic images and videos. In particular, nowadays, thanks to Generative Adversarial Networks [1], which are the backbone of these hyper-realistic images. With the advancement in GANs, the images are becoming real, which can not be distinguished by human eyes.

GANs were introduced by Ian Goodfellow et al. in 2014 described adversarial as one of the coolest things.. They are the neural networks that generate synthetic data given specific Input data. GANs consists of two networks models: Generative and Discriminative Model

### A. The Generative Model

This models take some random noise as input and tries to produce real input training images by fooling the discriminator model.

### B. The Discriminative Model

This model operates like a standard binary classifier that can classify images into different categories. It determines whether the generated image is coming from the given dataset or the generative model artificially generates it.

Figure 1 shows the general architecture and overview of the GANs

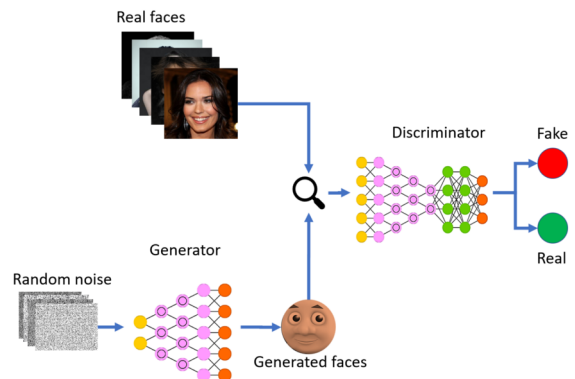


Fig. 1. An overview of the architectures of GANs

## II. TYPES OF GANs

### A. Progressive GAN(ProGAN)

Progressive GAN [2] also known as ProGAN is an extension to the generative Adversarial network training process that allows the stable training of generator models that can output large high-quality images compared to traditional GANs. It is an incremental process that involves the starting with a small image and progressively adding blocks of layers that increase the output size of the generator model and input size if the discriminator model until the required size is achieved. The problem with the traditional GANs is that they are limited to small dataset sizes and images of a few hundred pixels. This approach is instrumental in generating high-quality synthetic faces that are remarkably realistic.

### B. StyleGAN

There are many kinds of research going on around GANs to improve the quality, reducing model size and tuning hyper parameters, but one of the remarkable research results is StyleGAN [3], which was proposed in 2018. It is the extension of Progressive GAN. Roughly speaking, the StyleGAN

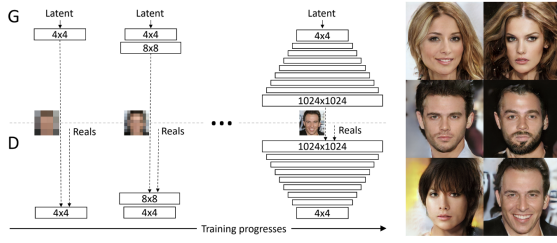


Fig. 2. Progressive growing of GAN architecture

network consists of two features: generating high-resolution images using ProGAN and incorporating image styles into each layer using AdaIN [4]. AdaIN is an abbreviation for Adaptive Instance Normalization, a normalized method for style transfer proposed in 2017.

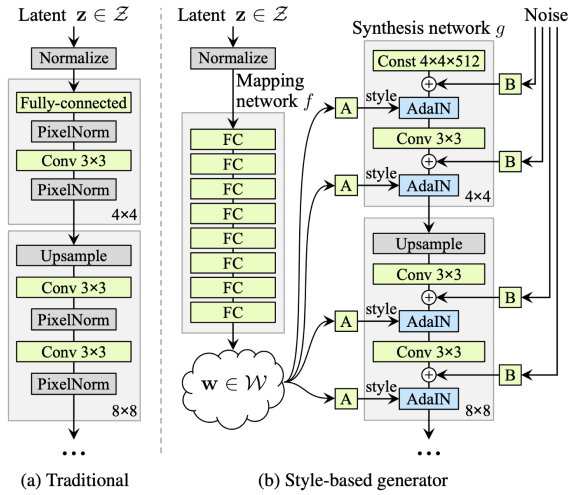


Fig. 3. Style based Generator architecture

### C. StyleGAN2

This architecture further enhances the quality of generated images as compared to the previous one. StyleGAN2 [5] was proposed in December 2019. The main purpose of this paper is to remove the artifacts from the generated images. It improves image generation's quality using the characteristics of Perceptual Path length (PPL) and the droplet noise and non-following modes. The key difference from the StyleGAN model is that it eliminates droplet modes by normalizing with estimated statistics instead of normalizing with actual statistics like AdaIN [4]. It uses a hierarchical generator with a skip connection to reduce eyes and tooth stagnation. Thus improves image quality by reducing PPL and smoothing latent space.

### III. LITERATURE SURVEY

With the evolution of GANs, there have been several hundreds of papers on using GANs to generate images. These papers focus on generating high-quality hyper-realistic images [1]–[3], [5]–[8]. On the other hand, the misuse of these

images also started, and hence, there is a need for an approach to detect whether the image is GAN generated or not. Several methods have already been proposed in the architecture to detect them. Some of them exploit facial artifacts, like asymmetries in teeth, colors of eyes, artifacts arising from the faces' underlying geometry, especially area near the nose, ears, the border of the eyebrows, and face [9]. Color components and information also plays a vital role in exposing these images in [10], [11]. In particular, [10] proposes to use features shared by different GAN architecture to transform the feature map into RGB images. In [12], the author suggests that the face landmarks are different in real and fake images and hence, uses them to distinguish it from the real images.

The presence of specific GAN artifacts is the one way to expose them and suggest that the generated image may be characterized by specific fingerprints just like a natural image. These artificial fingerprints are studied in [13], [14], they show that different fingerprints can characterize each GAN, and these fingerprints can be used in image forensics.

