Removing face-mask from an image using GANs

Basavaraj Rajendra Sunagad[1][7015644], Divesh Kumar[1][7010048], and
Abhijith Srinivas Bidaralli[1][7015642]

Saarland University, Saarbrücken, Germany

Abstract. Due to the Covid-19 situation, majority of world population has had to live under strict hygiene regulations, like having to wear a facial mask, this presents new problems in verifying the identity of the person. As the Covid-19 situation elongates there seems to be a need for facial features generation from images of people wearing facial medical masks. We wish to solve this problem using the generative features of the deep learning models. Recent advancements in deep learning models have given way to GANs (Generative Adversarial Networks) which help us to generate synthetic images from some conditional data. We wish to utilize this to generate predictions of a person's face occluded by a medical facial mask. Furthermore, we wish to run a face detection algorithm on the generated images to verify how many of the images generated using our GAN model are correctly detected as faces of people.

Keywords: Face-mask · GAN · Pix2Pix

1 Introduction

Generative adversarial networks (GANs) are novel breakthroughs in the field of artificial intelligence. They are able to produce new data samples that match the training data, GANs can synthesize images of human faces, though the faces do not belong to an actual person. GANs are able to reach this level of accuracy by coupling a generator that generates the target output with a discriminator distinguishes real data from the output of generator. The generator tries to fool the discriminator, and the discriminator tries not to be fooled. Though GAN models are able to generate new reasonable examples from a given data set, there is no way to influence the features/attributes of the generated images other than to try to figure out the intricate relationship between the latent space input to the generator and the generated images. Conditional Generative Adversarial Network, or cGAN, is a sort of GAN, that involves the conditional generation of images by the generator model. Images can be generated based on a condition which acts as a class label and this allows for a particular set of images to be generated selectively. We intend to use Pix2Pix GAN, a common approach for an image-to-image based translation. The approach is based on the cGAN, where target images are generated depending on what input images are given to it. The Pix2Pix GAN learns the loss function such that the generated image is reasonable in content and a reasonable translation of the input image

2 Related work

Pluralistic Image Completion by Chuanxia Zheng, Tat-Jen Cham and Jianfei Cai at NTU [6] produces multiple plausible image patches from a missing patch the model is able to generate multiple and diverse plausible results with various structure, color and texture, the drawback to this is that it is not trained specifically to generate nose and mouth which are key features of face covered by a face mask. Pix2Pix GANs were published in year 2017 (CVPR) by Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros [1] in which they introduded a new cGAN architecture based on U-Net generator and Patch-GAN as discriminator. Pix2Pix was a simple implementation that generalized the task of image to image translation, Here the generator produces images based on provided input image along with random noise, the discriminator tries to classify a patch of image as real or fake. Pix2Pix architecture is a better fit to achieve our goal but the l1 regularization used by Pix2Pix tends to produce blurred images and certain unwanted artifacts

3 Proposed method

3.1 Dataset synthesis

As there is no readily available dataset of images of people wearing facial masks, we had to synthesize our own dataset. We start with a subset of Celeb-A dataset[5] and setup a data pre-processing pipeline to artificially add medical masks onto the faces. Celeb Faces Attributes dataset contains 202,599 images of celebrities faces of the size 178×218 from 10,177 celebrities, each image is annotated with 40 labels that indicate facial features like hair color, age and gender. We detect key facial features like nose bridge and chin to place the mask appropriately [2]. We also used different style of masks to add onto faces of people to generalize better as can be observed below. Here are a few images from the synthetic data created by us.

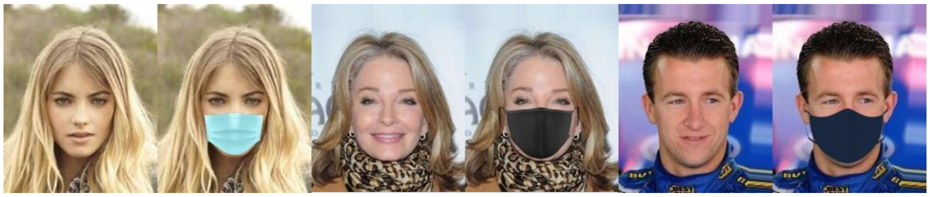

Fig. 1. Some images generated after adding mask onto celebrity images.

3.2 Pix2Pix

Pix2Pix is a cGAN based on U-Net architecture for generator and Patch-GAN for discriminator[4,3]. The main purpose of such a network is to generalize image to image translation task. The notwork tries to map input image to output image by learning a general loss function.

Pix2Pix uses a mini version of U-Net for generator model where we take an image and apply some convolution layers then down sample(stride 2 or pooling layers), again convolution layers and down sample, then up sample and some convolution layers. There are Skip connections from down-sampling layers (i) to up-sampling layers(n-i) to preserve spatial information of images, where n is the total number of layers.

