

DeepFakes: Masking and Unmasking Faces using Adversarial Network

Rahul Allam, Abhishek Shivdeo and Gaurav Bhosale

Abstract—Through this paper, we aim to apply deep learning-based image translation techniques to deduce transfer functions that translate images from the source image space to the destination image space and vice versa. Specifically, we apply the above technique to an application of great consequence in the present world we live in: Translation of unmasked faces (face without a face mask) to masked faces (face with face mask), and translation of masked faces to unmasked faces. We exploit the capabilities of generative modeling, which essentially learns the regularities or patterns in input data in a way that the model can be used to generate fake images that plausibly could have been drawn from the original data-set. Specifically, we use Generative Adversarial Networks (GANs), an approach to generative modeling using deep learning methods, and present the fakes generated by our trained models in both image domains so that the reader can inspect the quality of the generated fakes. We further talk about the various experiments and trials performed in terms of tuning hyper-parameters of the network and present the best model for this application based on a GAN numerical evaluation metric. We also plot, present, and discuss the behavior of various GAN losses, and make deductions that aid our understanding of the model. We also talk about the advantages, the disadvantages and the future scope of work that need to be pursued to overcome the limitations of the model. We also present the link to our PyTorch code base that we used to generate the fake images.

I. INTRODUCTION

As the pandemic in 2020 hit us, our lives changed in many ways. One of the significant life changes that we had to adopt was wearing face masks in the external world. Even though face masks shield us from the contagious virus, wearing masks has made it difficult to identify people. For current state-of-the-art face recognition programs, commonly used for security verification purposes, the use of face masks presents an arduous challenge since these programs were typically trained with human faces devoid of masks. But now, due to the onset of the COVID-19 pandemic, the algorithms are forced to identify faces with masks. [21] Using unmasked images, the most accurate algorithms fail to authenticate a person about 0.3% of the time.

Masked images increased even these top algorithms' failure rate to about 5%, while many otherwise competent algorithms failed between 20% to 50% of the time[20]. Removing masks to identify an individual even briefly increases the risk of spreading the virus. Additionally, wearing masks also gives an additional element of disguise to people which has increased the crime rates such as robberies and break-ins [7]. Hence, it's critical for the state-of-the-art face recognition algorithms, to be able to recognize masked faces as accurately as possible. But to train such models, it is important to have comprehensive and realistic masked face images for building robust modern

Face Recognition applications. Through this paper, we would be using deep learning techniques to apply realistic face masks on unmasked faces, so that we can generate pairs of images of the individual, both with and without a mask. We also attempt unmasking a masked individual, which essentially reconstructs the masked area of an individual.

The following are the main challenges that one will face while addressing this problem:

- 1) The various possibilities of the human face orientation in a picture make it challenging to apply a realistic face mask.
- 2) Unmasking (predicting the masked area of the face) a masked individual is a challenging application as it entails the prediction of facial features of the human, such as nose and lip's shape, color, orientation. This is especially difficult in the absence of pairs of images, which in complete essence is the problem we are striving to address!
- 3) Adjusting the fake mask's transparency and luminosity based on the average brightness in the histogram could be trickier, especially in under-exposed and high dynamic range images.

When we achieve our project objectives, it can provide the following benefits:

- 1) The generated paired data-set, where the individual is both, wearing as well as not wearing a mask, can aid in building/training robust modern Face Recognition Models.
- 2) Unmasking masked individuals would aid in surveillance applications. This has significant applications such as, but not restricted to, identifying individuals from their masked videos or images in surveillance footage.

II. BACKGROUND

Since the time of the pandemic, masks have become an integral part of our day-to-day lives. Even though today, when the world has discovered the vaccine [29] and the number of cases of COVID-19 has been mitigated, countries are still alert and taking precautions to stop the spread of the novel coronavirus. One of the few stringent restrictions, that still remains, is wearing masks indoors and in public places which is an effective way in stopping the spread of the virus [3]. But since the introduction of masks, modern face recognition

systems are finding it difficult to identify faces accurately with their masks on.

