Spring 5-24-2019

Fast High Resolution Image Completion

Chinmay Mishra
San Jose State University

Fast High Resolution Image Completion

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Chinmay Mishra

May 2019

The Designated Project Committee Approves the Project Titled

Fast High Resolution Image Completion

by

Chinmay Mishra

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2019

Dr. Teng-Sheng Moh	Department of Computer Science
Dr. Katerina Potika	Department of Computer Science
Prof. Kevin Smith	Department of Computer Science

ABSTRACT

Fast High Resolution Image Completion

by Chinmay Mishra

This paper presents a method for image completion, an active research area in the field of computer vision. The method described in the paper aims at achieving comparable results to other state of the art methods with approximately four and a half times reduction in training time. It is a two step procedure which involves image completion and enhancing the resolution of the completed image. We use the SSIM metric to evaluate the quality of the completed image and to also time our model against other image completion models.

Key Terms - Generative Adversarial Network, Convolution Neural Network, Image Completion, Image In-painting

ACKNOWLEDGMENTS

I would like to thank my project advisor Dr. Teng Moh for his support and guidance through the course of the project. His weekly meetings and insights on the road blocks that I used to hit helped me complete my project with ease. I would like to thank my committee members Dr. Katerina Potika and Prof. Kevin Smith for their time and suggestions.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

The task of image completion involves the generation of patches for the missing regions of an image. This task can be broken down to the prediction of the context of the image. It is fairly complex because the model has to not just complete the image but also understand the structure of the objects in the image to complete it consistently. There are numerous applications of image completion such as image restoration, removal of objects from images and enhancing resolution of low resolution images. There are a lot of methods that have been proposed in this field from completing an image based on a patch or using deep networks to understand the structure of the object and complete it. But in all these techniques the resultant networks are huge and have lots of variables which make them very time consuming to train and require a vast amount of training data. We propose a model that will complete images with quality comparable to other state of the art methods while taking lesser time to train and consuming lesser memory.

CHAPTER 2

Related Work

A large body of literature exists on various aspects of image completion. A few of the papers which are related to the method proposed in this paper are discussed.

2.1 Image generation

Since the introduction of GAN(Generative Adversarial) by Ian Goodfellow in his paper [1], it has been the center of research for various types of problems such as image generation, image transformation, super resolution and image completion. GANs are used whenever there is a requirement of predicting data in a particular distribution. They consist of two neural networks, a generator, G, and a discriminator D. The G network generates data from a random vector z, sampled from a prior distribution p_z while the D evaluates whether the incoming data is generated or sampled from data distribution p_{data}. The G and D networks are trained by optimizing the loss function:

$$\min_G \max_D V(G,D) = E_{h\ p_{data}(h)}[log(D(h))] + E_{z\ p_z(z)}[log(1-D(G(z))], \quad (1)$$

where h is the sample from the p_{data} distribution; z is from a random encoding in the latent space. Using GAN Radford in the paper [2] further developed a more stable architecture which comprised of convolutional layers with strides used for reducing the size of the image while removing all the max pooling layers. The architecture also used batch normalization on all the layers to normalize the data being passed to every layer.

2.2 Image Transformation

Image transformation is the problem to take an image and apply the desired transformations on it. Justin Johnson in his paper [3] introduced perceptual loss for the task of image transformation. Image transformation tasks previously used

per-pixel loss between the output and the generated image by Dong in the paper [4] for super-resolution and Cheng [5] for colorization of images, the switch to perceptual loss gave similar results but three times faster. The perceptual loss unlike pixel-wise loss which focuses on per pixel loss focuses more on the high-level feature similarity. Perceptual loss is also used in the paper [6] to transform low resolution images to high resolution images. The mean squared error(MSE) loss function results in images appearing overly-smooth, whereas using perceptual loss generates images with high texture detail.

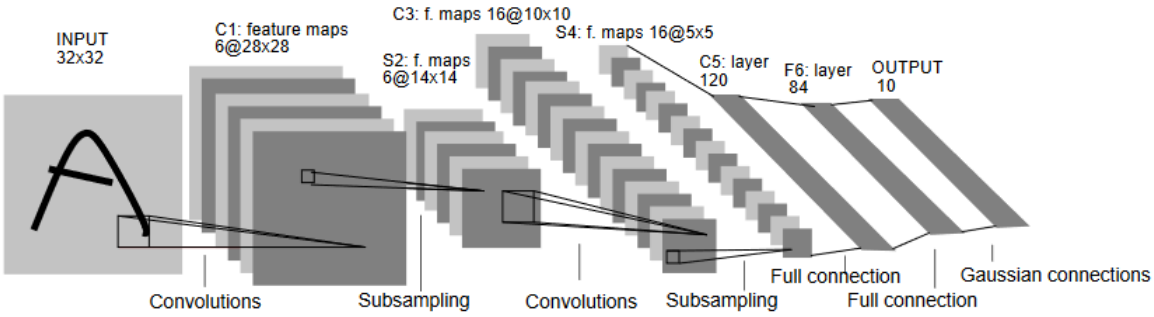

Figure 1: Architecture of LeNet-5, a convolutional neural network for digit recognition. **Source:** Adapted from [7]

2.3 Image Completion

There have been many different methods proposed to complete an image. Rares in the paper [8] introduces an edge-based image restoration method where the model looks at the surrounding edge of the hole and interpolates from that to fill the missing part. This method fails to complete the image consistently when the missing part of the image contains objects which aren't in the other parts of the image. It also fails when the missing parts are huge in the image. To overcome the shortcomings in the methods that use techniques such as diffusion which couldn't handle large missing parts, Pathak in the paper [9] introduced a model that learns the features of the image which helps the model generate a plausible hypothesis for the missing part. In the

paper they trained the model using both the pixel-wise reconstruction loss, as well as an adversarial loss. The adversarial loss helped the model to learn structures in the images and complete the images more accurately.

