Improved Semantic Inpainting Architecture Augmented with a Facial Landmark Detector

Mirza Sami
School of Computing, University of Alabama at Birmingham, USA
mtsami@uab.edu

Israt Naiyer
Department of Computer Science and Engineering, Brac University, Bangladesh
israt.naiyer@g.bracu.ac.bd

Ehsanul Khan
Department of Computer Science and Engineering, Brac University, Bangladesh
ehsanul.amin.khan@g.bracu.ac.bd

Jia Uddin
AI and Big Data Department, Endicott College, Woosong University, South Korea
jia.uddin@wsu.ac.kr

Abstract: *This paper presents an augmented method for image completion, particularly for images of human faces by leveraging on deep learning based inpainting techniques. Face completion generally tend to be a daunting task because of the relatively low uniformity of a face attributed to structures like eyes, nose, etc. Here, understanding the top level context is paramount for proper semantic completion. The method presented improves upon existing inpainting techniques that reduce context difference by locating the closest encoding of the damaged image in the latent space of a pre-trained deep generator. However, these existing methods fail to consider key facial structures (eyes, nose, jawline, etc.,) and their respective location to each other. This paper mitigates this by introducing a face landmark detector and a corresponding landmark loss. This landmark loss is added to the construction loss between the damaged and generated image and the adversarial loss of the generative model. The model was trained with the celeb A dataset, tools like pyamg, pillow and the OpenCV library was used for image manipulation and facial landmark detection. There are three main weighted parameters that balance the effect of the three loss functions in this paper, namely context loss, landmark loss and prior loss. Experimental results demonstrate that the added landmark loss attributes to better understanding of top-level context and hence the model can generate more visually appealing in painted images than the existing model.The model obtained average Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PNSR) scores of 0.851 and 33.448 for different orientations of the face and 0.896 and 31.473, respectively, for various types masks.*

Keywords: *Structural image inpainting, generative adversarial networks, facial landmark, synthetic image.*

Received December 27, 2019; accepted February 2, 2021
https://doi.org/10.34028/iajit/19/3/9

1. Introduction

Image completion is the process of taking any damaged or corrupted image and filling up the missing spaces with relevant information. Often times the completion task demands that the missing region synthesized by an algorithm is semantically accurate. Semantic inpainting of corrupted regions thus require a high-level understanding of the surrounding regions [21]. Several algorithms [3, 6] try to fill up missing regions by performing patch matching with known regions of the images; but these solutions are only limited to a low-level understanding of the entire image [15]. Understanding the top level context of the human face is quite difficult given the immense variation that any face has due to structures like eyes, nose, lips, etc. The methods like PatchMatch [3] and the total variation approaches [1] are strictly constrained to the input images for inferring the missing region which makes it completely unsuitable for inpainting the human faces where instances like deducing the patch of a missing nose from the patch containing the eye is quite impossible. A more robust approach for such techniques could be to rely on external databases [7] or looking up from the internet [27]. such databases may help the algorithm to identify the similar looking patches but they fail to maintain proper relevance of the patch with the given (corrupted) images.

Our research is about generating obscured parts in images to give the most realistic result. Synthetic data production is something that is well within our grasp. Hence, the paper relies on cutting edge technologies like Deep Learning to do a significantly better on the immensely challenging task of filling up the corrupted region of a masked image by most semantically accurate data.

Unsupervised deep learning parametric models can learn the feature representation of a given dataset, and once learnt, it can be utilized or inference tasks. The inference is done based on the information and it had learnt from the surrounding regions of the corrupted places. Due to the recent development of powerful deep generative models like the Deep Convolutional Generative Adversarial Network (DCGAN) [23], it is now possible to regenerate large portions of missing regions faithfully. One of the most recent contributions is made on deep learning-based image inpainting by Yeh *et al*. [28]. In that model, the algorithm tries to find the closest match of the masked image in the

latent space of a deep generative model like the DCGAN.

Our proposed model is closest to the work done with semantic image inpainting, however in contrast, our model utilizes a face landmark detector, proposed by Kazemi and Sullivan [11], to identify the locations of key facial structures like eyes, noses, lips and the jawline to ensurea more comprehensive understanding of high-level features rather than just pixel values. We therefore introduced a landmark loss that measures the distance between facial points on the input and the inpainted image. The main contribution of our work is the usage of a pretrained DCGAN [28] for determining the large corrupted portions of the image. A face landmark detector aids in sharpening the generated images which is much more semantically accurate with its surrounding. Thus, our proposed architecture outperformed state of the art semantic model in circumstances where the face was obscured in various angels as well as when a random mask was applied in different positions.

Rest of the paper is organized as follows. Section 2 includes background study. Proposed model is presented in section 3. Section 4 includes experimental setup and result analysis. Finally, concludes the paper in section 5.

2. Background Study

Image inpainting comprises of texture inpainting and semantic inpainting. This paper primarily focuses on the facial image inpainting that strictly maintains the semantic as well as the structure. To achieve that this proposed model has used the deep generative model along with a landmark detector. The following sub-section introduce the technologies behind the proposed architecture.