Several attempts were made to build face recognition systems with masks on. For eg, Safa Teboulbi et al [25] built a model pre-trained on CNNs such as MobileNet, ResNet Classifier, and VGG to detect whether the person is wearing a mask or not. Other attempts to build mask detector models were done as well [18] [16] [24] [22] [17]. But detecting whether the person is wearing a mask or not, did not solve the problem of poor performance of face recognition systems on masked faces. One of the attempts, to build models that could recognize faces with masks, includes the method by Hoai Nam Vu et al [27] who combined deep learning and Local Binary Pattern (LBP) features to recognize the faces with masks on using RetinaFace which was a supervised face detector to achieve an f1-score of 87% on a data-set of 300 masked subjects.

But even these methods fail to achieve higher accuracy. This is due to the fact that there aren't many high-quality, large enough data-sets of paired images with masked and unmasked faces. This data-set of pairs of images can be used to train deep learning models which can unmask the masked faces to predict the original faces without the mask. In his paper [15] Zhen Li proposes deep learning approaches to unmask a masked face in the image. Zhen Li first segments out the mask using a simple CNN and then the adversarial networks are used to remove the masked region and fill the space with an impainting of the face [19] [8]. But to build and train a model which gives better accuracy, a high-quality dataset of pairs of masked and unmasked faces is needed to train the deep learning adversarial model and not many such datasets exist. An attempt by Adnane Cabani et al [6] was made to generate a dataset of masked images called MaskedFace-Net by using image processing on a dataset of normal high resolution unmasked faces called Flickr-Faces-HQ (FFHQ). Although this method gave good results, it used image processing to mask the faces in the image and thus was bound to human errors, which demanded high human efforts and hours.

In our paper, we propose a deep learning approach to generate a large dataset of masked images out of unmasked or normal faces. We make use of a dataset that contains high-quality unmasked images and we use masked images from the same dataset to train a GAN to learn how to mask an unmasked face in an image. We will be using cycle GAN [31] which gives better results than other GANs in general and create more realistic images [4] [28]. CycleGAN is trained using masked and unmasked images to generate realistic masked images when an unmasked image is given as an input. Our model is also capable of unmasking the masked faces in an image. This unmasking technique can be eventually used to build more robust face recognition systems without having the individual remove their face mask. It can have numerous use-cases such as unmasking faces of individuals in surveillance footage to recognize the individual.

In the following sections, we also talk about the various models we trained using different combinations of hyper-parameters of the network, present the fakes generated in both unmasked and masked domains for visual inspection, deduce the performance of each model based on a quantitative evaluation metric, and further demonstrate the cyclic nature of the model.

III. METHODS

We use the CycleGAN[30] architecture to transform the images of unmasked faces into masked faces. The above problem is made more intricate by the fact that there does not exist a pair-to-pair data-set for masked to unmasked faces. We define pair to pair data-set as the same face without the mask and with the mask. An important distinction to make here is between paired and unpaired data-sets.

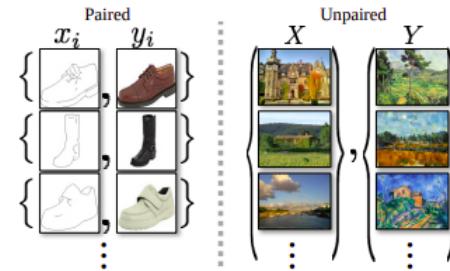


Fig. 1: Paired (left) and unpaired(right) training examples.
[30]

The paired training data consists of examples x_i, y_i^N where for each x_i , a corresponding y_i exists. Our data-set on the other hand resembles the right hand side in Figure 1, with no information on matches between pairs or no pairs available. In a variety of cases, it is difficult to obtain such a data-set or too expensive. Such is our case of masked and unmasked faces where no data-set is available for paired data-sets.

This leads us to the task of unpaired image-to-image translation. CycleGANs have previously been used for tasks such as horse-zebra, summer-winter etc. This task can be formulated as an image from a source domain X to a target domain Y in the absence of paired examples. We define the goal as mapping $G : X \rightarrow Y$ such that the distribution of images from $G(X)$ is indistinguishable from the distribution Y , using an adversarial loss. CycleGAN introduced another loss because this mapping is highly under-constrained. To address this, they couple it with an inverse mapping $F : Y \rightarrow X$ and introduce a cycle consistency loss to enforce $F(G(X)) \approx X$ and $G(F(Y)) \approx Y$.