CHAPTER 3

Method

In this paper the proposed model is broken down into two parts. The first part is the image completion part which is followed by the enhancing part. The image completion part comprises of three networks, a single completion network and two discriminator networks which are used to train the completion network to complete the image realistically. The training process involves the discriminator networks trying to determine whether the image has been completed realistically, whereas the generator tries to fool the discriminator into believing that the completed image is real. The completion network is fed a lower resolution image which the network completes and the completed image is then fed to the super resolution network which transforms the image to a higher resolution. The generator and the discriminator networks are based on the architecture discussed in the paper [10] by Satoshi Iizuka with constraints described in the paper [2] on the architecture, which will make model more stable to train. The completed images are then fed to another network to enhance the images to a higher resolution. The image enhancement network is based on the model described in paper [6], with loss function as perceptual loss first discussed in the paper [3], to transform images instead of pixel-wise loss for the reconstruction of images.

3.1 Convolutional Neural Network

In the model proposed in this paper the networks are built using convolutional layers, which were introduced in papers [11] and [7] . The convolutional neural network architecture was introduced because the regular neural network architecture wouldn't scale well for images as for even a 32x32x3 image, the neural net would have 32*32*3 = 3072 weights and for a 256x256x3 image the number of weights would go up to 120,000 weights. The convolutional layers arrange the images as 3D vector unlike the traditional architectures where the data is represented by a single layer of neurons.

The Convolutional layers use operators to reduce the size of the data conserving the spatial structure of the image. In a feed forward network the image is flattened and the spatial structure of the image is not maintained, hence a convolutional network is used which uses filters in order to capture the spatial structure of the image. In the figure 1 the architecture of the LeNet-5, which shows the working of a convolutional neural network. The figure shows how the input image is convoluted into different sizes of 3D vectors by using filters which organized as different planes in every layer. The convolutional layers are usualy followed with a non-linear activation function like a Rectified Linear Unit (ReLU) which is decribed in the paper [12].

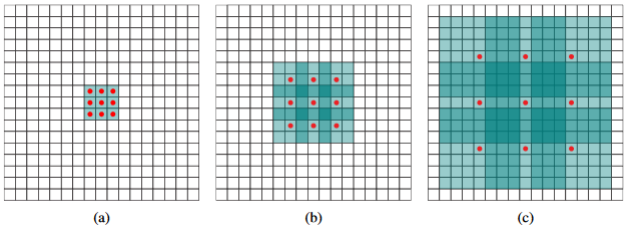

Figure 2: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage.(a) F1 layer is produced from F0 by 1-dilated convolution, i.e. each element in F1 has a receptive field of 3x3. (b) similarly in image b a 2-dilated convolution is used; each element has a receptive field of 7x7. (c) here 4-dilated convolution layer is shown which has a receptive field of 15x15.
Source: Adapted from [13]

3.2 Dilated Convolutional Layer

The model in addidtion to the traditional convolutional layers also uses a special kind of convolutional layer called Dilated Convolutional Layer introduced in the paper [13]. In this paper the authors talk about how semantic segmentation is a different problem than image classification and the conventional convolutional network are not efficient for that work. So they introduced dilated convolutional network which enables exponential expanse of the receptive field without the loss of resolution or coverage. The figure 2 shows that even though the number of parameters remain same

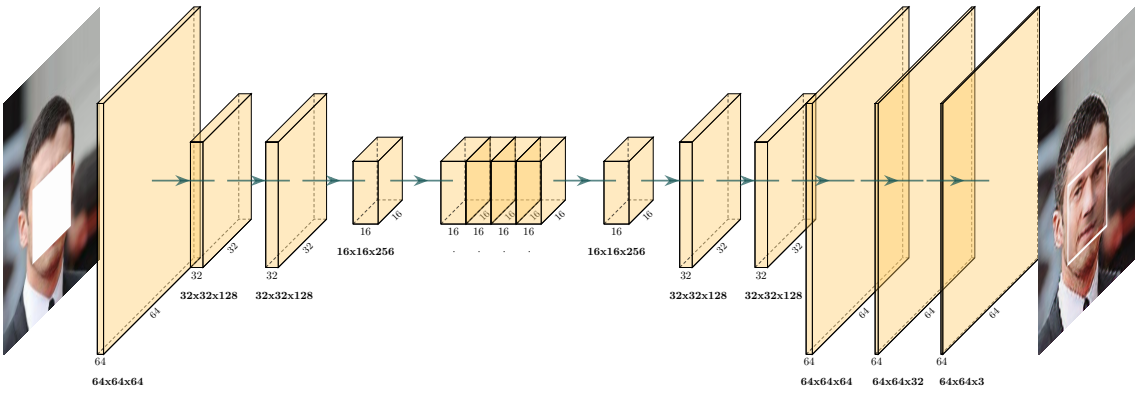

Figure 3: Overview of the generator architecture of image completion network.

in each layer but the receptive field grows exponentially. This allows the network to capture large features without increasing the number of trainable parameters.

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t) \tag{2}$$

$$(F * lk)(p) = \sum_{s+lt=p} F(s)k(t) \tag{3}$$

Equation 2 is for the standard convolution and equation 3 is for a dilated convolution. The summation in the second equation is s+lt=p which will cause the kernel to skip some points during convolution which in turn increases the area of the input that can be used as input without increasing the number of weights required for the convolution.

3.3 Batch Normalization

The convolutional and the dilated convolutional layers in the model are followed by a batch normalization layer. These layers were introduced by Sergey Ioffe in the paper [14]. Deep neural networks usually suffer the phenomenon called internal covariate shift, which happens due to the change in the distribution of inputs for every layer because of which the learning rates have to be low and the parameter initialization becomes an issue. To overcome these problems batch normalization

layers are introduced after every convolutional layer. Since the convolutional layer modifies the data, the batch normalization layer helps in normalizing the value before it is sent to the next layer. This normalization of the inputs before passing them to the next layer helps overcome the internal covariate shift problem and allows the model to have a high learning rate.