2.1. Deep Generative Models

Generative Adversarial Networks (GANs) are state of the art deep learning techniques for training parametric models. Deep learning architecture deploying this technique has shown immense success at synthesizing images that are both high quality and visually appealing [7, 22]. At the core of this framework are a pair of neural networks. A generator- *G*, and a discriminator- *D*. The generator has the objective of mapping a random vector *z*, which can be sampled from a prior distribution P_z, to the image space. On the other hand the *D* has the objective of finding out the likelihood of the generated image being from the image dataset. Hence, *G* aims to generate realistic images, while *D* takes in the role of an adversary, always trying to discriminate between the image generated from *G* and the real image that originated from the dataset's distribution P_{data}. The *G* and *D* networks can be trained with their combined objective function as presented in the following Equation (1).

$$\underbrace{\min}_{G}\ \underbrace{\max}_{D} V(G,D) = \mathbb{E}_{h\sim p_{data}(h)}\big[\log\big(D(h)\big)\big] + \mathbb{E}_{z\sim p_z(z)}\big[\log\big(1-D(G(z))\big)\big] \quad (1)$$

ANs have become extremely popular for fulfilling the image inpainting task that is semantically coherent since it understands the context of the image. Further, GANs produce sharper images compare to other models like Variational Auto Encoder (VAEs). In the proposed model we have used a DCGAN. The DCGAN architecture gains it ability to generate images from the Generative Adversarial training method in conjunction with the Convolutional Neural Network (CNN) architecture. A significant improvement to this effort came with the development of the Laplacian Pyramid of GAN (LAPGAN) [21] which worked on an iteratively upscaling low resolution generated images.

The authors of the DCGAN architecture made three significant modifications to the architecture. First, all convolutional net [6, 8] which gets rid of maxpooling and strided convolutions and gives the network the capability to learn its own spatial down sampling. It allows a DCGAN to learn its own spatial up sampling and discriminator [10]. It replaces the deterministic spatial pooling functions (such as maxpooling) with strided convolutions, allowing the network to learn its own spatial down sampling. In the proposed model, DCGAN is used as a generator. Second change is in the decision to get rid of fully connected layers presented over the convolutional features. It is important because the global mean pooling can yield greater stability for the model but also can reduce the speed of convergence of the network. In the GANs, first layer is a fully connected layer that takes in the random noise, applies matrix multiplication and reshapes the data into a four dimensional tenor from this point onwards the convolution part begins. In the discriminator side, the final convolution layer is flattened and then put through as a single sigmoid output. Third changes are the use of Batch Normalization [28] which makes the learning more stable due to normalizing the input with zero mean and unit variance, before being fed to every node in the neural network.

2.2. Facial Landmark Detector

The model of landmark detection used in this paper was proposed by Kazemi and Sullivan [11] and earlier it was used in the multiple layers of regressors. To understand how it works let's consider xi $\in\mathbb{R}^2$, which will be the *x, y* coordinates of the *i*th landmark point in a given input image *I*. The vector S = $(x^T_1, x^T_2, \ldots x^T_p,)^T \in \mathbb{R}^{2p}$, where T is a transpose, marks all the *p* facial landmark points in the image *I*.

$$\mathcal{L}(\vec{x},\hat{\vec{x}}) = \sum_{l=0}^{L} w_l E_l \quad (2)$$

$$\hat{S}^{(t+1)} = \hat{S}^t + r_t\big(I,\hat{S}^{(t)}\big) \quad (3)$$

Furthermore, $\hat{S}^{(t)}$represents the current estimate of S. Every regressor in the cascade can be represented by $r_t(...)$. The regressors will take in *I* and the current estimate $\hat{S}^{(t)}$ and predict and update the vector, which will then be added to $\hat{S}^{(t)}$ to get the improved estimate $\hat{S}^{(t+1)}$. The predictions made by the regressor r_t are dependent on attributes such as pixel intensity values of the input image *I* and it is indexed relative to $\hat{S}^{(t)}$. A result of this process is the formation of geometric invariances that can give the assurance with the cascade progresses, the correct locations on the face are getting semantically indexed. The range of outputs expanded by the cascade is only guaranteed to be in a linear subspace of training data if the first estimate $\hat{S}^{(0)}$ is a member of the same space. Thus, no further constraints on the predictions has to be applied. This means that the initial shape can be taken to be the mean shape centered around the training data. Furthermore, it can be scaled with the bounding box of any regular face detector. The regressors are trained according to the techniques in [17] which utilizes the gradient tree boosting algorithms [20] and calculates the loss as the sum of square.

2.3. Back Propagation to Input

The focal issue that back-propagation works with is the assessment of the impact of a parameter on a function whose calculation includes a few elementary steps. The chain rule is the answer to this issue; however back-propagation exploits the specific type of the capacities utilized at each progression (or layer) to give an exquisite and local system [19]. In this section, we will discuss some past research that used 'back propagation to input' and from the equation used by them, we will derive our process of back propagating. In order to create another texture based on a given picture, [4] suggested that Gradient descent is utilized from a white noise image to discover another picture that coordinates the Gram-matrix portrayal of the first picture. By minimizing the mean-squared distance between the entries of the gram matrix of the original image and gram matrix of the generated image, the optimization is achieved.

Let $\vec{x}$ and $\hat{\vec{x}}$ be the original image and the image that is generated, and G^l and $\hat{G}^l$ their respective Gram-matrix representations in layer *l*. The contribution of layer *l* to the total loss is that where w_l are weighting factors of the contribution of each layer to the total loss.

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - \hat{G}_{ij}^l)^2 \tag{4}$$

The derivative of E_l with respect to the activations in layer l can be computed analytically, as shown in Equation (3).