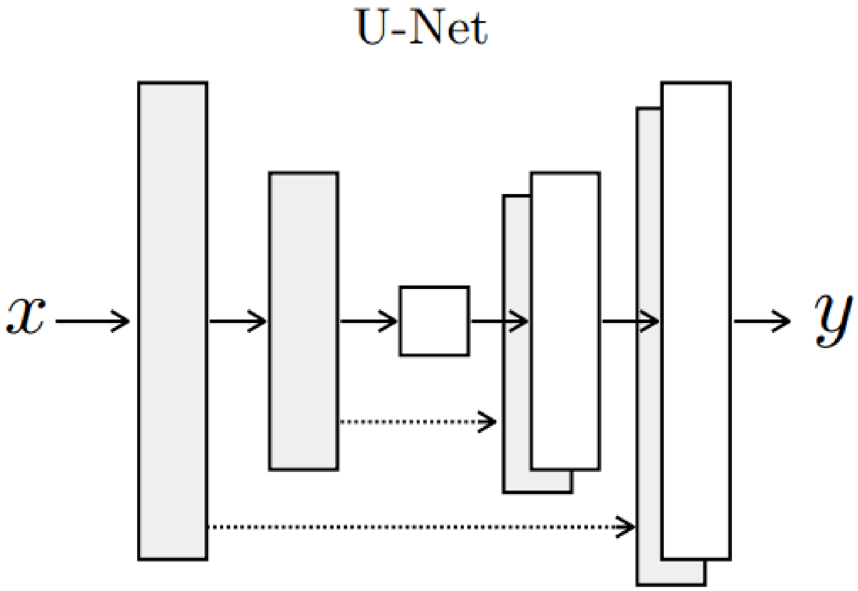

Fig. 2. U-Net architecture

The authors of Pix2Pix introduce a novel GAN for the discriminator called Patch-GAN, that tries to classify a patch pf size $N \times N$ of an image as real of fake. This has several benefits, we have fewer parameters to train which makes the model faster, restricting to a smaller patch tends to produce sharper images and we can easily scale the model for higher resolution images. The authors of Pix2pix experimented with several patch sizes and found a patch size of 70×70 to be producing best results

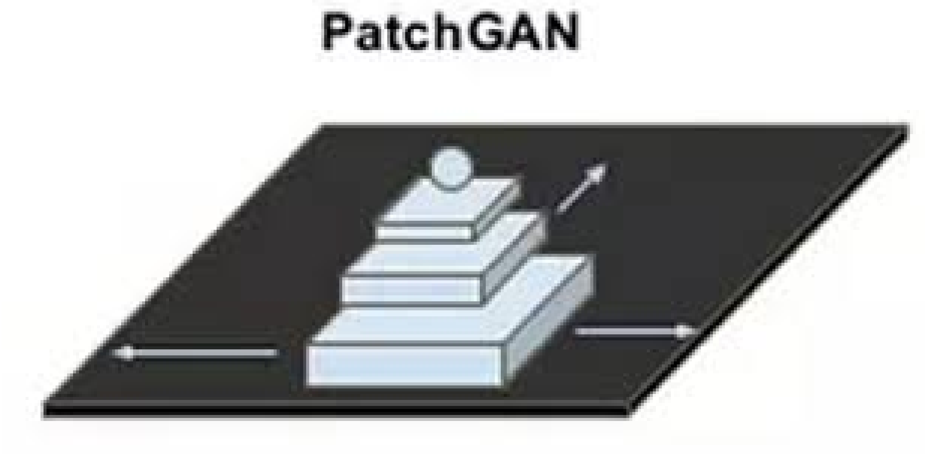

Fig. 3. A Patch-GAN architecture

The objective of Pix2Pix is similar to a conditional GAN objective

$$L_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}[\log(1 - D(x, G(x, z))] \tag{1}$$

where G tries to minimize the objective i.e, produce results as close to labels as possible and D tries to maximize the objective i.e, tries to distinguish between real and generated images as best as possible. Authors of Pix2Pix have used l1 regularization with Generator to be near ground truth output.

$$L_{L1}(G) = E_{x,y,z}[||y - G(x, z)||_1] \tag{2}$$

The final objective is

$$G^* = argmin_{(G)} max_{(D)}[L_{cGAN}(G, D) + \lambda L_{L1}] \tag{3}$$

3.3 Modification of loss function

Cosine similarity measures the similarity between two vectors of an inner product space[1]. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. We chose cosine similarity loss because it is robust to illumination changes, as a vector V and a.V are maximally similar while computing cosine similarity. Hence Cosine similarity is a relative comparison instead of absolute. L1 loss is absolute. We also chose cosine similarity over L1 loss because it produces better loss values than L1. It computes the dissimilarity between two images. If two images are dissimilar, it will be penalized more, i.e the loss will be more. If two images are similar the loss will be less. In L1 loss we are taking sum of the absolute differences between the two image vectors. In the original implementation, L1 loss was chosen to reduce the blurring effect. It was not checking for the similarity between the two images.