3.4 Deconvolutional Layer

Deconvolutional layers are used in interpolation of images from their feature vectors. These layers were used in [15] for image segmentation and also in [2] for generation of images. There are various interpolation methods like nearest neighbour, bi-linear and bi-cubic but the deconvolutional layer instead has learnable parameters which are trained in a similar fashion to convolutional layer but in a backward manner. These layers help the model learn high-level features during the decoder part of the network while generating the image from the latent vectors.

3.5 Completion Network

The Completion network comprises of multiple fully-connected convolutional layers with a few dilated convolution layers sandwiched in between them. The input to the completion network is an RGB image with a mask(the mask contains 1 for parts that have to be completed) and the output of the network is an RGB image. The input image is first resized to 64X64 and everything is scaled down to that resolution including the holes in the images. This approach is different from the one discussed in the paper [10] as our focus is to make the training of the network faster and more stable. The architecture of the completion network follows that of an encoder-decoder. The input image is processed with convolution operators to extract features and then passed on to the decoder part which restores the image to the input resolution completing the holes in the input image. The encoder uses convolutional and dilated

convolutional layers to extract the features and the decoder uses deconvolutional layers to reconstruct the image. All the layers in the completion network are followed by batch normalization, which helps in the making the model more stable as found in the paper [2] and also helps in the overcoming the problem of internal covariate shift.

3.6 Context Discriminators

The paper [10] defines two different types of discriminators global and local. The discriminators together are trained to discern whether the input image is an image from the dataset or is a completed image using the generator in the completion network. The context discriminator consist of convolutional layers which extract features from the image over several layers and finally the features from both the networks are concatenated which is used to predict whether the image is real or has been completed. The global discriminator takes as input the whole image, which is in our case is a 64x64 image and compresses to output a single 1024 dimensional vector. This global discriminator is used to check the consistently of the whole image, whether it has similar features to the images the network has been trained upon. The local discriminator does similar thing but the input is 32x32 patch of the image around the completed region. The output from both the discriminator networks are then concatenated together into a 2048-dimensional vector which are then passed through a sigmoid activation layer to give a value in the range [0, 1], which represents the probability of the image being real.

3.7 Residual Network

In the field of computer vision, the idea of stacking up layers of convolution to make the model better at image recognition or image generation worked till 20+ layers, after which the the accuracy dropped due to problems like vanishing gradients. To overcome this problem the idea of residual layers were introduced by Kaiming in the

paper [16]. The residual layers help in mapping identity functions which is otherwise very complicated using the convolutional kernels. These residual layers have used to improve the accuracy and speed for single image super resolution [17],[18].

Figure 4: Flowchart of the whole model with the completion and the super resolution modules

3.8 Loss Functions

The Mean Squared Error (MSE) function which calculates the pixel wise error between the images usually struggle to handle the high level features between the images in question. Therefore, minimizing MSE results in generation of overly-smooth images as it finds the pixel-wise average of plausible solutions. In this paper, we will be using perceptual loss [3] along with an adversarial loss as in defined in the paper [6].

3.9 Super Resolution

The architecture for super resolution is based on architecture in the paper [6]. The architecture consists of a generator and a discriminator network which are trained in an adversarial manner [1]. The generator network which has residual blocks (similar to the model described in [3]) of convolution layers, batch normalization layers and relu activation. These blocks are sandwiched between convolutional layers which help in tranforming the shape of the data that is being passes throught the network. The resolution of the image is increased by two trained sub-pixel convolution layer as proposed in paper [18]. The discriminator network is based on the architectural guidelines summarized in paper [2] and it is trained to maximize equation (4). In the super resolution network, the generator instead of an upsampling layer as mentioned

in the paper [3] uses transposed convolutional layer as discussed in the completion network. The transposed convolutional layer helps in a more accurate construction of the image in the decoder part of the network in comparison to an upsampling layer.

$$\min_{\theta G}\max_{\theta D}\mathbb{E}_{I}^{HR} \backsim p_{t}rain(I^{HR})[logD_{\theta D}(I^{HR})] + \mathbb{E}_{I^{LR}} \backsim p_{G}(I^{LR}[log(1 - D_{\theta D}(G_{\theta G}(I^{LR}))] \tag{4}$$

Here, $D_{\theta D}$ is the discriminator and $G_{\theta G}$ stands for the generator. Both the networks are trained in an adversarial manner to optimize the equation (4).

3.10 Optimization Algorithm

Every deep learning problem essentially requires us to optimize a loss function using a neural network. To facilitate the search for the minima or to optimize on a loss function, algorithms are used called as optimization algorithms. For our problem of image completion and super resolution we will use the Adam optimizer [19]. In the paper, the authors have shown promising results for adam optimizer in terms of speed of training over other optimization algorithms such as adadelta, stochastic gradient descent.

3.11 Training

The image completion network and the image enhancement network are trained on the celebA dataset but separately. The input images to the network are resized to 64x64. These images are then used to create masks which contain the information of the missing parts. The low resolution images and the masks are then passed to the completion network to output a complete image, which is then evaluated by the discriminator to be real or generated.

Image Completion Network The generator of the completion network is denoted by $C(x, M_c)$ and the discriminators are denoted as $D(x, M_d)$, with x the input image,

M_c the completion mask and M_d the mask passed to the discriminator. The networks for the completion part are trained using two loss functions together, a weighted Mean Squared Error (MSE) loss and an adversarial loss function. The MSE loss is used to allow the networks to train in a stable manner [10], whereas the adversarial loss is used to generate more realistic images [9]. The MSE loss function is defined as,

$$L(x, M_c) = ||M_c \odot (C(x, M_c) - x)||^2, \tag{5}$$

where $\odot$ is the pixel-wise multiplication and $||.||$ is the Euclidean norm. The adversarial loss turns the usual optimization problem into a min-max optimization problem where the discriminators are trying to get better at discriminating between the images, hence increasing the loss and the generator is trying to fool the discriminator into believing the generated images are real, hence decreasing the loss. The adversarial loss is defined as,

$$\min_C \max_D \mathbb{E}[logD(x, M_d)] + log(1 - D(C(x, M_c), M_c)], \tag{6}$$

where M_d and M_c denote a random mask and input mask respectively.