This leaves us with the task of unpaired image-to-image translation. Previously, CycleGAN has been used for image translation tasks such as horse-zebra, summer-winter, etc. The task we are trying to address, i.e. mapping images from unmasked domain to masked domain and vice versa, can

be formulated as mapping, where an image from a source domain X to a target domain Y in the absence of paired examples. Our goal is to find mapping $G: X \rightarrow Y$ and $F: Y \rightarrow X$ such that the statistical distribution of the generated fakes $G(X)$ is indistinguishable from the statistical distribution Y , using an adversarial loss. CycleGAN introduced another loss because this mapping is highly under-constrained. To address this, they coupled it with an inverse mapping $F: Y \rightarrow X$ and introduce a cycle consistency loss to enforce $F(G(X)) \approx X$ and $G(F(Y)) \approx Y$.

To explain GANs, Generative Adversarial Networks include two networks, a Generator $G(x)$, and a Discriminator $D(x)$. The generator tries to generate the data based on the underlying distribution of the training data whereas the discriminator tries to tell apart the fake images from the real ones. They play an adversarial game where the generator tries to fool the discriminator by generating data similar to those in the training set. The Discriminator is fooled when the generated fakes are so real that it can't tell them apart. Both of them are trained simultaneously on data-sets of images, videos, and audio files.

The generator $G(x)$ model generates images from random noise and then learns the data distribution of how to generate realistic images. Random noise is given to the generator which outputs the fake images and the real image from the training set is given to the discriminator that learns how to differentiate fake images from real images. The output of Discriminator $D(x)$ is the probability that the given input is real if the output is 1.0, and if the output is 0 the given input is identified as fake. Thus our goal is to get the output 1 (real) for all the fake images.

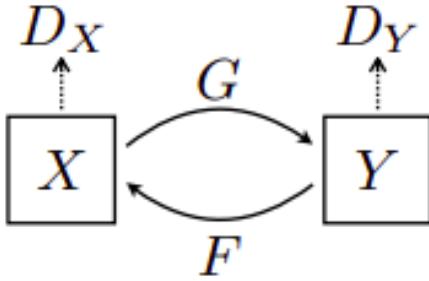


Fig. 2: Cycle GAN model[30]

1) CycleGAN: The CycleGAN approach has been used to add masks to our images. CycleGAN consists of two generators and two discriminators. Let's assume our domain X is unmasked images and domain Y is masked images. One of the generator (Generator G) translates images from domain X to domain Y. The second generator (Generator F) translates images from domain Y to domain X. Thus, we have two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$.

Each of these generators have an associated adversarial discriminator. We define them as D_y and D_x . The discriminator D_y encourages our Generator G to translate images from

domain X into the outputs indistinguishable from domain Y and vice versa.

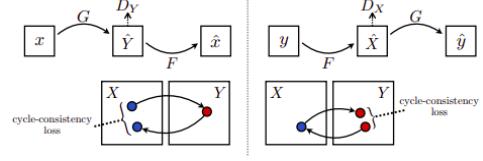


Fig. 3: Cycle consistency loss [30]

The intuition behind cycle consistency is that if we translate images from source domain to the destination domain and then translate the generated fake back to source domain, we should arrive at where we started: For forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) = x$, and backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) = y$. This means in our case if we generate an image we should be able to go from an unmasked image to a masked image and then get the same images back from a masked image. This loss provides further regularization for such an unconstrained problem.

We have two losses for this network: the Adversarial Loss and the Cycle-consistency loss. We can define the adversarial as

$$\begin{aligned} L_{GAN}(G, D_y, X, Y) = & \in_{y \sim (y)} [\log D_y(y)] + \\ & \in_{x \sim (x)} [\log(1 - D_y(G(x)))] \end{aligned}$$

The cycle consistency loss is forward cycle consistency loss + backward cycle consistency loss.

$$\begin{aligned} L_{cyc}(G, F) = & \in_{x \sim (x)} [| | | F(G(x)) - x | | | + \\ & \in_{y \sim (y)} [| | | G(F(y)) - y | | |] \end{aligned}$$

We define the total loss as:

$$\begin{aligned} L_{G, F, D_x, D_y} = & L_{GAN}(G, D_y, X, Y) + \\ & L_{GAN}(F, D_x, Y, X) + L_{cyc}(G, F) \end{aligned}$$

Through our training, we optimise over this loss.