Several other methods rely on deep learning networks to detect these hyper-realistic methods. [15] proposed a method to use the co-occurrence matrix to classify the GAN generated images. Other papers proposed so far include [16], [17], [18] showing a very good accuracy in detecting GAN generated images, even after compression.

### IV. PROBLEM WITH EXISTING SOLUTION

Many architectures have been proposed and used to detect and classify the images and working well also. Nevertheless, the main problem with them is that they do binary classification i.e., they classify the images as real or fake only. They cannot be used when we need to know the class of GANs the image belongs to. The other problem is that the new GAN architecture for generating images is proposed by the day, and these generated images cannot be classified with the existing solution. They require either detector to be retrained on the larger dataset or fine-tuned on them. The first solution requires a larger dataset and thus not feasible as the architecture is increasing day by day while the latter does not provide a satisfactory result. A simple fine-tuning can destroy the previous information required to classify, known as catastrophic forgetting [19].

### V. DATASET USED

The dataset is built using three different well known state-of-the-art generative architecture, as described below. All of them consist of several images of 256x256 and 1024x1024 high-quality images of different objects. For the real images, we have used FFHQ Dataset.

#### A. FFHQ Dataset

Flickr-Faces-HQ (FFHQ) is a high-quality image dataset of human faces, created initially as a benchmark for GAN. The dataset consists of 70000 high-quality PNG images at 1024x1024 resolution and contains considerable variation in terms of age, ethnicity, and image background. It also has



Fig. 4. Representative Images from our GAN Dataset

good coverage of accessories such as sunglasses, eyeglasses, hats, etc. Currently, we are using a total of 1001 images.

#### B. ProGAN Dataset

This dataset consists of the facial images of size 1024x1024 with center cropped generated using the correct code available with the paper[2]. The total number of images currently used for training consists of 1064 images.

#### C. StyleGAN Dataset

The size of this dataset is 1.49 GB consists of facial images of size 1024x1024 with center cropped generated using correct code available with the paper[3]. The total number of images used for training consists of 1000 images.

#### D. StyleGAN2 Dataset

This dataset is generated by the official source code available with the paper[5]. The size of the dataset is 1.26 GB of size 1024x1024 center cropped. The total number of

images used here is 1001 images.

Sample images are shown in Fig.4

TABLE I  
DATASET USED IN OUR EXPERIMENTS

Dataset	Size	No. of Images	Resolution
<b>FFHQ</b>	1.26 GB	1001	1024X1024
<b>Progressive GAN</b>	1.04 GB	1064	1024X1024
<b>StyleGAN</b>	1.49 GB	1000	1024X1024
<b>StyleGAN2</b>	1.26 GB	1000	1024X1024

## VI. PROPOSED METHOD

Several existing models can do detection and classification. However, none of them used the noise residual to classify the image. This is the change that we have proposed to the existing architecture. We used different existing pre-trained architecture





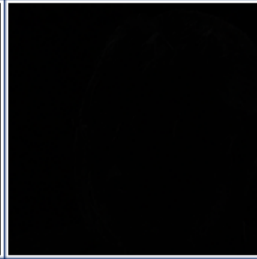

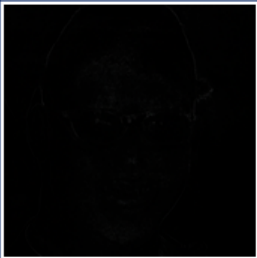


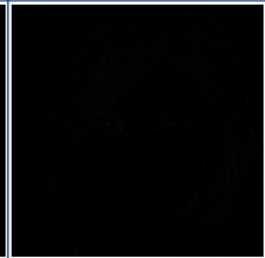
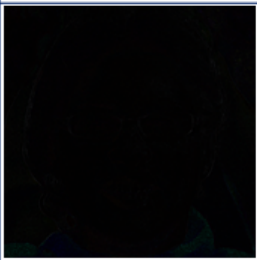

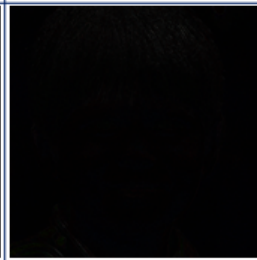
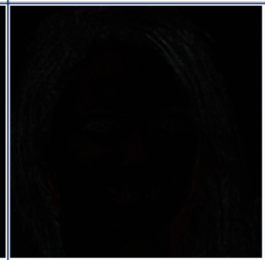
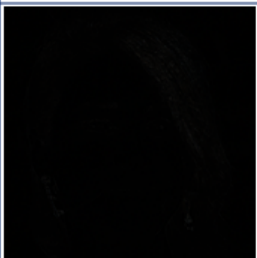
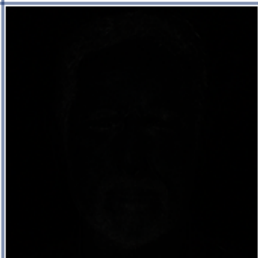


Real				
Progressive GAN(PROGAN)				
Style GAN 1				
Style GAN2				

Fig. 5. Corresponding Noise Feature Map of Images from our GAN Dataset

with the change in the final linear layer and the input image used for classifying.

#### A. Noise Stream Generation

RGB channels are not able to tackle all cases of classification. With the evolution of the GAN, the RGB channels going towards the real images, but since these hyper-realistic images have input elements as noise, so we utilize the local noise distribution to extract these features. This method is novel, and no one used till now to classify the image. Although this method is prevalent in image forensics like in the field of image manipulation techniques [20] and yields excellent results there.