The gradients of E_l, and thus the gradient of $L(\vec{x};\hat{\vec{x}})$ as explained in Equation (4), with respect to the pixels $\hat{\vec{x}}$ can be promptly figured utilizing standard error back-propagation [18]. The gradient $\partial L / \partial \hat{\vec{x}}$ can be utilized as input for some numerical advancement procedure. Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [14] is utilized is our work, which appeared a sensible decision for the high-dimensional enhancement issue at hand. The whole technique depends primarily on the standard forward-backward pass that is utilized to prepare the convolutional network. In this manner, despite the expansive intricacy of the model, texture generation is possible in a sensible time utilizing GPUs and performance-enhanced toolboxes for training deep neural systems [10]. As suggested by [26], this segment will depict a system for picturing the class models, learnt by the picture characterization ConvNets. Given an educated grouping ConvNet and a class of intrigue, the perception technique comprises in numerically creating a picture [24], which is illustrative of the class as far as the ConvNet class scoring model. More formally, let $S_c(I)$ be the score of the class c, processed by the arrangement layer of the ConvNet for a picture *I*. The L2-regularised image can be found in Equation (5), such that the score S_c is high with:

$$\arg\max S_c(I) - \lambda \|I\|_2^2 \tag{5}$$

Where λ is the regularization parameter. The back-propagation is utilized to optimize the layer weights to identify the ConvNet training strategy. The thing that matters is that for our situation the optimization is performed as for the input, while the weights are settled to those discovered amid the training. The optimization as initialized with the zero picture (for our situation, the ConvNet was prepared on the zero-focused picture information), and afterward included the preparation set mean picture to the outcome. It ought to be noticed that we utilized the (unformulated) class scores S_c, rather than the class posteriors, returned by the soft-max layer: $Pc = \frac{\exp Sc}{\sum_c \exp Sc}$. The reason why it was done is because by minimizing the scores of different classes, the boost of the class posterior can be accomplished. In this manner, we improve S_c to guarantee that the optimization focuses just on the class being referred to *c*.

$$\frac{\partial E_l}{\partial \hat{F}_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2} ((\hat{F}^l)^T (G^l - \hat{G}^l))_{ji} & if\ \hat{F}_{ij}^l > 0 \\ 0 & if\ \hat{F}_{ij}^l < 0 \end{cases} \tag{6}$$

3. Proposed Model

This paper is keenly focused on designing a model that works specifically for the human faces and it can generate synthetic images for filling in holes/noise in the input image. As depicted in Figure 1, the proposed model includes the facial landmark detector considering three different loses over the state-of-art

model presented in [28]. This model utilizes a pre-trained DCGAN [23] and the DCGAN architecture is trained using the Celeb A image dataset [16]. The dataset contains thousands human faces in different angles and lighting conditions. A trained DCGAN can sample random noise from a normal distribution, apply its newly learnt weights and biases to the noise data and convert the noise into an image. This in turn, gives the DCGAN the ability to sample anywhere from the normal distribution and generate completely new images of human faces (based on the images it had learnt from the dataset). This ability is crucial for the task at hand. Once a masked input image is given to the model as input, it calls the pre-trained DCGAN to randomly generate a batch of 64 images which has been sampled as '*z*' from the normal distribution. The generated images are then applied with the same mask used for the input image. Initially, the generated images look nothing like the input image, thus incurring a high loss. The loss is calculated based on three parameters. First, the *L1* distance between the masked input image and generated images. *L1* Loss Function is used to minimize the error which is the sum of the all the absolute differences between the true value and the predicted value. Second, the distance of the 68 landmark facial points and third, the adversarial loss [5] from the discriminator of the DCGAN. The total loss calculated is then used to find the gradient in terms of the input noise *z* (random samplings form the normal distribution) that is feed to the generator of the DCGAN. This gradient is used to back-propagate to the latent input space and traverse across it.

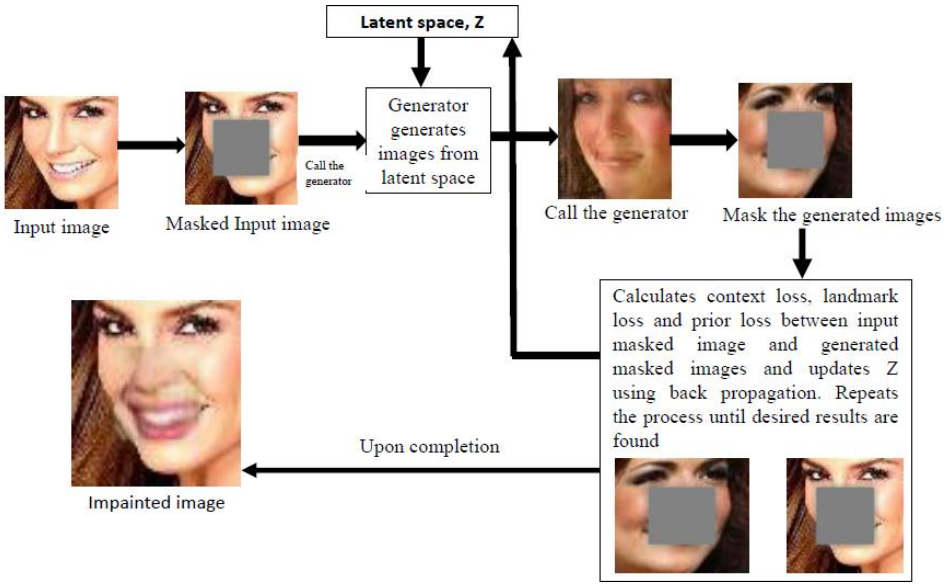


Figure 1. Block diagram of the proposed architecture.