2) Datasets: We use 2 different data-sets and curate them according to our use-case:

- 1) Flickr-Faces-HQ (FFHQ) data-set for unmasked images [12]
- 2) MaskedFace-Net data-set for masked images[5]

For unmasked faces, FFHQ is a high-quality image data-set of human faces, originally created as a benchmark for generative adversarial networks (GAN). The data-set consists of 70,000 high-quality PNG images at 1024×1024 resolution with variation and diversity in terms of subjects and objects in the frame. It also has good coverage of accessories such as eyeglasses, sunglasses, hats, etc.

For our masked face data-set, we use the MaskedFace-Net dataset, which is a dataset of human faces with a correctly or incorrectly worn mask (133,783 images) based on the Flickr-Faces-HQ (FFHQ) data-set. The masks are photo-shopped onto the faces. Although the dataset is based on FFHQ, the facemasks are incorrectly masked for most of the images. For both of these data-sets, we curate images based on the number of faces in images, mask placement, mask clarity, and realistic effect of the mask on the image. We use a subset of 6000 masked images and 6000 unmasked images for training and 1000 test images from each of them were used for testing.

3) Architecture and Training: The generator network architecture consists of three convolutions, several residual blocks [9], two fractionally-strided convolutions with stride 1/2, and one convolution. CycleGAN uses 6 blocks for 128×128 images and 9 blocks for 256×256 and higher-resolution training images. Similar to Johnson et al. [11], we use instance normalization[26]. For the discriminator networks we use 70×70 PatchGANs [10] [14] [13], which aim to classify whether 70×70 overlapping image patches are real or fake. To train the network we used a number of different hyper-parameter such as batch size, batch sequence, normalization type, types of optimization algorithm, change in loss from L1, log-loss, or L2 loss. The above experiments were done using 4 Nvidia Tesla V100 GPUs and took around 12 hours to run 200 epochs.

4) Evaluation: To evaluate our work, we use a qualitative metric in the form of visual inspection of images. The visual study is similar to the likes of perceptual studies of Amazon Mechanical Turks with participants shown a sequence of images asking them to label them as real or fake. We also use a quantitative metric in the form of FID score[23]. The Frechet Inception Distance (FID), is a metric for evaluating the quality of generated images and is generally used to assess the performance of generative adversarial networks. FID measures the distance between the distributions of generated and real samples.

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\sum_r + \sum_g - 2(\sum_r \sum_g)^{1/2})$$

Lower FID is better, meaning they are more similar to real and generated samples as measured by the distance between their distributions.

IV. RESULTS

In this section, we talk about the various GAN performance evaluation methodologies and metrics we used to evaluate our model[1]. We also present the fakes generated from our model in both masked and unmasked domains and ask the reader to visually assess the quality of the generated fakes. Finally, we evaluate the model based on a quantitative GAN metric, showcase all our training trials with variation in hyper-parameters, and talk about our inferences and deductions.

GAN's have proved to be quite effective at generating high-quality synthetic images in multiple domains. The generator models are not trained directly. Instead, the generator models are trained by the discriminator model, which learns to tell apart the real images from the fakes. GAN's lack an objective function, which makes it difficult to compare the performance of different models. This implies that there is no generally agreed-upon way of evaluating a generator model of a GAN. There are both pros and cons to GAN evaluation metrics[2]. The objective evaluation of GAN generator models remains an open problem. While there are several measures that have been suggested, as its stands, there is no agreement on which metric captures the strengths and limitations of models to be used as a standard and fair metric for model comparison[1].

Firstly, we talk about the manual inspection of images. Evaluation of GAN generators through the manual assessment of images synthesized by a generator model is one of the commonly used practices to assess the quality of generated fakes. This involves using the generator model to generate fake images, then evaluating the quality of the images in relation to the target domain. Below, in Figure 4 and Figure 5, we present the generated fakes from both generators G_x and G_y .



Fig. 4: Deep Fakes generated by the best Generator model G_x



Fig. 5: Deep Fakes generated by the best Generator model G_y

Finally, we talk about quantitative GAN evaluation metrics, which refer to the computation of numerical scores to encapsulate the quality of generated fakes. The model performance was evaluated based on Fréchet inception distance (FID)[23]. FID performs well in terms of robustness and computational efficiency. It is consistent with human judgments. A lower FID score implies the images are highly correlated i.e belong to the same domain. The underlying network of FID captures vision-specific features of an input image. These activations are computed for a set of real and fake generated images and are interpreted as a multivariate Gaussian distribution. The