$$similarity = \frac{x_1.x_2}{max(\|x_1\|_2.\|x_2\|_2, \epsilon)}$$

4 Experimental Results

We trained on a subset of Celeb-A dataset as Pix2Pix takes longer to train. We took a training set of 4000 pairwise images of masked and unmasked images. Masked face images were fed to Generator as conditions and corresponding unmasked face images were used as labels. Training over 4000 images on GPU took around 8hrs for 500 Epochs. We tried out two variations of the cosine similarity equation. We first computed the loss as mean of all cosine similarity losses across the images in a given batch. We then computed the loss as sum of all cosine similarity losses across the images in a given batch.
Comparative results are as below:

Table 1. FID scores

Loss Function	FID Score
Pix2Pix with L1 Loss	**65.73**
Final cosine similarity loss as the mean of all cosine similarity losses for a given batch.	**56.26**
Final cosine similarity loss is the sum of all cosine similarity losses for a given batch.	**50.81**

5 Conclusions and Future Work

From the results we can say that the cosine similarity loss works really well in place of the L1 loss. Particularly, the variation we chose, which was the sum of all cosine similarity losses over a given batch worked the best. The cosine similarity loss penalized well whenever the images were highly dissimilar. We have observed during training models that the convergence is not so stable this

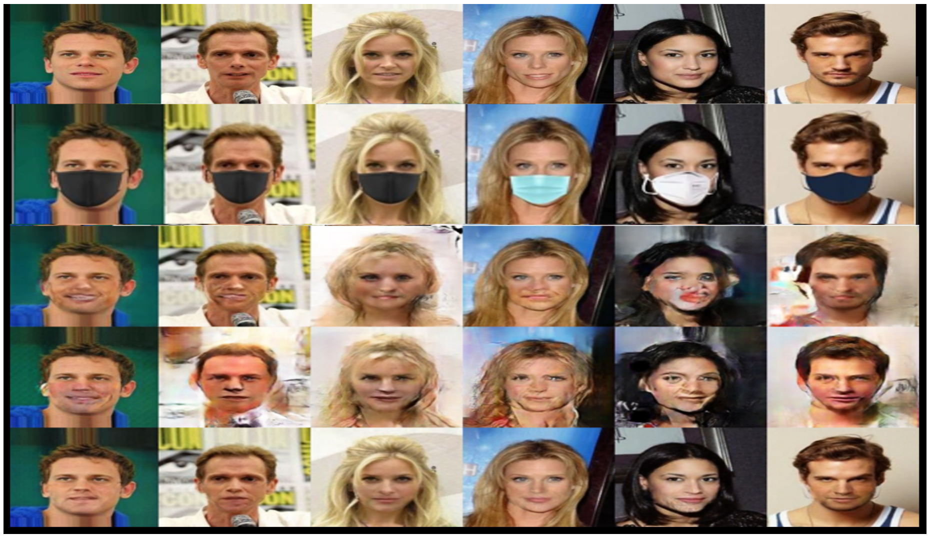

Fig. 4. 1st Row : Original Images, 2nd Row : Masked images, 3rd Row : Pix2Pix results, 4th Row : Mean of all cosine similarities, 5th Row : Sum of all cosine similarities.

is because GANs in general suffer from training instability, adding GAN training stabilizers WGAN-GP for better training stability and performance will improve the model performance.

6 Assignments of each group member

6.1 Basavaraj

1. Creating the data-set of pairwise images of masked and unmasked faces.
2. Setting up frame work for evaluation of results.
3. Training and evaluating Pix2Pix model with l1 loss.

6.2 Abhijith

1. Worked on the sum variation of the cosine similarity loss.
2. Implemented the formulation, training and evaluation of the Pix2Pix model based on this loss.
3. Generated the FID score for this model, which was used in the comparative results.

6.3 Divesh

1. Worked on the mean variation of the cosine similarity loss.
2. Training and evaluating Pix2Pix model with mean of cosine similarity loss.

References

1. Cosine similarity https://www.machinelearningplus.com/nlp/cosine-similarity/
2. Bhandary, P.: Addition of facial masks onto images. https://github.com/prajnasb/observations
3. Brownlee, J.: How to develop a conditional gan (cgan) from scratch (2017), https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/
4. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)
5. Liu, Z., Luo, P., Wang, X., Tang, X.: Dataset : Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015), https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html,
6. Zheng, C., Cham, T.J., Cai, J.: Pluralistic image completion (2019)