Super Resolution Network After the completion network has completed the images realistically but at a low resolution, the super resolution network is used to enhance the image to a higher resolution. While training, the aim of the network is to estimate a high resolution image from a given low resolution image. The high resolution image is represented by I^{HR} and the low resolution image is represented by I^{LR}. The training phase involves creation of low resolution images by passing the high resolution image through various filters like Gaussian. The goal of this network is to formulate a generation function which when fed low resolution images can generate its corresponding high resolution images. To achieve this, a feed-forward CNN $G_{\theta G}$ is trained parameterized by θG. θG here denotes the weights and the baises of each layer and is optimized by a loss function l^{SR} . The perceptual loss l^{SR} is defined as a

weighted loss of several loss components that help in defining certain characteristics of the recovered high resolution image.

Most super resolution papers use the MSE loss defined in equation 7 for image transformation [4, 18], but the solutions for the MSE often lack high frequency content and have overly smooth texture.

$$l^{SR}_{MSE} = 1/(r^2WH)\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I_{x,y}HR - G_{\theta G}(I^{LR})_{x,y})^2 \tag{7}$$

Therefore, the MSE loss was not used and based on the ideas of [3, 20, 21] a loss function was built based using the VGG network defined in [22]. This loss function uses the VGG network to extract features from the generated image and compares that with the features extracted from the actual image. This loss function is defined equation 8.

$$\begin{aligned} l^{SR}_{VGG/i,j} = 1/(W_{i,j}H_{i,j})\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\Theta_{i,j}(I^{HR})_{x,y} \\ -\Theta_{i,j}(G_{\theta G}(I^{LR}))_{x,y})^2 \end{aligned} \tag{8}$$

In addition to the modified MSE loss another adversarial loss is also used with a discriminator which will push the generator towards generating more realistic images. The generative loss l^{SR}_{Gen} is defined as the summation of the probabilities of the discriminator $D_{\theta D}(G_{\theta G}(I^{LR}))$ over all training data.

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -logD_{\theta D}(G_{\theta G}(I^{LR})) \tag{9}$$

CHAPTER 4

Experiments

Both the completion and the super resolution networks are trained on the celebA dataset [23].

4.1 Image Completion Network

The image completion network has three networks, one completion network and two discriminators. The completion network consists of convolutional layers, dilated convolutional layers and deconvolutional layers. The aim of the paper is to devise a model that completes image comparable to other state of the art methods with a significant reduction in the training time. The architecture used for the generator is shown in table 1. The resolution of the input image is 64x64, this is done so as to reduce the amount of computations that are done. This also enables us to have a comparatively shallower model which can detect the important features and complete the image. We will look at some of the metrics applied on the completed images through different models. We will apply these metrics locally on the completed region and globally on the whole image measure the consistency of whole image as well as the completed region. The training phase of the completion network involves initially training the generator network alone, during this time since the discriminator has not been trained and cannot discriminate whether the completed image is real or fake. During this phase all the images sent to the discriminator is randomly classified as real or fake. After the generator is trained for a few iterations, the discriminator is trained to discriminate real images and fake ones. This creates a sudden peak in the generator's loss, as now the discriminator is able to identify fake images from real. This sudden peak in generator's loss creates a sudden dip or peak in the values of different metrics.

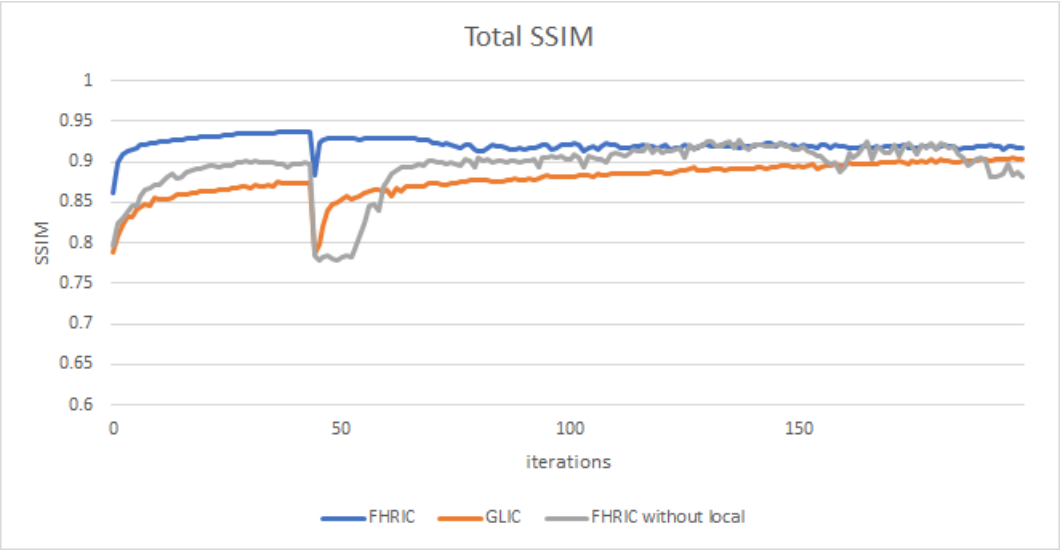

Figure 5: The total ssim of different models over 200 iterations for 30000 images.