TABLE I: FID scores FID_{AB} and FID_{BA} of generator models G_{AB} and G_{BA} with change in hyper-parameters

Trial	Batch Size	Norm	Learning policy	Loss	FID_{AB}	FID_{BA}
1	16	Instance	Linear	Vanilla	17.48	51.62
2	16	Instance	Linear	Lsgan	17.07	261.78
3	16	Batch	Linear	Vanilla	20.25	65.31
4	16	Batch	Linear	Lsgan	18.96	65.5
5	16	Instance	Cosine	Vanilla	22.15	47.68
6	16	Batch	Cosine	Vanilla	20.33	85.14
7	16	Batch	Cosine	Lsgan	32.18	71.29
8	32	Instance	Linear	Vanilla	18.06	51.5
9	32	Instance	Linear	Lsgan	19.14	48.39
10	32	Batch	Linear	Lsgan	23.08	100.31

distance between these distributions is determined using the Frechet distance. A lower FID score implies the distribution of the generated fakes matches the distribution of ground truth real images.

The training procedure involved rigorous experimentation with hyperparameters such as learning policy (Linear, Cosine), GAN loss (Vanilla, Lsgan), normalization techniques (Instance, Batch), and batch size (16, 32). Each model was trained for 200 epochs each. Post the training phase, we made predictions on the test data-set. The computed FID scores for both cases, i.e., translation from the unmasked domain (A) to masked domain (B) FID_{AB} , and from the masked domain (B) to the unmasked domain (A) FID_{BA} , using different hyperparameters can be seen in Table I.

Through the above training trials, we achieved the least FID_{AB} of 17.07 with a generator model G_{AB} , which was trained with the following values of hyper-parameters: batch size 16, instance normalization, linear learning policy and Lsgan optimization loss. We also observe that the FID_{AB} score ranges between 17.07 and 32.18. The best score of 17.07, coupled with small range means that the generator network G_{AB} is finding it relatively easier to learn how to apply a fake mask onto an unmasked face, irrespective of the variations in the hyper parameters.

Similarly, we achieved the least FID_{BA} of 48.39 with a generator model G_{BA} , which was trained with batch size 32, instance normalization, linear learning policy and vanilla optimization loss. The best score of 48.39 implies the generator network G_{BA} is finding it slightly difficult to generate the masked area of the face. We also observe that FID_{BA} score ranges between values 48.39 and 261.78, which implies the performance of the network varies significantly with change in hyper parameters.

Finally, we plot the variation of losses over 200 epochs training. From figure 10, we can see generator losses G_A , G_B and discriminator losses D_A , D_B keep oscillating without ever consistently increasing or decreasing with time. This is because cycleGAN uses min max optimization[30]. But with training over multiple epochs, we see a decrease in oscillations. In general, discriminator and generator losses trend is ignored because our generator is only as good as our discriminator and vice versa. For instance, if mediocre

generator can fool a bad discriminator into believing the generator images are real even if they are not. Hence, we track cycle consistency losses and their trend with number of training epochs. We see the cycle loss decrease with every additional epoch of training. This decrease implies that the generator models are tending towards a desirable state, where we can achieve the ideal translation, i.e. $G_B(G_A(x)) \approx x$ and $G_A(G_B(y)) \approx y$.

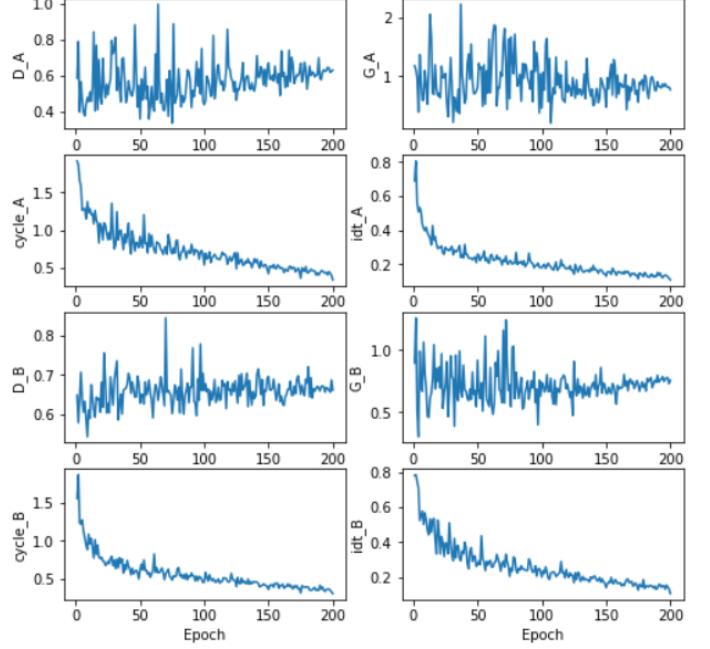


Fig. 6: Plots of losses D_A , G_A , $cycle_A$, idt_A , D_B , G_B , $cycle_B$, idt_B over 200 epochs training