As mentioned earlier, deep learning models use RGB channels as input, but none of them uses the noise stream as an input. Inspired by the progress and advancement of Steganalysis Rich Model(SRM) Features in the image forensics [21], we use these SRM features to extract local noise features. [21]

proposed various SRM Filter kernels that can be used to generate local noise features, but [20] shows that with the help of only three filters, we can achieve decent performance and these filters are shown in Fig. 6. The noise feature Map using these three kernels is calculated for all the images present in the dataset and can be used for the training directly. Noise feature map of the images present in Fig.4 is shown in Fig.5.

$$\frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \frac{1}{12} \begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \quad \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Fig. 6. SRM Filters kernels used to extract Noise Feature Map

## B. Model Architecture

After generating the image and building the dataset, we used three network architecture that performs well in binary classification and tries to compare the results with RGB channel and noise feature map as input, respectively. The network architecture we have are as follows:

### A. VGG19

In 2014, a couple of architecture was more significantly different and made another jump in performance. VGG is an architecture layer with a pair of convolution layers, a poolings layer, and a fully connected layer. It has two variants: VGG 16 and VGG19, having 16 and 19 layers, respectively. So this is quite a costly computation with 138M total parameter, and each image has a memory of 96MB, which is so much large than a regular image.

### B. ResNet

The main base element of ResNet is the residual block, which helps in increasing accuracy. The computation becomes complex as we go deeper into the network. One of the main reasons for using ResNet because of the number of FLOPs(Floating Point Operations per second). Additionally, with Resnet, the deeper model with residual network helps capture a large amount of variance within the data.

### C. Densenet

DenseNet is composed of Dense blocks. Within those blocks, the layers are densely composed and connected. Each layer gets its input from the previous layers output feature maps. This extreme reuse of the residuals creates deep supervision because every layer receives more supervision from the previous layer, making it more powerful. This method gives an advantage, i.e., a shorter connection between layers close to input and output means we stick with the theme of dense network with less vanishing gradients

We use these three networks, performing six experiments. Out of which three has given input as RGB channel image and used the as current-state-of-art and other three using noise feature map as input to the model architecture as our proposed model.

## VII. RESULT

We calculated train, test, and validation dataset accuracy for each of the experiments.

These values are listed in Table II. The loss curves for training and validation, the duration for each iteration, accuracy after each epoch are also attached with the report<sup>1</sup>. (see APPENDIX)

Apart from that, these two major observations have been made on seeing these curves:

- 1 In each case, the training and validation loss curve, converge faster when Noise feature map as compared to RGB Channel as an input

- 2 The model attains saturation earlier in noise stream input with less number of the epoch as compared to RGB channels as an input.

TABLE II  
TRAINING, TESTING AND VALIDATION DATASET ACCURACY OF  
DIFFERENT MODEL ARCHITECTURE

Model Architecture	Training Set	Validation Set	Test Set
VGG19	100	98.4	98.59
Resnet	97.319	99.2	98.39
Densenet	99.4	98.5	99.4
VGG_SRM	99.34	98.4	97.38
Resnet_SRM	98.7	99.8	99.168
Densenet_SRM	99.15	99.8	99.4

## VIII. CONCLUSION

This work aims to demonstrate that a particular type of GAN exhibits a different and unique pattern that can be used to distinguish the GAN generated hyper-realistic images from the real ones with the existing architecture just by modifying the input to that model. We believe sufficient experiments and results support the facts. There are lots of question arise by this experiment, such as how the noise residual depends on the network, both its architecture and its specific parameters.

## IX. DISCUSSION AND FUTURE WORK

There are abrupt spikes in all the loss curve, and the reason behind that might be is random sampling. We will try to reduce them. Till now, we have only tested on the specific type of images only. In the future, we will try to use it on the different types of images simultaneously. The cross-evaluation of the two datasets is how we can evaluate generalizability. We can also test the accuracy of the images on the different JPEG compression.

## ACKNOWLEDGMENT

We want to extend our heartiest thanks to our supervisor Professor Vinod Pankajakshan, Department of Electronics and Communication Engineering, Indian Institute of Technology, Roorkee, for providing us an opportunity to work on this challenging project. His support and continued motivation helped us to gain insight into various aspects of research and development. We would never have been able to complete our work without our supervisors' guidance, help from friends, and support from our families and loved ones. Last but not least, we thank God/Nature from the inner part of the soul who made all the things possible.