In order to inpaint significant portions of missing data both the generator and the discriminator will have to train with undamaged, normal data. The GAN based generator for the purpose is the aforementioned DCGAN. Once training is completed, the generator *G* of the DCGAN can draw points *z* from a uniform distribution P_z and synthesize images that looks similar to the samples from the dataset distribution P_{data}. Although GANs are capable of generating new images, they alone are not enough for the inpainting task as the produced images are random and irrelevant to the corrupted image. According to the model in [27], it is known that given a *G*, which has an efficient representation of the data, must not have a representation of the foreign (corrupted) image in its encoded latent space representation P_{data}.

Thus, the goal should be to locate the encoding $\hat{z}$ that has the highest similarity to the damaged image at hand, all the while maintaining a constraint to the latent space. As the algorithm traverses through the latent space, the search for the closest encoding gets optimized with each step of finding $\hat{z}$.

Once $\hat{z}$ is found, the generator *G* can be called upon to convert the encoding of the latent space to an image. Concretely, Yeh *et al*. [28] has formulated the procedure of locating $\hat{z}$ as an optimization problem. Thus, the variable *y* represent the damaged image to recover and a binary mask which is the same size as the image and denoted by *M* is used to mark the missing parts. Using the notations, the nearest encoding $\hat{z}$ can be represented as:

$$\hat{z} = \arg\min\{\mathcal{L}_c(z|y,M) + \mathcal{L}_p(z)\} \quad (7)$$

Where $\mathcal{L}_c$ denotes the context loss which purposes is to constraint the generated image provided the input damaged image *y* and the mask *M*; $\mathcal{L}_p$ denotes the prior loss that purposes to penalizes the images that does not look visually appealing.

3.1. Weighted Context Loss

Yeh *et al.* [28] used a context loss to frame such information by calculating a *L2* norm between the generated image *G(z)* and the uncorrupted portion of the input image.

L2 Loss Function is used to minimize the error which is the sum of the all the squared differences between the true value and the predicted value. This method however, is not very efficient as it treats every pixel equally. Pixels that are far away from the missing region should naturally be treated with lesser precedence compared to that of pixels which are closer to the missing region. As a result, the importance of an uncorrupted pixel is directly correlated with the number corrupted pixels surrounding it. This idea can be better articulated with the Equation (8).

$$W_i = \begin{cases} \sum_{j \in N(i)} \frac{(1-M_j)}{|N(i)|} & if\ M_j \neq 0 \\ 0 & if\ M_j = 0 \end{cases} \quad (8)$$

Where *i* is the pixel index, W_i denotes the importance weight at pixel location *i*, *N(i)* refers to the set of neighbors of pixel *i* in a local window, and *i*, |N(*i*)| denotes the cardinality of *N(i)*. Finally, the contextual loss $\mathcal{L}_c$ in Equation (9) can be defined as a weighted *L1*-norm difference betweenthe recovered image and the uncorrupted portions. Here, $\odot$ denotes element wise multiplication.

$$\mathcal{L}_c(z|y, M) = \|W \odot (G(z) - y)\|_1 \quad (9)$$

3.2. Land Mark Loss

The landmark loss can be calculated because of the dlib landmark detector [12], which takes in an input image and finds the (*x*,*y*) coordinates of 68 points on the face.

$$\mathcal{L}_{d,x}(x) = \begin{cases} 0, & |x - \hat{x}| \leq 3 \\ 0.5, & |x - \hat{x}| > 3 \end{cases} \quad (10)$$

$$\mathcal{L}_{d,y}(y) = \begin{cases} 0, & |y - \hat{y}| \leq 3 \\ 0.5, & |y - \hat{y}| > 3 \end{cases} \quad (11)$$

This detector as applied on normal images and masked images and it works well on both. For calculating loss, the (x, y) coordinates for the damaged image has to be obtained, this becomes the ground truth. Then the DCGAN is asked to generate a batch of 64 images. These images are then applied with the same mask and fed to the landmark detector in order to obtain their (x,y) coordinates. For every facial point of the generated image, if the absolute value distance of its x-coordinate, surpasses the corresponding x-coordinate of the same facial point of the ground image by three points, the land mark loss $\mathscr{L}_{d,y}$ is increased by 0.5. This particular value was selected for measuring the loss on a distance of three because this particular parameter yielded the best results in the experimentation. Similarly, another 0.5 is added to $\mathscr{L}_{d,y}$ for the y-coordinated for a distance greater than 3 in either direction. Finally, the total loss is added for all 68 points for all 64 images in the batch and that yields the final landmark loss $\mathcal{L}_d$ for that batch.

3.3. Prior Loss

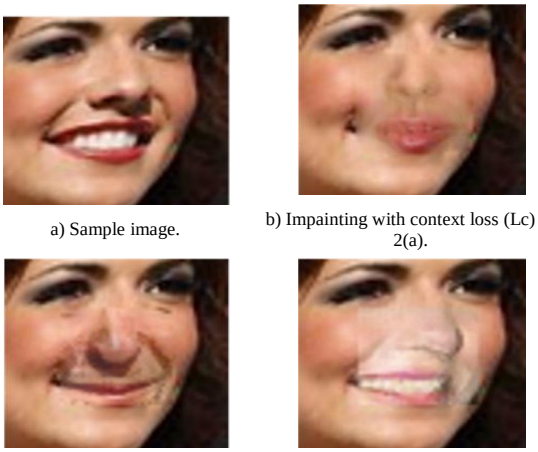
The prior loss refers to the punishment that is applied to the system based on high-level image feature representations instead of pixel wise differences. The prior loss can be measured from the pre-trained image classifier (the discriminator, *D*) and is used to ensure that the recovered image is highly close to the training set. In GANs, the discriminator, *D*, is trained to differentiate generated images from real images. Therefore, the prior loss is very similar to the GAN loss for training the discriminator *D*, i.e.

$$\mathcal{L}_p(z) = \lambda \log\left(1 - D(G(z))\right) \quad (12)$$

Here, λ is a parameter to balance between the two losses; *z* is updated to fool *D* and make the corresponding generated image more realistic. Without $\mathcal{L}_p$ the mapping of *y* to *z* would have been highly difficult to achieve.