This is the most important marker that decides the convergence of the model. To demonstrate the concept of cycle consistency loss, we feed an unmasked image to the generator G to apply a mask on it and generate a masked fake image. We input the masked fake we generated in the earlier step to the generator F to get back the unmasked fake version of the image. This essentially captures the translation from source to destination domain and back to source domain, to check if we arrive back where we started. This behavior can be observed in figure 7 below. Further, we also use inverse losses idt_A and idt_B decrease over a period of 200 epochs. The inverse losses enforce the condition that if a real image is fed to the generator, the output should be identity. As the both generators learn over training time, their performance on a real image tends towards identity, which can be seen in the plots of identity loss idt_A and idt_B .

Additionally, we explicitly outline the fake part of the generated image in both unmasked and masked image domains. This can be seen in figures 8 and 9.

Below, in figure 10, we also present some of the fakes generated in unmasked and masked domains during the training procedure. During the initial phases of training, it can be seen that both masking and unmasking capabilities of the model



Fig. 7: Cycle loss demonstration on test samples of unmasked domain X

are far from real. But with increase in number of training epochs, we see that masking looks realistic enough by the time we reach the 30th epoch. On the other hand, unmasking does not look good until we reach 120th epoch or more. We can see that unmasking takes more time to train in comparison to masking. This is because finding a translation from domain A to B is relatively easier task, as this entails finding the facial features where the mask ought to be and generating a fake. But the inverse problem of going from domain B to A is difficult problem and the masked area of the face could have any type of facial features possible, which could be infinite.

Finally, we only appreciate the good when we get a sense of how bad it was or how bad it could have been. Hence, in figure 11, we present some fakes generated in our initial trials that are pretty terrible and no where close to the spectrum of reality so that the reader realizes how far our model has come in generating realistic fakes in both masked and unmasked domains. In some cases, one might have to ask "Is that even human?!".

V. CONCLUSION

In this paper, we aimed and successfully formulated a methodology to generate a data-set containing pairs of images containing masked and unmasked faces of an individual. We achieved this using a deep learning approach in which we trained a cycleGAN on a data-set of high-quality unpaired masked and unmasked images. After the adversarial network was trained, it was able to generate realistic images of masked



Fig. 8: Pictures in the left column correspond to real images. Pictures in the right column correspond to fakes. The fake part has been outlined in the fakes in right column

faces from an input unmasked image. Since generative adversarial networks learn both ways, we could also use the same deep learning model unmasking the input image with its mask on. As there is no strong allotted metric of evaluation for GANs, we used both qualitative and quantitative metric to evaluate our model.

In the qualitative metric, the user was asked to guess and filter out the ground truth from the GAN-generated outputs out of 22 images that were randomly presented to the user. Each correct answer fetches the user 1 point. As it stands, we received 100 responses and the following is the analysis: Of the 2200 predictions, the users wrongly guessed 881 times. This gives us an accuracy of 40.08% which implies, the generated fakes were able to dupe the user 40.08% of time, which is impressive, considering the discerning sight we humans possess, thus validating our model through visual inspection. The statistics for the survey can be seen in Fig. 12, where the average score of the user is 13.24. For the quantitative metric, we used the Frechet inception distance that basically measures the closeness of the generated fake image by the generator and the ground truth by mathematically calculating the difference between scaled-down features of both the images. We also, further, experimented with the hyper-parameters of the our deep learning model to see their effect on the FID. We found out that for generating fake masked images from real unmasked images, a generator network trained with a batch size of 32 and Least Squares Generative Adversarial Network (LSGAN) loss, having an instance normalization, and a linear learning policy gives us the best generator. Similarly, the



Fig. 9: Pictures in the left column correspond to real images. Pictures in the right column correspond to fakes. The fake part has been outlined in the fakes in right column

generator trained with a batch size of 16 and Vanilla loss, having an instance normalization and a cosine learning policy was observed to be the best for generating unmasked images from original masked images as their input.