1. `SSIM` - The structural similarity index (SSIM) metric is used to evaluate the quality of the completed image. SSIM helps us to quantify the degradation in image quality, after it has been processed and operations like image completion, restoration and compression have been applied over it. SSIM was first introduced in the paper [24], where the author shifted the focus of the measuring the image quality from visible errors to degradation of structural information in the generated images. In other papers, they have used SSIM metric [15], but they have used it on the whole image. The problem with using it on the whole image is that the SSIM value is high regardless of how well the missing parts have been completed. We will be using SSIM for the missing part as it is a better indicator of the quality of the completion.

2. `MSE` - The Mean Square Error (MSE) metric is used to find the pixel wise difference between the generated image and the original image. The problem with this metric is that it doesn't look for the high level features that were present in the images, instead looks at a pixel level how the completion has happened.

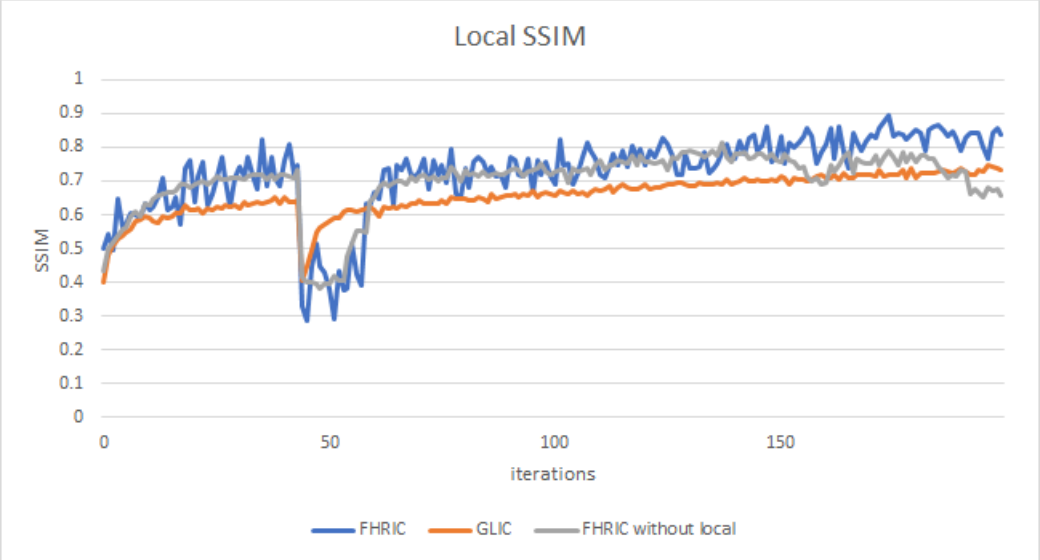

Figure 6: The local ssim of different models over 200 iterations for 30000 images.

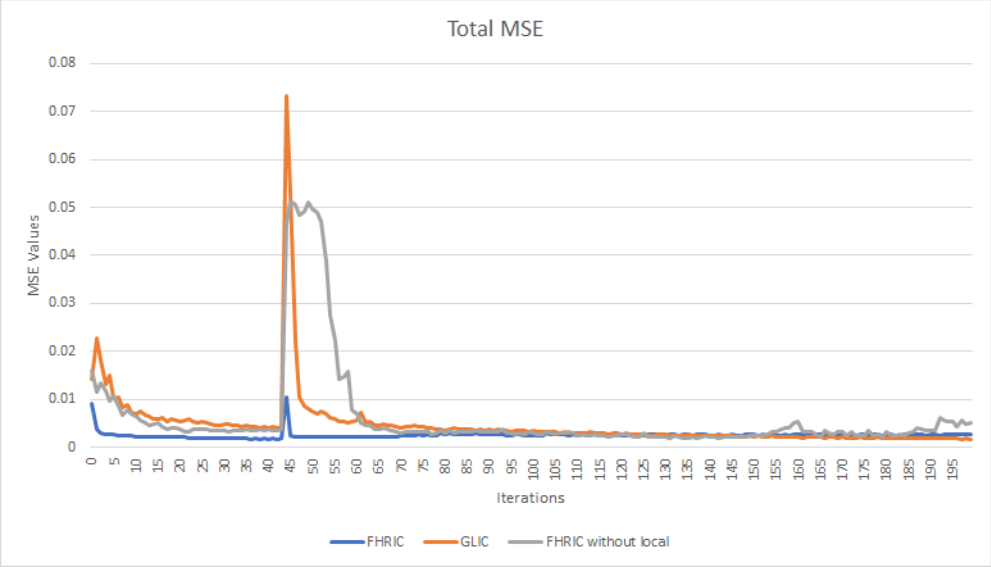

Figure 7: The total mse of different models over 200 iterations for 30000 images.

This metric does not give us a clear indication of which model performs better. It is clear from both the graphs (7 and 8) of the total and local mse values that this metric does not help us differentiate between the models.

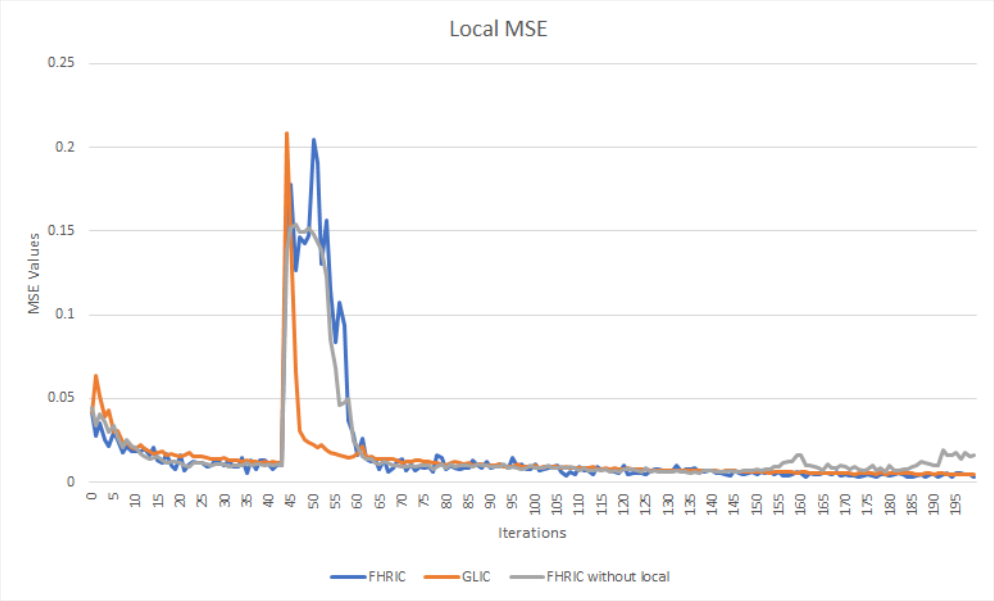

Figure 8: The local mse of different models over 200 iterations for 30000 images.