The usage of prior loss, context loss and the landmark loss, made it possible to achieve a desirable mapping of the corrupted image from the latent space representation which is denoted by $\hat{z}$. After generating $G(\hat{z})$, the inpainting result can be easily obtained by overlaying the uncorrupted pixels from the input. Poisson Blending [22] to sharpen the final results is applied in [28] but it is not utilized in the proposed model. Thus, the final loss, *L*, is used in the presented model calculated presented in Equation (13). Figure 2 illustrates who different types of loss work on a sample image.

$$L = \lambda_1 \mathcal{L}_c + \lambda_2 \mathcal{L}_d + \lambda_3 \mathcal{L}_p \quad (13)$$



a) Sample image. b) Impainting with context loss (Lc) of 2(a).

c) Impainting with context and landmark loss (L_c,L_d) of 2(a). d) Impainting with context, landmark, and prior loss (L_c,L_d, L_p) of 2(a).

Figure 2. Inpainting on a sample image with context, prior and landmark loss.

3.4. Training

The entire operation of the model begins by taking in a normal image. Then a binary mask can be applied to it.

The masks can be a centered box, left or right aligned or completely randomly generated.

Once the image been processed, work begind with the DCGAN. The generative model, *G*, takes a random 100 dimensional vector sampled from a uniform distribution and generates a 64x64x3 image. On the other side of the DCGAN generator is the discriminator model, *D*, which is in reverse order. The input layer of the generator is an image of dimension 64x64x3, it is then followed by a number of convolution layers. In the layers the image dimension is reduced to half, and the channel size is doubled the size of the previous layer. The output layer has the softmax activation function. Then, the training of the DCGAN model is done according to [23] and optimizer is used presented in [13]. This model used λ =0.003 for all the testing purpose. Once the generator is trained, output a batch of 64 images is extracted. The entire batch of image is then masked with the same mask used for the ground truth image.

Once masking is complete, the weighted context loss $\mathcal{L}_c$ as per Equation (9) between the ground truth image and all the masked generated image in the batch is measured. Then the 68 facial point coordinates for the ground image is first obtained with the landmark detector. This model then find the facial points for each image in the batch and with that calculate the landmark loss $\mathcal{L}_d$ with Equations (10, 11).

The prior loss, $\mathcal{L}_p$ is calculated as Per Equation (12). Finally, all three losses are calculated and the total loss L is found according to Equation (13). This loss can now be used to back propagate to the input latent dimension z for optimization (as depicted in Figure 1). In the inpainting stage, $\hat{z}$ in the latent space z has to be found using back-propagation. This model used Adam Optimizer for optimization and restricted z to [-1; 1] for all iterations, in order to achieve the best possible result. The model was ran for 1000 iterations using the tensor flow [2] framework on a NVDIA 1050Ti GPU enabled machine.

4. Experimental Setup and Results Analysis

The following sub sections present the details about the dataset used for validating the proposed model along with experimental results in various considerations.

4.1. Datasets and Masks

The proposed model is primarily evaluated on the CelebFaces Attributes Dataset (CelebA) [16]. This dataset is composed of 202,599 face images with coarse alignment. Some samples of images from the dataset are shown in Figure 3.

For experimentation purpose, around 250 images from the dataset is sampled and applied with the masking for semantic hole-filling. All images used for training and hole-filling are initially cropped at the centre (focused primarily at the face) to 64x64 (depicted in Figure 3). The images capture face at different angles, lightning condition and a wide range of skin tone. Once the images are processed, they are ready to be fed into the DCGAN model architecture. The DCGAN is trained and tested according to the instruction previously mentioned in this paper. This model has mostly been experimented with faces at different angles. Additionally, it is also tested with different types of masks, the main variant of the masks used is the central square mask. The mask applied on the testing images would cover as much as 25% of the face, yet the model is still capable of generating the missing parts of the face.

Figure 3. The celeba dataset.

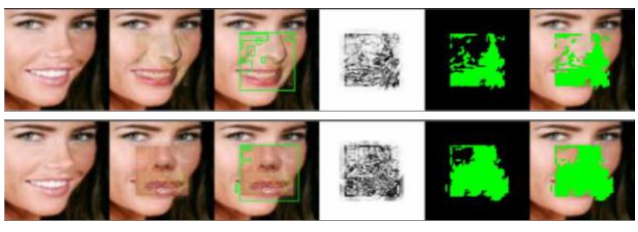
4.2. Dataset for Face Parsing

Although the primary dataset for the model is the CelebA dataset, it is not very effective for training the facial landmark detector that is used in this model. Instead, this model used a landmark detector that is trained on the iBUG 300-W dataset [24]. This dataset has numerous facial images and each face has 68 segment labels that considers all major facial components like eyes, nose, lips, etc. Since this model used a pre-trained landmark detector directly from dlib's library, it does not undergo any changes during the training and only does detection.