## REFERENCES

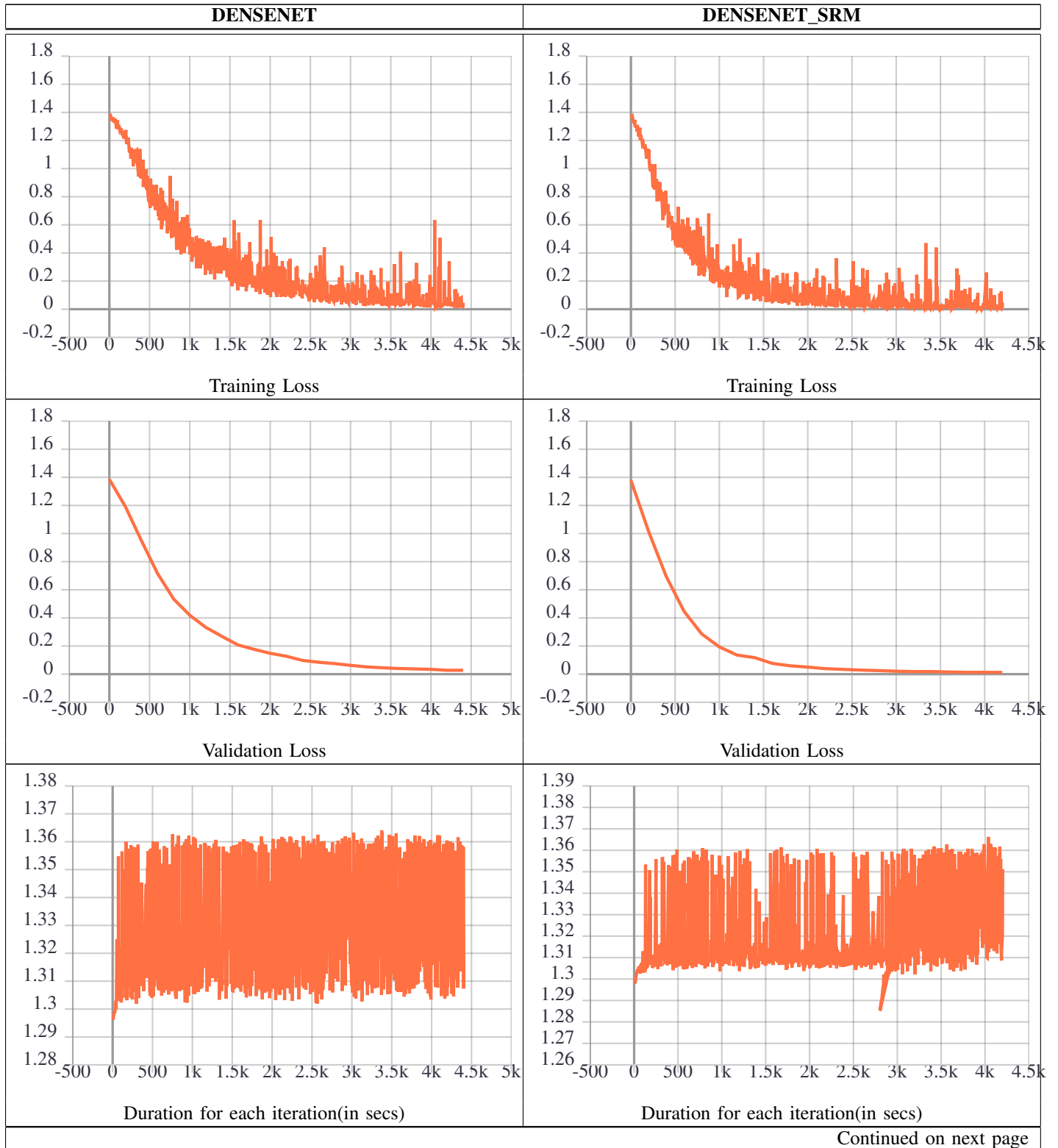
- [1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *CoRR*, vol. abs/1710.07035, 2017. arXiv: 1710.07035. [Online]. Available: <http://arxiv.org/abs/1710.07035>.

<sup>1</sup>Source code for this project is available on <https://github.com/Shubhanshu07/Classification-of-GAN-Generated-Fake-Faces>

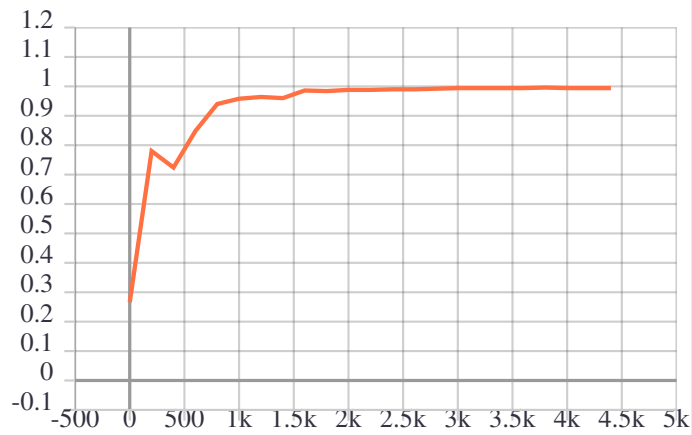
- [2] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *CoRR*, vol. abs/1710.10196, 2017. arXiv: 1710.10196. [Online]. Available: <http://arxiv.org/abs/1710.10196>.
- [3] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, vol. abs/1812.04948, 2018. arXiv: 1812.04948. [Online]. Available: <http://arxiv.org/abs/1812.04948>.
- [4] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. DOI: 10.1109/iccv.2017.167.
- [5] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *ArXiv*, vol. abs/1912.04958, 2019.
- [6] Karras, Tero, Laine, Samuli, Miika, Hellsten, Janne, Lehtinen, Jaakko, Aila, and et al., *Analyzing and improving the image quality of stylegan*, Mar. 2020. [Online]. Available: <https://arxiv.org/abs/1912.04958>.
- [7] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *ArXiv*, vol. abs/1606.03498, 2016.
- [8] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.
- [9] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 83–92.
- [10] S. McCloskey and M. Albright, "Detecting gan-generated imagery using color cues," *CoRR*, vol. abs/1812.08247, 2018. arXiv: 1812.08247. [Online]. Available: <http://arxiv.org/abs/1812.08247>.
- [11] H. Li, B. Li, S. Tan, and J. Huang, "Detection of deep network generated images using disparities in color components," *CoRR*, vol. abs/1808.07276, 2018. arXiv: 1808.07276. [Online]. Available: <http://arxiv.org/abs/1808.07276>.
- [12] X. Yang, Y. Li, H. Qi, and S. Lyu, "Exposing gan-synthesized faces using landmark locations," *CoRR*, vol. abs/1904.00167, 2019. arXiv: 1904.00167. [Online]. Available: <http://arxiv.org/abs/1904.00167>.
- [13] N. Yu, L. Davis, and M. Fritz, "Attributing fake images to gans: Analyzing fingerprints in generated images," *CoRR*, vol. abs/1811.08180, 2018. arXiv: 1811.08180. [Online]. Available: <http://arxiv.org/abs/1811.08180>.
- [14] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do gans leave artificial fingerprints?" *CoRR*, vol. abs/1812.11842, 2018. arXiv: 1812.11842. [Online]. Available: <http://arxiv.org/abs/1812.11842>.
- [15] L. Nataraj, T. M. Mohammed, B. S. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, "Detecting GAN generated fake images using co-occurrence matrices," *CoRR*, vol. abs/1903.06836, 2019. arXiv: 1903.06836. [Online]. Available: <http://arxiv.org/abs/1903.06836>.
- [16] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of gan-generated fake images over social networks," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018, pp. 384–389.
- [17] X. Xuan, B. Peng, J. Dong, and W. Wang, "On the generalization of GAN image forensics," *CoRR*, vol. abs/1902.11153, 2019. arXiv: 1902.11153. [Online]. Available: <http://arxiv.org/abs/1902.11153>.
- [18] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental learning for the detection and classification of gan-generated images," in *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2019, pp. 1–6.
- [19] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *CoRR*, vol. abs/1612.00796, 2016. arXiv: 1612.00796. [Online]. Available: <http://arxiv.org/abs/1612.00796>.
- [20] P. Zhou, X. Han, V. Morariu, and L. Davis, "Learning rich features for image manipulation detection," Jun. 2018, pp. 1053–1061. DOI: 10.1109/CVPR.2018.00116.
- [21] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012. DOI: 10.1109/tifs.2012.2190402.