3. `PSNR` - Peak Signal to Noise Ratio is another metric for measuring the image quality degradation. Higher value of PSNR signifies better quality of the generated image compared to the original image. PSNR is based on MSE, MSE calcluates the cumulative squared error whereas PSNR represents the peak error between the generated and the original image.

$$MSE = (\sum_{m,n}(I_1(m,n) - I_2(m,n)^2)/m * n$$

Here m and n represent the row and the column of the pixel being compared. Using this the PSNR is computed as follows:

$$PSNR = 10\log_{10}(R^2/MSE)$$

Where R is the maximum fluction occuring in the input image. The graphs (9 and 10) show that our model gets similar psnr values as the state of the art model.

Since out of all the metrics SSIM is able to capture the structural information

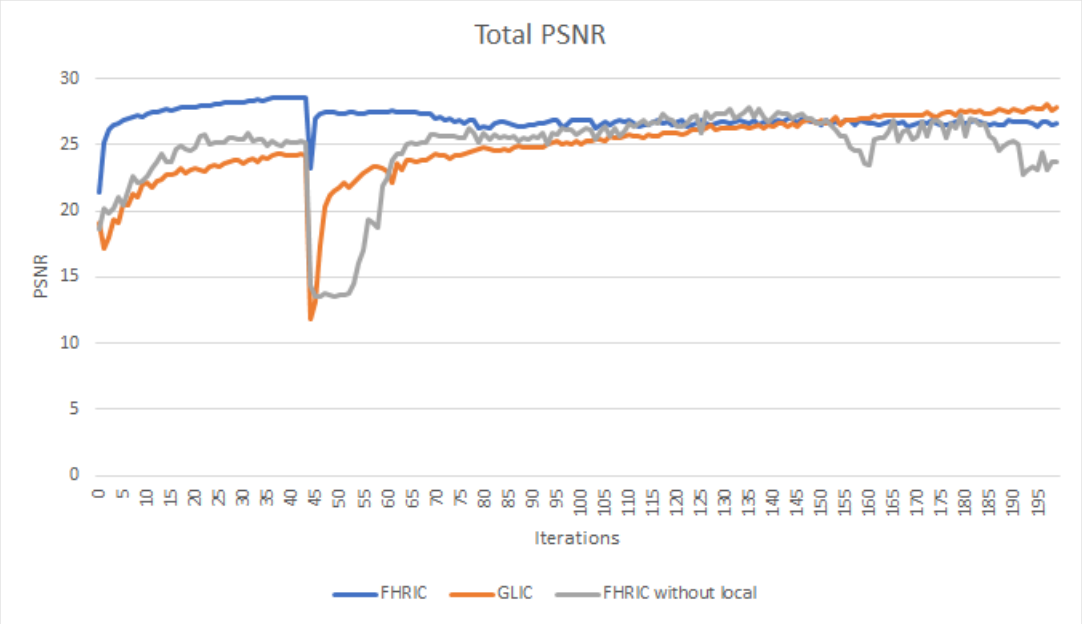

Figure 9: The total psnr of different models over 200 iterations for 30000 images.

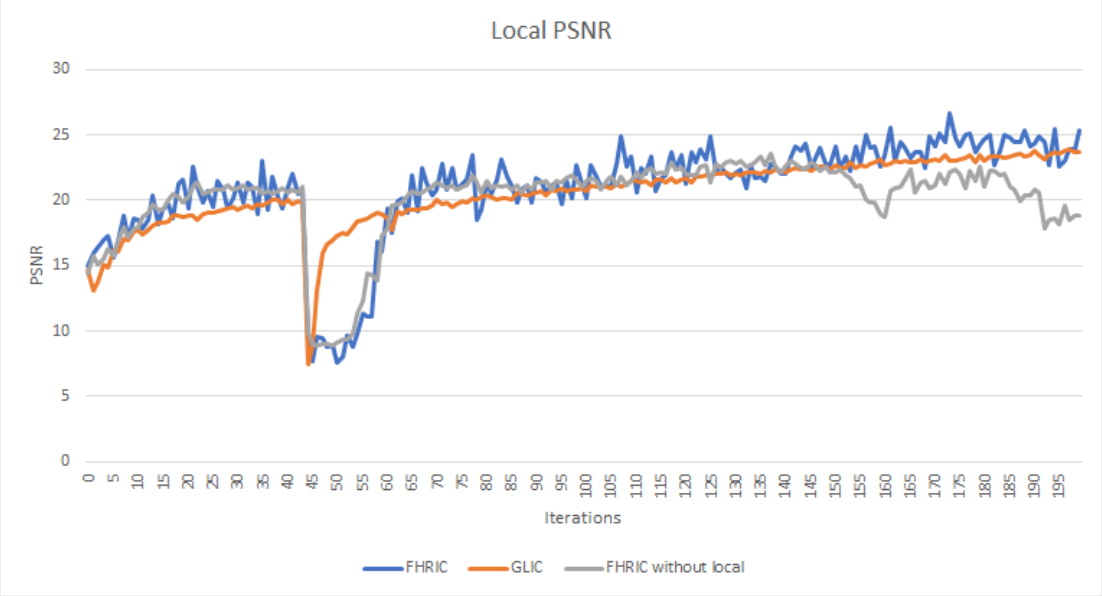

Figure 10: The local psnr of different models over 200 iterations for 30000 images.

of the image, we use only SSIM for evaluating our model and to time it against the state of the art model. The figure 11 shows that the completion isn't proper but the ssim scores are very high. Figure 6 shows the SSIM values of the completed parts

Figure 11: Images completed using our method and GLIC method.