4.3. Qualitative Comparisons

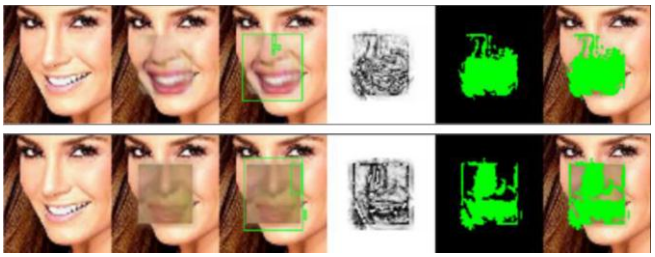
The comparisons that follow, intend to showcase the inpainting ability of the proposed model.

Since the primary focus of this model is to perform well in situations where the structural integrity of the face must be preserved, the images (Figure 4) used for testing depict faces that are oriented at an angle away from the central view.



a) Original. b) Our model. c) Different regions. d) SSIM difference. e) Black mask. f) Overlaid.

Figure 4. SSIM difference visualization with a cutting-edge(mod.1).



a) Original. b) Our model. c) Different regions. d) SSIM difference e) Black mask. f) Overlaid.

Figure 5. SSIM difference visualization with a cutting-edge(mod.2).

This feature is important for the testing purpose because such images test the model's ablity to truly consider the structure of the face when inpainting, rather than mindlessly overlapping any image produced by the generator. A good measure for structural consistencey is the Structural Similarity Index (SSIM) [25] that finds the degradation of quality of an image after processing. The SSIM metric can be used to measure the difference between the original image and its inpainted version.

To visualize this, please refer to (Figure 4), the first image on the top row, depicts a person whose face is oriented towards the right from the viewer's perspective, labelled as Original. The next image, depicts the end result after the Original image has been masked and then inpainted by the proposed model. The difference value between the images is measured using SSIM [25]. This value is then thresholded using OpenCV, and then OpenCV rectangles are drawn around the regions of difference as depicted by the third image, labelled Different Regions. Within the OpenCV rectangles, countours of difference is calculated and drawn as demonstrated by the image labelled SSIM Difference. The white region signifies no differences, while the gray areas are drawn to demonstrate the contours of the differences identified. In order to visual the difference more distinctly, the contours are drawn over a black mask and then finally overliad on top of the original image as depicted by the image labelled Overlaid. The same process was repeated with the (inpanting) output from a state of the art model, Semantic Inpainting (SI) [28] (labelled Other Model) and depicted in the bottom row of images of (Figure 4). If the Overlaid image from the proposed model is compared to the Other Model, it can be observed that the inpainted image from the proposed model has less difference with the *Original* image. It is especially noticiable around the bridge of the nose and the upper lips, where the proposed model is better at mataining the structural consistence of facial landmarks such as the nose and lips, with respcet to the overall orientation of the face.

The same analysis is repeated again, this time with an image where the face is oriented to the left from the perspective of the viewer (presented in Figure 5). Again, if the Overlaid image from the proposed model is compared to the Other Model, a similar trait can be observed.

The inpainted image from the proposed model has less difference with the Original image, particularly around the bridge of the nose.

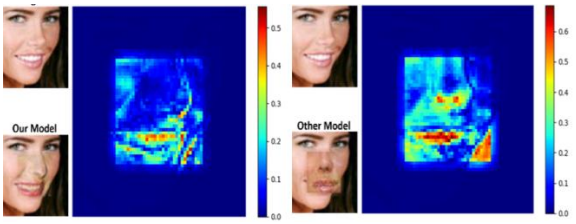


Figure 6. Heat map comparison of model 1 depicted in (Figure 4).

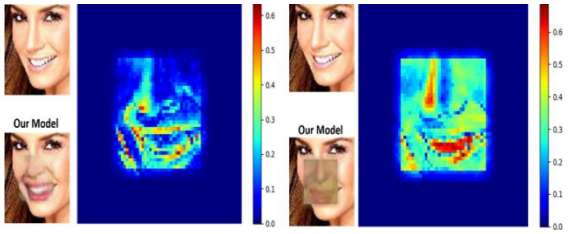


Figure 7. Heat map Comparison of a model (depicted in Figure 5).

The inpainted image from the Other Model shows clear distinction around the nose and the lips from the Original Image. However, the Overlaid image from the proposed model seems to show a greater difference around the lips, athough visually the image produced by the proposed model seems to be far more accurate. This inconsistency may be atttributed to other variables such as luminance and contrast that are taken into account while measuring SSIM. Furthermore, the analysis above finds differences in a binary manner, drawing contours as long as there is a difference that exceeds the defined threshold, it fails to give a sense as to what degree does the twoimages iffer by.

To alleviate this issue, a heat map of the images is drawn. A heatmap can provide a sense of scale for the diffence between the images using the color spectrum [29], where blue represents simlarity and red represents difference. If the images in (Figure 6) is observed, it can be noted that the image produced by the proposed model is generally more blue, wherease the image from the other model is generally more cyan. Futhermore, the inconsistency of the orientation of the nose with respect to the orientation of the face can be easily distinguished in the image produced by the Other Model. The difference is apparent around the nose and the lips, where the Other Model has patches of areas which are green or even red, signifying considerable difference. The image from the proposed model also has some patches of red, but these are around the teeth, which is not intended to be explicitly taken into consideration by the proposed model.

Similarly, another heat map is also presented in (Figure 7) and it uses the same images that are analysed in (Figure 5). The heat map helps clear the disparity between visual observation and the SSIM

differnce that is observed before. If (Figure 6) is observed, it can be easily noted that the inpainted image from the Other model is significantly more different than the Original image. The heat map image from the Other Model displays green and yellow shades all around the inpainted region, with red patches along the nose and lips. The heat map for the proposed model on the other hand does show some difference with the Original image, however that difference is signigficantly lower as compared to the Other model. This further reinforces the notion that the proposed model is more effective at maintaining the orientation of facial landmarks with respect to the oreintation of the face.