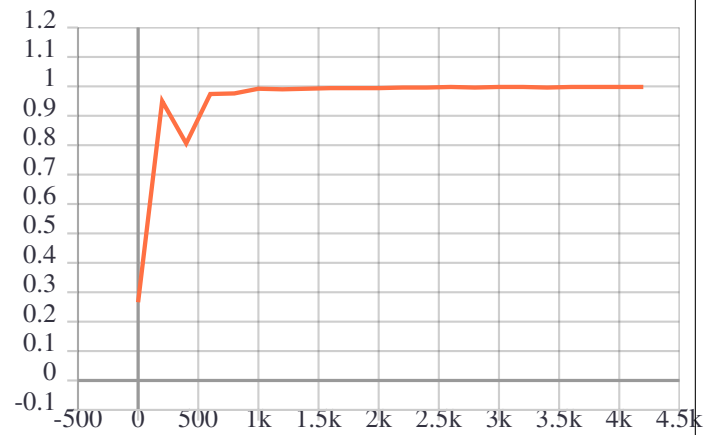
# APPENDIX



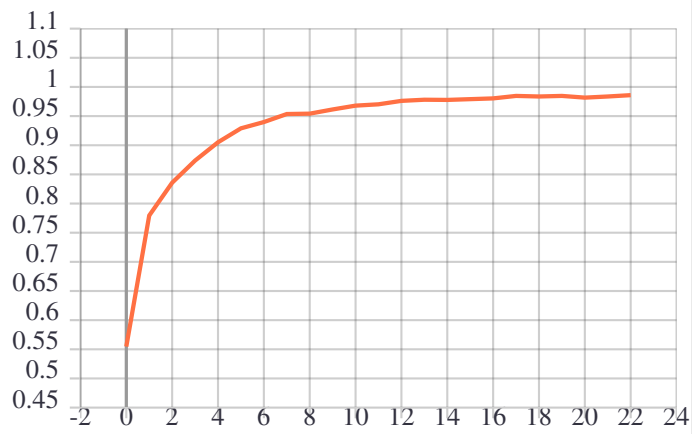
Continued from previous page



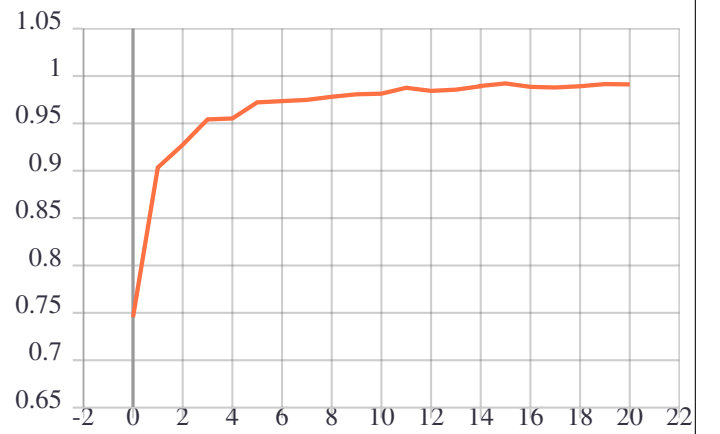
Validation Set Accuracy



Validation Set Accuracy



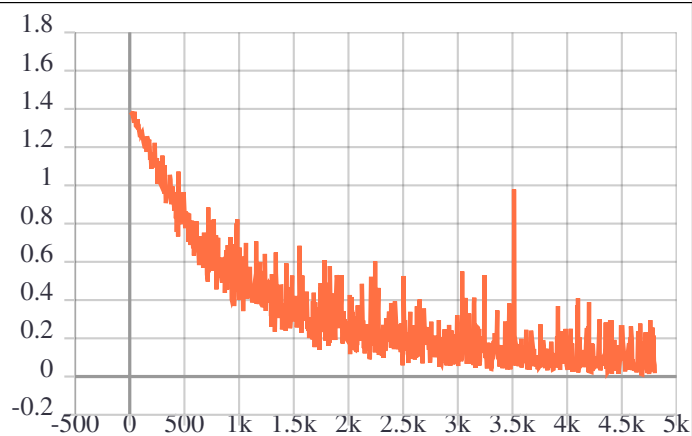
Training Set Accuracy



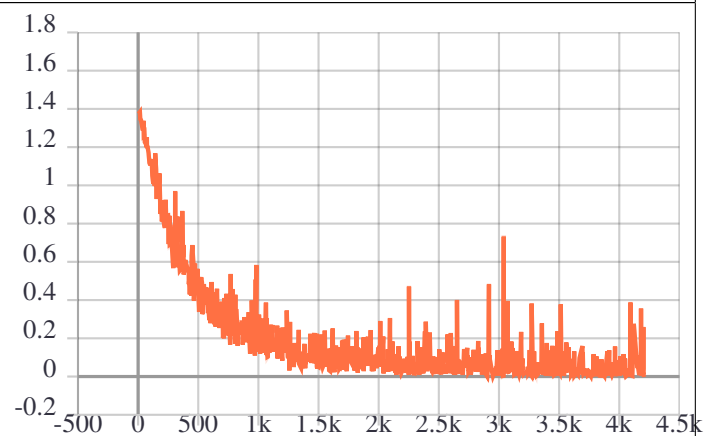
Training Set Accuracy