Figure 12: Images completed using our method and our method without the local discriminator.

against iterations for 3 different models ours (FHRIC), globally and locally consistent image completion method (GLIC) and ours without the local discriminator. From the figure 6 it can be seen that our model gives better ssim value of around 0.84 compared to 0.74 of the GLIC model. We also conducted another experiment with fixed size holes in the image and trained them over 2000 images for 300 iterations. We set a baseline of 0.85 to compare both the models. We chose this value because the values reported in the paper [25] (which also deals with face completion) were below 0.85. In the experiment we noted the time it took both the networks to reach 0.85 local SSIM value. The figure 13 shows the graph of the local ssim values over the iterations. Our model FHRIC attains 0.85 local SSIM on the 249th iteration and the model GLIC attains local SSIM on the 286th iteration. It takes our model to get to 249th iteration 1 hour 6 mins and the GLIC model takes 5 hours 22 mins to reach the 286th iteration.

Table 1: Architecture of the completion network.

Type	Kernel	Dilation	Stride	Outputs
conv.	5x5	1	1x1	64
conv.	3x3	1	2x2	128
conv.	3x3	1	1x1	128
conv.	3x3	1	2x2	256
conv.	3x3	1	1x1	256
dilated conv.	3x3	2	1x1	256
dilated conv.	3x3	4	1x1	256
dilated conv.	3x3	8	1x1	256
conv.	3x3	1	1x1	256
deconv.	4x4	1	1/2x1/2	128
conv.	3x3	1	1x1	128
deconv.	4x4	1	1/2x1/2	64
conv.	3x3	1	1x1	32
output	3x3	1	1x1	3

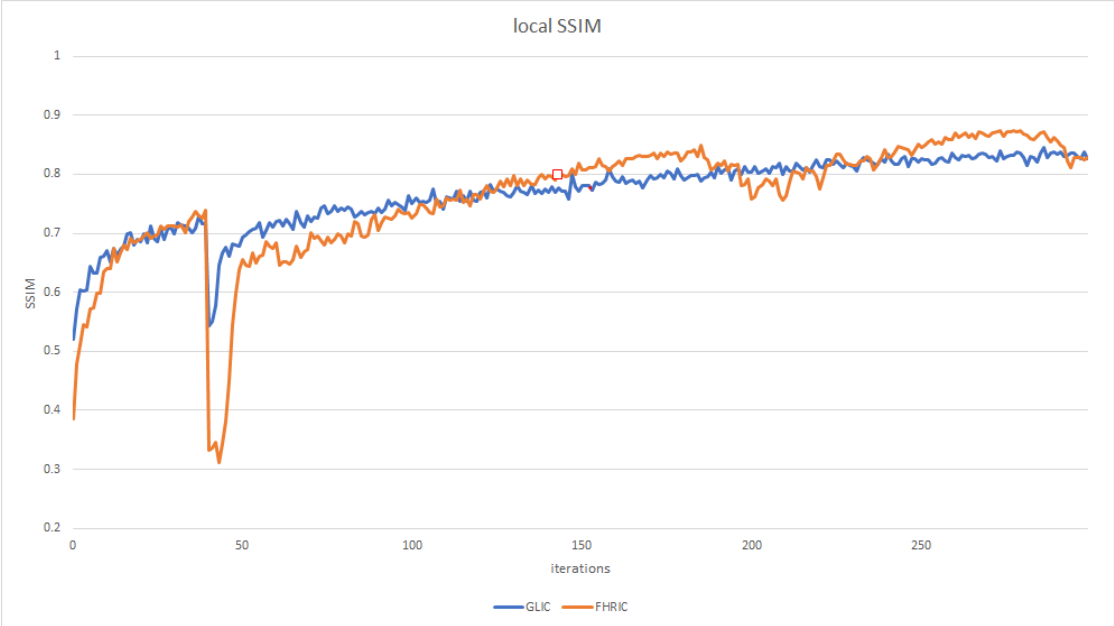

Figure 13: The local ssim of FHRIC and GLIC models over 300 iterations for 2000 images.

4.2 Super Resolution Network

After the completion of the image, the images are then passed through the image transformation network which enhances the resolution of the images from 64x64 to

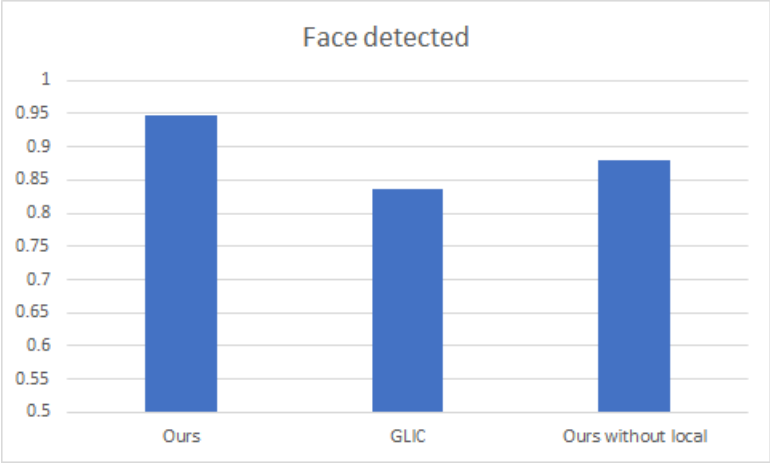

Figure 14: The percentage of faces detected by each individual method

Table 2: Architecture of the super resolution network.