4.4. Quantitative Comparisons

This section of the paper will provide a more concrete measure for the differences in inpainting ability between the proposed model and the state-of-the-art model. This paper would like to clarify again that this model is not made to recreate the ground truth image, rather it tries to fill in the damaged place with the most similar possible content. To give a more concrete understanding of the performance of this model, this paper will be comparing it with the cutting-edge model, Semantic Inpainting (SI) [28]. The real images from the dataset are used as ground truth reference.

Table 1. SSIM score for different orientation of face.

Face Orientation	SI (Lc+Lp)	Ours (Lc+Ld+Lp)
Centre	0.865	0.888
Left	0.819	0.837
Right	0.815	0.848
Random	0.815	0.829

Table 2. PSNR score for different orientation of face.

Face Orientation	SI (Lc+Lp)	Ours (Lc+Ld+Lp)
Centre	33.281	33.463
Left	32.616	33.218
Right	32.863	33.373
Random	32.971	33.738

Tables 1 and 2 provide the results on the CelebA dataset for two tests - PSNR and SSIM [25]. Compared to the model presented, the PSNR and SSIM values as depicted in Tables 1 and 2, of the latest models are generally similar or slightly better most cases. The main exception is for the random masks, where this model and SI seem to do much better. The prime reason behind our model doing better is the addition of a Landmark loss (Ld). We did an ablation study where we removed the landmark loss(Ld) from our proposed architecture and the results werenot as good as the ones presented in the Tables (1, 2, 3, and 4).

Table 3. SSIM score for different type of masks.

Mask Orientation	SI (Lc+Lp)	Ours (Lc+Ld+Lp)
Centre	0.876	0.889
Left	0.880	0.885
Right	0.901	0.911
Random	0.879	0.898

Table 4. PSNR score for different type of masks.

Mask Orientation	SI (Lc+Lp)	Ours (Lc+Ld+Lp)
Centre	31.981	32.644
Left	31.785	30.491
Right	29.523	30.981
Random	30.156	31.778

Tables 3 and 4 show the results of proposed model for four different mask orientations considering the SSIM and PSNR. It has been observed that the proposed model exhibits better performance then state of art model with higher SSIM and PSNR for various considered scenarios.

However, quantitative results do not represent well the real performance of the method especially when the ground truth image can be variable depending on the dataset. Similar observations can be noted in [9s], where better visual results corresponds to lower PSNR values. Nevertheless, for random holes, this method performed better in most cases both in PSNR and SSIM scoring. The presented method could outperform the state of art model because the undamaged pixels are spread more widely throughout the image, and PSNR is a much more meaningful scoring technique in this case.

5. Conclusions

This paper presents a deep generative network for structural and semantically coherent image inpainting. The system comprises of a pre-trained GAN, which has two neural networks, a generator and a discriminator, a parser network for landmark detection and adversarial loss for generative model. The proposed model is comparatively more effective at orchestrating semantically legitimate and realistic images for the key missing facial parts from arbitrary clamor. This strategy enhances existing inpainting methods that reduce context difference by finding the nearest encoding of the obscured picture in the latent space of a pre-trained GAN. Furthermore, it is reasoned that the additional landmark loss credits to better comprehension of best dimension setting and thus more outwardly engaging inpainted pictures. This system can be improved in the future by building a better parser network algorithm for more effective and faster landmark detection. Overall, our model scored consistently proved to score higher on the SSIM and Signal-to-Noise Ratio (PNSR) score metrics as opposed to the state of art model. An average SSIM and PNSR score of 0.851 and 33.448 was obtained while the face was at various angles, and a score of 0.897 and 31.473 with various types of masks.

Acknowledgement

This research is funded by Woosong University Academic Research in 2022.

References

[1] Afonso M., Bioucas-Dias J., and Figueiredo M., "An Augmented Lagrangian Approach to The Constrained Optimization Formulation of İmaging İnverse Problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681-695, 2011.

[2] Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M., Kudlur M., Levenberg J., Monga R., Moore S., Murray D., Steiner B., Tucker P., Vasudevan V., Warden P., Wicke M., Yu Y., and Zheng X., "Tensorflow: a System for Large-Scale Machine Learning," *in Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation*, Savannah, pp. 265-283, 2016.

[3] Barnes C., Shechtman E., Finkelstein A., and Goldman D., "Patchmatch: A Randomized Correspondence Algorithm for Structural İmage Editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 24, 2009.

[4] Gatys L., Ecker A., and Bethge M., "Texture Synthesis Using Convolutional Neural Networks," *in Proceedings of the 28th International Conference on Neural Information Processing Systems*, Montreal Canada, pp. 262-270, 2015.

[5] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., and Bengio Y., "Generative Adversarial Nets," *in Processing of Advances in Neural İnformation Processing Systems*, Montréal CANADA, pp. 1-9, 2014.

[6] Huang J., Kang S., Ahuja N., and Kopf J., "Image Completion Using Planar Structure Guidance," *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 1-10, 2014.

[7] Hays J. and Efros A., "Scene Completion Using Millions of Photographs," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 4, 2007.

[8] Islam, N., Sulaiman N., Al Farid F., Uddin J., Alyami S., Rashid M., Majeed A., and Moni M., "Diagnosis of Hearing Deficiency Using EEG based AEP Signals: CWT and Improved-VGG16 Pipeline," *PeerJ Computer Science*, vol. 7, pp. e638, 2021.