**RESNET**

**RESNET\_SRM**

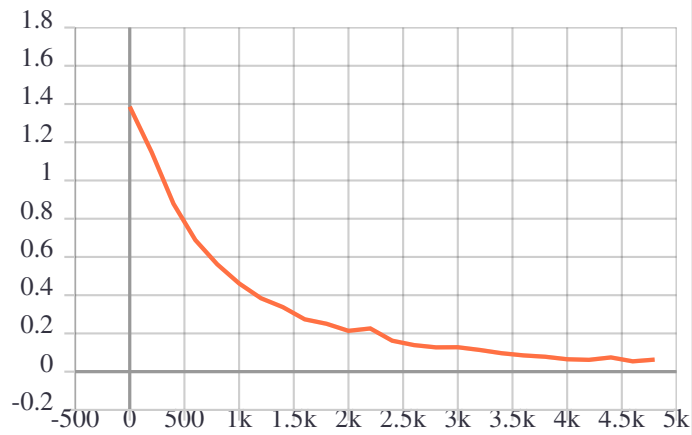


Training Loss

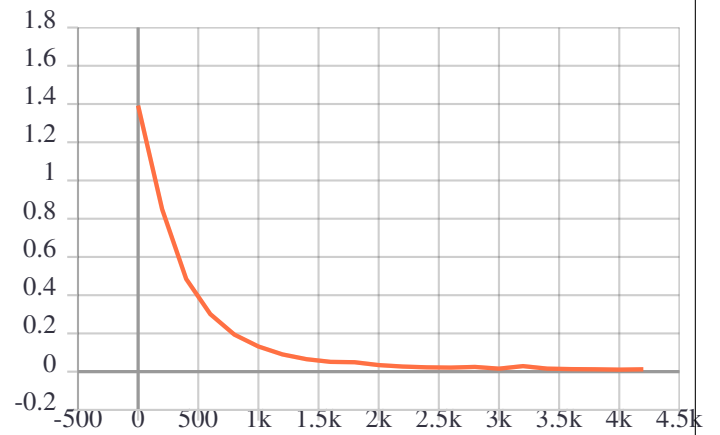


Training Loss

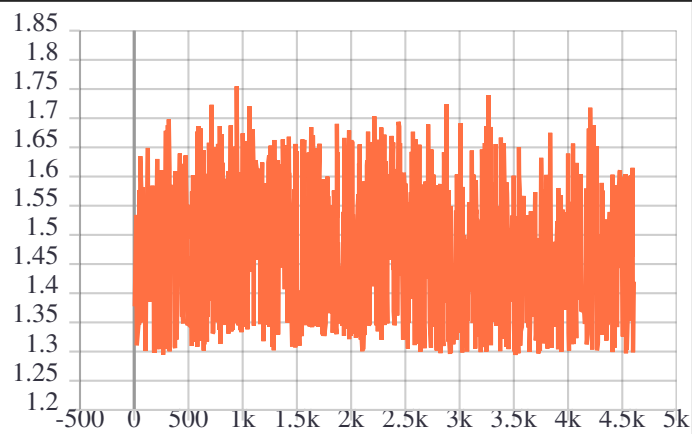
Continued on next page



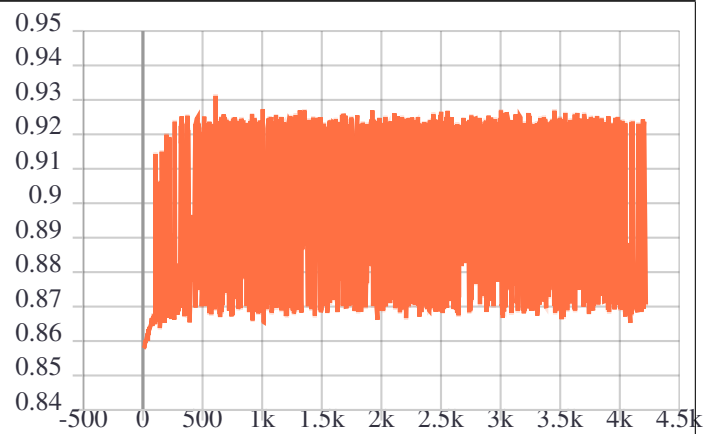
Validation Loss



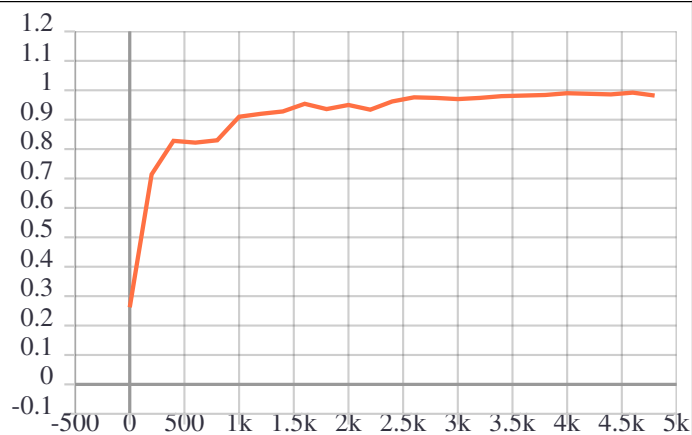
Validation Loss



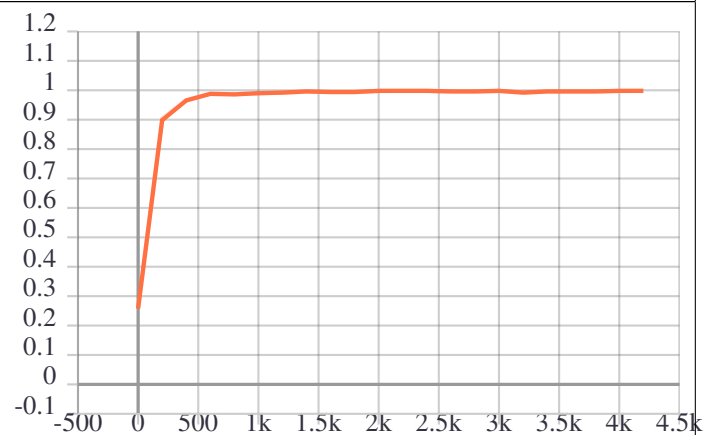
Duration for each iteration(in secs)