Now, we would like to talk about the advantages and disadvantages of our model to give the reader a picture of the limitations of our implementation. Talking about advantages, Our model does not require a paired set of images to train our network. In simpler terms, we don't need an unmasked and masked version of the same face to train our network. Hence this can be used in situations where a paired data-set is unavailable. Additionally, our implementation is a more unsupervised method compared to the other methods [15], where they remove the mask first using image segmentation and then fill the space using adversarial models. Talking about disadvantages, because of a concentrated data-set consisting of faces aligned in a particular direction, we faced a covariance shift, and thus our standard of fakes degraded on tilted or differently aligned faces. Also, the luminosity of the mask generated by our model is not taking the average brightness of the image into consideration. That means, the model would apply a mask of the same luminosity in both correctly exposed and underexposed images. This makes it relatively easier to identify the fake mask in under-exposed images or high dynamic range images.

Now, we would like to talk about the issues that we would like to address as a part of the future scope of our work. As mentioned earlier, the data-set we curated contain majority

Epoch	Fakes
1	
10	
20	
30	
50	
80	
120	
150	
200	

Fig. 10: Fakes of both domain A and B generated as a function of epochs



Fig. 11: Fakes generated by our earlier trials

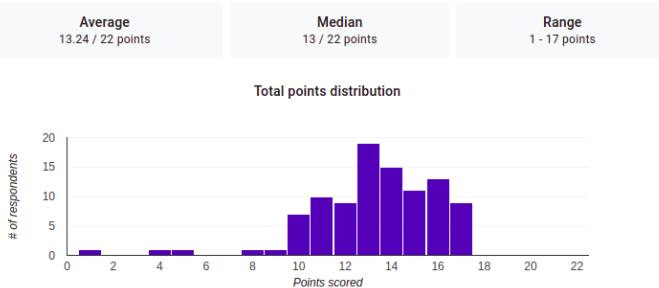


Fig. 12: The result from the survey of 100 people

of faces that are aligned forward. This means model fails to perform realistic masking and unmasking on faces that are out of alignment in the frame. Hence, in the future, we would like to address this by training on faces that have a broad range of facial orientations and poses, and enable the model to generate realistic fakes in both masked and unmasked image domains.

These models can be used to generate realistic-looking fake masked as well as unmasked images. The masking can be used to generate data for training modern face recognition systems to generate pairs of masked and unmasked images. While on the other hand, unmasking can be used to unmask individuals, for example, in surveillance footage, and can have an immense impact on the security systems. That being said, such generative and facial recognition techniques must be generalized to pictures of real faces with a variety of masks in order to be practically applied. In the future, we plan to use a range of masks so that it would help create fakes in both image domains, both with and without a varied set of masks.