[9] Johnson J., Alahi A., and Fei-Fei L., "Perceptual Losses for Real-Time Style Transfer And Super-Resolution," *in Proceedings of European Conference on Computer Vision*, Amsterdam, pp. 694-711, 2016.

[10] Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., and Darrell T., "Caffe: Convolutional Architecture for Fast Feature Embedding," *in Proceedings of the ACM International Conference on Multimedia*, Orlando Florida, pp. 675-678, 2014.

[11] Kazemi V. and Sullivan J., "One Millisecond Face Alignment with an Ensemble of Regression Trees," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, pp. 1867-1874, 2014.

[12] King D., "Dlib-ml: A Machine Learning Toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755-1758, 2009.

[13] Kingma D. and Ba J., "Adam: A Method for Stochastic Optimization," *in Proceedings of the 3rd International Conference for Learning Representations*, San Diego, 2014.

[14] Khondaker A., Khandaker A., and Uddin J., "Computer Vision-based Early Fire Detection Using Enhanced Chromatic Segmentation and Optical Flow Analysis Technique," *The International Arab Journal of Information Technology*, vol. 17, no. 6, pp. 947-953, 2020.

[15] Li Y., Liu S., Yang J., and Yang M., "Generative Face Completion," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, pp. 3911-3919, 2017.

[16] Liu Z., Luo P., Wang X., and Tang X., "Deep Learning Face Attributes in the Wild," *in Proceedings of the IEEE International Conference on Computer Vision*, Santiago, pp. 3730-3738, 2015.

[17] Le V., Brandt J., Lin Z., Bourdev L., and Huang T., "Interactive Facial Feature Localization," *in Proceedings European Conference on Computer Vision*, Florence, pp. 679-692, 2012.

[18] LeCun Y., Bottou L., Orr G., and M¨uller K., Efficient backprop," in Neural Networks: Tricks of the Trade, Springer, 2012.

[19] Le Cun Y., "A Theoretical Framework for Back-Propagation," *in Proceedings of the Connectionist Models Summer School*, Pittsburg, pp. 21-28, 1988.

[20] Meir R. and Rätsch G., *Advanced Lectures on Machine Learning*, Springer, 2003.

[21] Pathak D., Kr¨ahenb¨uhl P., Donahue J., Darrell T., and Efros A., "Context Encoders: Feature Learning by İnpainting," *in Proceedings of IEEE Confrence on CVPR*, Las Vegas, pp. 2536-2544, 2016.

[22] P´erez P., Gangnet M., and Blake A., "Poisson İmage Editing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 313-318, 2003.

[23] Radford A., Metz L., and Chintala S., Unsupervised Representation Learning with Deep Convolutional Generative Adversary Networks," *in Proceeding of 4th International Conference on Learning Representations*, San Juan, pp. 1-15, 2016.

[24] Sagonas C., Tzimiropoulos G., Zafeiriou S., and Pantic M., "300 Faces in-The-Wild Challenge: the First Facial Landmark Localization

Challenge," *in Proceedings of the IEEE International Conference on Computer Vision Workshops*, Sydney, pp. 397-403, 2013.

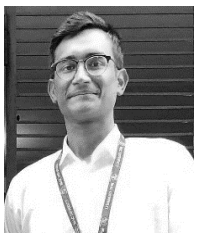
[25] Shoron S., Islam M., Uddin J., Shon D., Im K., Park J., Lim D., Jang B., and Kim J., "A Watermarking Technique for Biomedical Images," *Electronics*, vol. 8, no. 9, pp. 975, 2019.

[26] Simonyan K., Vedaldi A., and Zisserman A., "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," arXiv Preprint arXiv: 1312.6034, 2013.

[27] Whyte O., Sivic J., and Zisserman A., "Get out of my Picture! İnternet-Based İnpainting," *in Proceedings of the British Machine Vision Conference*, London, pp. 1-11, 2009.

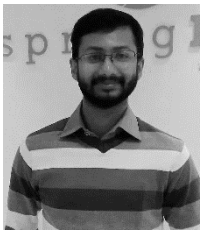
[28] Yeh A., Chen C., Lim T., Schwing A., Hasegawa-Johnson M., and Do M., "Semantic İmage İnpainting with Deep Generative Models," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, pp. 5485-5493, 2017.

[29] Zhu C., Byrd R., Lu P., and Nocedal J., "Algorithm 778: L-BFGS-B: FORTRAN Subroutines for Large-Scale Bound-Constrained Optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550-560, 1997.



Mirza Sami completed B.Sc. Degree in Computer Science and Engineering (CSE) from Brac University (BracU), Bangladesh in 2018. Currently, he is a PhD student at University of Alabama at Birmingham, USA. His research interests are computer vision and artificial intelligence.

Israt Naiyer completed B.Sc. Degree in Computer Science and Engineering (CSE) from Brac University (BracU), Bangladesh in 2018. Currently, she is a Software Engineer, SQA at Therap Service LLC. Her research interests are Artificial Intelligence, Computer Vision, Program Ananlysis.



Ehsanul Khan completed his B.Sc. Degree in CSE from BracU, Bangladesh in 2019. His research interests include fire detection and computer vision.

Jia Uddin received Ph.D. in Computer Engineering from the University of Ulsan, Korea, in January 2015. He is an Assistant Professor in AI and Big Data Department, Endicott College, Woosong University, South Korea and an Associate Professor (On Leave), Computer Science and Engineering Department at BracU, Bangladesh. His research interests include fault diagnosis, computer vision, and multimedia signal processing. He is the corresponding author of this paper.