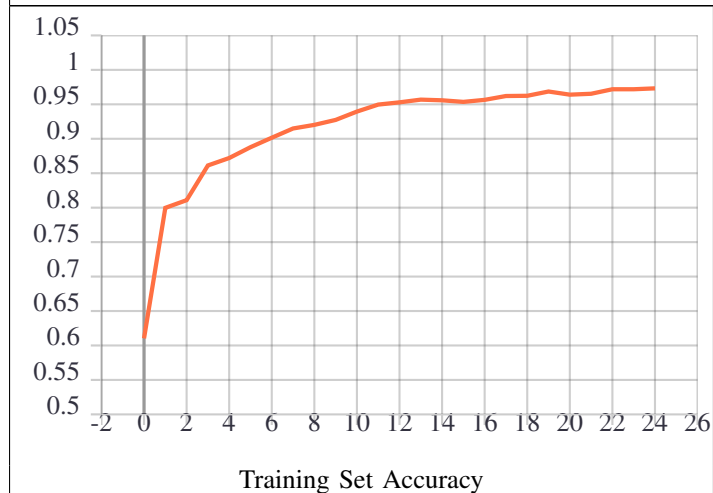
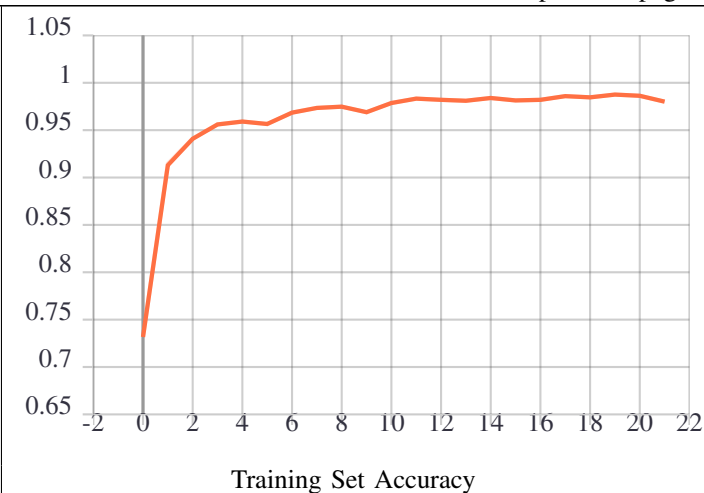
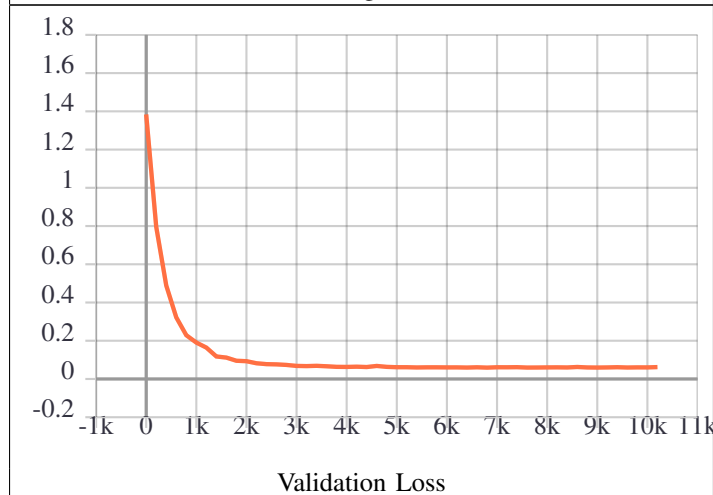
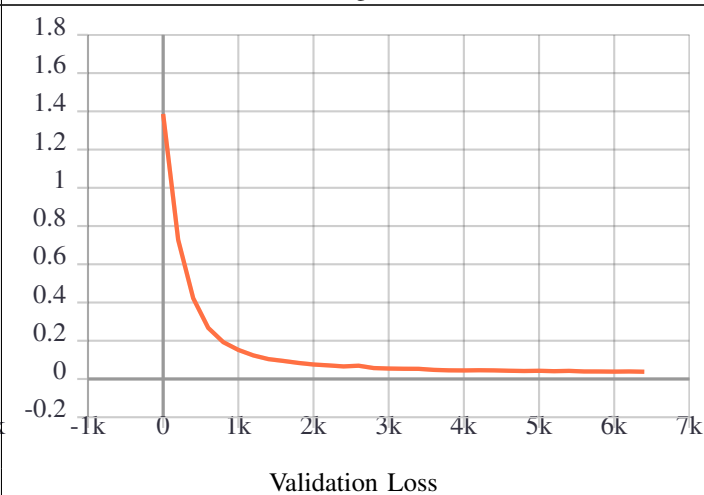
Duration for each iteration(in secs)