Type	Kernel	strides	filters
conv.	9x9	1	64
16 residual blocks			
conv.	3x3	1	64
conv.	3x3	1	64
2 deconvulational			
deconv.	4x4	1/2x1/2	128
conv.	3x3	1	256
conv.	9x9	1	3

256x256.This enhancement of the completed images are done to generate a higher resolution image. SSIM index works by trying to match the features present in both the images. We went further and use a face detection algorithm implemented in the library [26] to detect faces from the completed image. By this we can evaluate whether the images that are completed resemble a face or not. The figure 14 shows the percentage of faces detected by the algorithm when run against the test images that were completed by the different methods.

The tables 1 and 2 show the final architectures of the completion network and the super resolution network respectively. The table only mentions the different

Table 3: The PSNR and SSIM values on the test sets

Method	PSNR	SSIM
Globally and Locally Consistent	27.73	0.90
FHRIC without Local	23.37	0.88
FHRIC	26.61	0.91

types of convolutional layers used in these networks. Each convolutional layer in the completion network is followed by a batch normalization and a relu activation layer. The completion network is trained for 200 epochs and on 30,000 images, which took around 10 hours on a machine with 16GB ram and a Nvidia Tesla P100 GPU. The completion is done on images with reduced resolution of 64x64 and after completion, the reslution of the images are enhanced to 256x256 using the super resolution network. The super resolution network is trained for 30000 epochs with a batch size of 1. Both the networks are trained using 30,000 images from the celebA dataset.

The table 3 shows the PSNR and SSIM values of the test set when used with the different methods. The figure 15 shows a few selected images that were completed using our method. The values show that our method achieves comparable score or better in completing images, taking a fraction of the time to train the models. The image completion network took 10 hours and 45 minutes and the super-resolution network took 2 hours to enhance the images and transform them from 64x64 to 256x256.

(a) Input image (b) Completed image (c) Hi-res image (d) Original image

Figure 15: A few images that were completed and enhanced with our method and in comparisson to the original image.

CHAPTER 5

Future Work

The model at this state takes in low resolution images with missing parts and then completes it followed by enhancing the image to 256x256. This create the problem of not being able to use the existing parts of the image during the transformation of the image. Future work would include being able to incorporate the residual network within the completion network and be able to train the single network to complete and enhance the resolution of the image.

The proposed model currently looks at a single set of images before completing them. Future work could also include a sequence of images being sent to the model. This would allow the model to learn and predict the movement of the subject and complete missing parts in a video.

CHAPTER 6

Conclusion

In conclusion, the model is able to complete an image and output high resolution images. We used local SSIM to evaluate the quality of the completed images and to time our model. Our model took 1 hour 6 minutes to complete it's training to achieve the same SSIM index as the GLIC model which took 5 hours 22 minutes. Our model is 4.87 times faster than the GLIC model and the memory usage is also significantly less as the images that being dealt with are of lower resolution 64x64 and are later enhance to 256x256 using the super resolution network. The network also works great with 30,000 images whereas other state of the art models [10] worked with half a million images.

LIST OF REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial nets,'' in *Advances in neural information processing systems*, 2014, pp. 2672--2680.

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

[3] J. Johnson, A. Alahi, and L. Fei-Fei, ''Perceptual losses for real-time style transfer and super-resolution,'' in *European conference on computer vision*. Springer, 2016, pp. 694--711.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, ''Image super-resolution using deep convolutional networks,'' *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295--307, 2016.

[5] Z. Cheng, Q. Yang, and B. Sheng, ''Deep colorization,'' in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 415--423.

[6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, ''Photo-realistic single image super-resolution using a generative adversarial network,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681--4690.

[7] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, ''Gradient-based learning applied to document recognition,'' *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278--2324, 1998.

[8] A. Rares, M. J. Reinders, and J. Biemond, ''Edge-based image restoration,'' *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1454--1468, 2005.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, ''Context encoders: Feature learning by inpainting,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536--2544.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, ''Globally and locally consistent image completion,'' *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 107, 2017.

[11] K. Fukushima, ''Neocognitron: A hierarchical neural network capable of visual pattern recognition,'' *Neural networks*, vol. 1, no. 2, pp. 119--130, 1988.

[12] V. Nair and G. E. Hinton, ''Rectified linear units improve restricted boltzmann machines,'' in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807--814.

[13] F. Yu and V. Koltun, ''Multi-scale context aggregation by dilated convolutions,'' *arXiv preprint arXiv:1511.07122*, 2015.

[14] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' *arXiv preprint arXiv:1502.03167*, 2015.

[15] J. Long, E. Shelhamer, and T. Darrell, ''Fully convolutional networks for semantic segmentation,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431--3440.

[16] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770--778.

[17] C. Dong, C. C. Loy, and X. Tang, ''Accelerating the super-resolution convolutional neural network,'' in *European conference on computer vision*. Springer, 2016, pp. 391--407.

[18] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, ''Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874--1883.

[19] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' *arXiv preprint arXiv:1412.6980*, 2014.

[20] J. Bruna, P. Sprechmann, and Y. LeCun, ''Super-resolution with deep convolutional sufficient statistics,'' *arXiv preprint arXiv:1511.05666*, 2015.

[21] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' *arXiv preprint arXiv:1409.1556*, 2014.

[22] Z. Liu, P. Luo, X. Wang, and X. Tang, ''Deep learning face attributes in the wild,'' in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730--3738.

[23] Z. Liu, P. Luo, X. Wang, and X. Tang, ''Deep learning face attributes in the wild,'' in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, ''Image quality assessment: from error visibility to structural similarity,'' *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600--612, 2004.

[25] Y. Li, S. Liu, J. Yang, and M.-H. Yang, ‘‘Generative face completion,’’ in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3911--3919.

[26] D. E. King, ‘‘Dlib-ml: A machine learning toolkit,’’ *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755--1758, 2009.