REFERENCES

- [1] Hamed Alqahtani, Manolya Kavakli-Thorne, and Dr. Gulshan Kumar Ahuja. “AN ANALYSIS OF EVALUATION METRICS OF GANS”. In: July 2019.
- [2] Ali Borji. “Pros and Cons of GAN Evaluation Measures”. In: *CoRR* abs/1802.03446 (2018). arXiv: 1802.03446. URL: <http://arxiv.org/abs/1802.03446>.
- [3] John T Brooks and Jay C Butler. “Effectiveness of mask wearing to control community spread of SARS-CoV-2”. In: *Jama* 325.10 (2021), pp. 998–999.
- [4] Yash Burad and Kushal Burad. “A COMPARATIVE STUDY OF CYCLE GAN AND PROGRESSIVE GROWING GAN FOR SYNTHETIC DATA GENERATION”. In: () .
- [5] Adnane Cabani et al. “MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19”. In: *Smart Health* 19 (2021), p. 100144. ISSN: 2352-6483. DOI: <https://doi.org/10.1016/j.smhl.2020.100144>. URL: <https://www.sciencedirect.com/science/article/pii/S2352648320300362>.
- [6] Adnane Cabani et al. “MaskedFace-Net–A dataset of correctly/incorrectly masked face images in the context of COVID-19”. In: *Smart Health* 19 (2021), p. 100144.
- [7] *Covid 19 masks add an element of difficulty to identifying robbery suspects*. <https://cbs6albany.com/news/coronavirus/covid-19-masks-add-an-element-of-difficulty-to-identifying-robbery-suspects>.
- [8] Nizam Ud Din et al. “A novel GAN-based network for unmasking of masked face”. In: *IEEE Access* 8 (2020), pp. 44276–44287.
- [9] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.
- [10] Phillip Isola et al. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *CoRR* abs/1611.07004 (2016). arXiv: 1611.07004. URL: <http://arxiv.org/abs/1611.07004>.
- [11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. In: *CoRR* abs/1603.08155 (2016). arXiv: 1603.08155. URL: <http://arxiv.org/abs/1603.08155>.
- [12] Tero Karras, Samuli Laine, and Timo Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *CoRR* abs/1812.04948 (2018). arXiv: 1812.04948. URL: <http://arxiv.org/abs/1812.04948>.
- [13] Christian Ledig et al. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”. In: *CoRR* abs/1609.04802 (2016). arXiv: 1609.04802. URL: <http://arxiv.org/abs/1609.04802>.
- [14] Chuan Li and Michael Wand. “Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks”. In: *CoRR* abs/1604.04382 (2016). arXiv: 1604.04382. URL: <http://arxiv.org/abs/1604.04382>.
- [15] Zhen Li. “Facial Recognition for People Wearing Masks”. In: () .
- [16] Bishwas Mandal, Adaeeze Okeukwu, and Yihong Theis. “Masked Face Recognition using ResNet-50”. In: *arXiv preprint arXiv:2104.08997* (2021).
- [17] Elliot Mbunge et al. “Application of Deep Learning and Machine Learning Models to Detect COVID-19 Face Masks-A Review”. In: *Sustainable Operations and Computers* (2021).
- [18] Preeti Nagrath et al. “SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2”. In: *Sustainable cities and society* 66 (2021), p. 102692.
- [19] Kamyar Nazeri et al. “Edgeconnect: Generative image inpainting with adversarial edge learning”. In: *arXiv preprint arXiv:1901.00212* (2019).
- [20] Mei Ngan, Patrick Grother, and Kayee Hanaoka. *Ongoing Face Recognition Vendor Test (FRVT) Part 6A: Face recognition accuracy with masks using pre- COVID-19 algorithms*. en. 2020-07-24 2020. DOI: <https://doi.org/10.6028/NIST.IR.8311>.
- [21] *NIST Launches Studies into Masks’ Effect on Face Recognition Software*. <https://www.nist.gov/news-events/news/2020/07/nist-launches-studies-masks-effect-face-recognition-software>.
- [22] Mohammad Marufur Rahman et al. “An automated system to limit COVID-19 using facial mask detection in

- smart city network”. In: *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRON-ICS)*. IEEE. 2020, pp. 1–5.
- [23] Maximilian Seitzer. *pytorch-fid: FID Score for PyTorch*. <https://github.com/mseitzer/pytorch-fid>. Version 0.2.1. Aug. 2020.
- [24] Shilpa Sethi, Mamta Kathuria, and Trilok Kaushik. “Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread”. In: *Journal of Biomedical Informatics* 120 (2021), p. 103848.
- [25] Safa Teboulbi et al. “Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention”. In: *Scientific Programming* 2021 (2021).
- [26] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lem-
pitsky. “Instance Normalization: The Missing Ingredient for Fast Stylization”. In: *CoRR* abs/1607.08022 (2016). arXiv: 1607.08022. URL: <http://arxiv.org/abs/1607.08022>.
- [27] Hoai Nam Vu, Mai Huong Nguyen, and Cuong Pham. “Masked face recognition with convolutional neural networks and local binary patterns”. In: *Applied Intelligence* (2021), pp. 1–16.
- [28] Per Welander, Simon Karlsson, and Anders Eklund. “Generative adversarial networks for image-to-image translation on multi-contrast mr images-a comparison of cyclegan and unit”. In: *arXiv preprint arXiv:1806.07777* (2018).
- [29] Qianhui Wu et al. “Evaluation of the safety profile of COVID-19 vaccines: a rapid review”. In: *BMC medicine* 19.1 (2021), pp. 1–16.
- [30] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: *CoRR* abs/1703.10593 (2017). arXiv: 1703.10593. URL: <http://arxiv.org/abs/1703.10593>.
- [31] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.