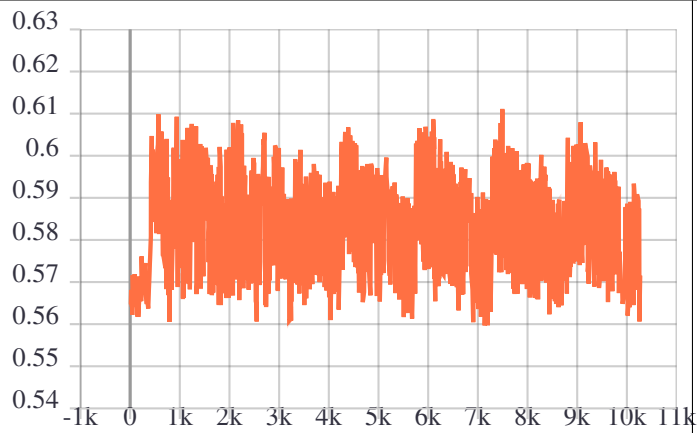


Validation Set Accuracy

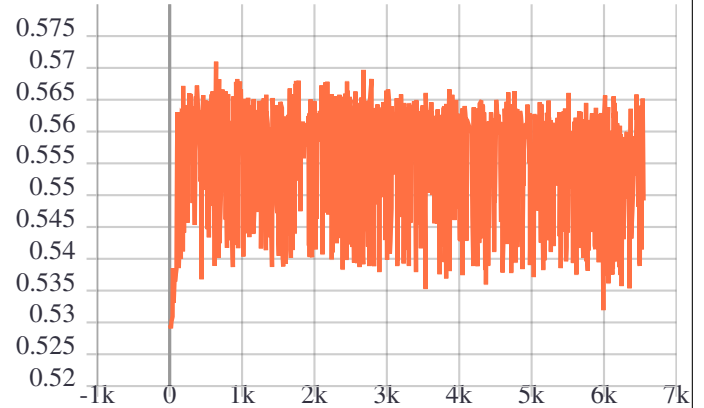


Validation Set Accuracy

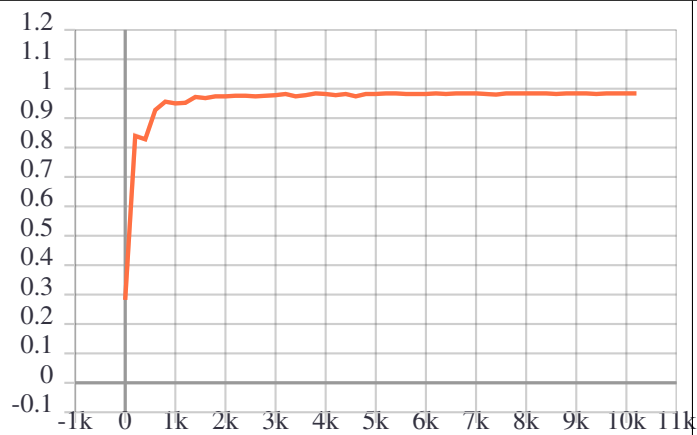
**VGG19****VGG19\_SRM****VGG19****VGG19\_SRM****VGG19****VGG19\_SRM**



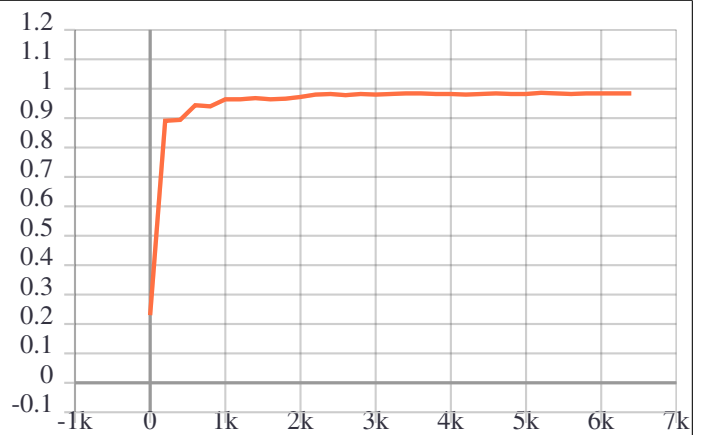
Duration for each iteration(in secs)



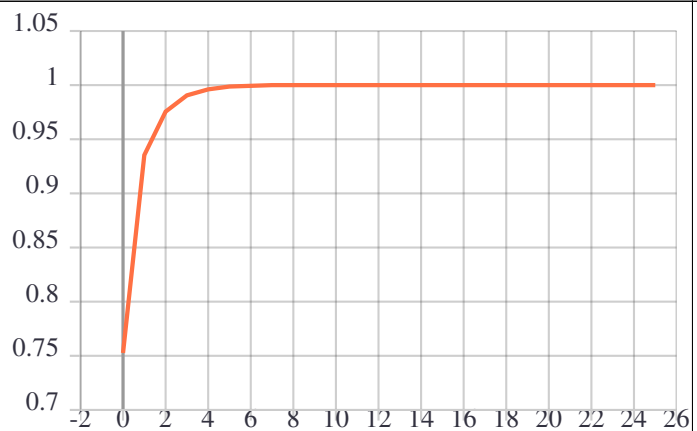
Duration for each iteration(in secs)



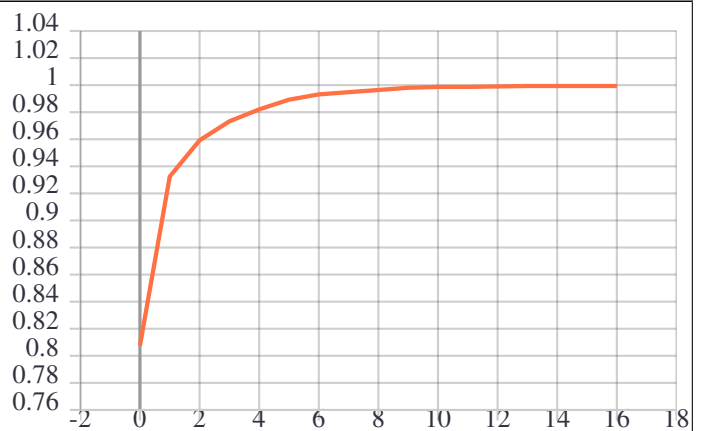
Validation Set Accuracy



Validation Set Accuracy



Training Set Accuracy



Training Set Accuracy