Facial image inpainting with implicit information of face landmarks

G048 (s1636732, s1655829)

Abstract

Image inpainting is a task to reconstruct an image by filling the missing region in that image. A state-of-art method, EdgeConnect (Nazeri et al., 2019), is able to reconstruct realistic images on masked images from image datasets CelebA (Liu et al., 2015), Places2 (Zhou et al., 2017), and Paris StreetView (Pathak et al., 2016). However, this method failed on reconstructing face features at correct locations when images are the side views of faces, covered in shadows or with large mask. Our method improves on models in EdgeConnect by introducing facial landmarks information implicitly. We evaluate our model on CelebA dataset and show that it outperforms the models in EdgeConnect.

1. Introduction

Image inpainting is a task of filling missing regions of an image through synthesizing alternative contents such that the modification is visually realistic and semantically correct. It has many applications, such as image restoration, and image editing to remove unwanted content.

method such as "EdgeConnect" (Nazeri et al., 2019), "Foreground-aware Image Inpainting" (Xiong et al., 2019), tackle the problem by separating the task in two stages. They firstly reconstruct the edge map of the corrupted images. Then they reconstruct the image using the produced edge map, which prevents the blurred images from being generated, enables the reconstruction of complex structure. Nevertheless, these methods would fail to produce complex facial structures on semantically correct locations when reconstructing side views of faces and faces covered in shadow. We believe it is because their model of reconstructing edge map will not generate a reasonable edge structure when the masked image contains rarely seen or complex structure (e.g. side view of faces). Our hypothesis is: adding additional information such as landmarks to model would help to generate a more reasonable edge map structure hence improve the performance for inpainting.

We limit our focus on face images and want to solve the problem of wrongly located facial structures in the reconstructed images. We want our model to be able to deduce the correct locations of different facial structures given the masked face images. Based on EdgeConnect, we add an intermediate branch in the original models to predict facial landmarks and introducing facial landmark information

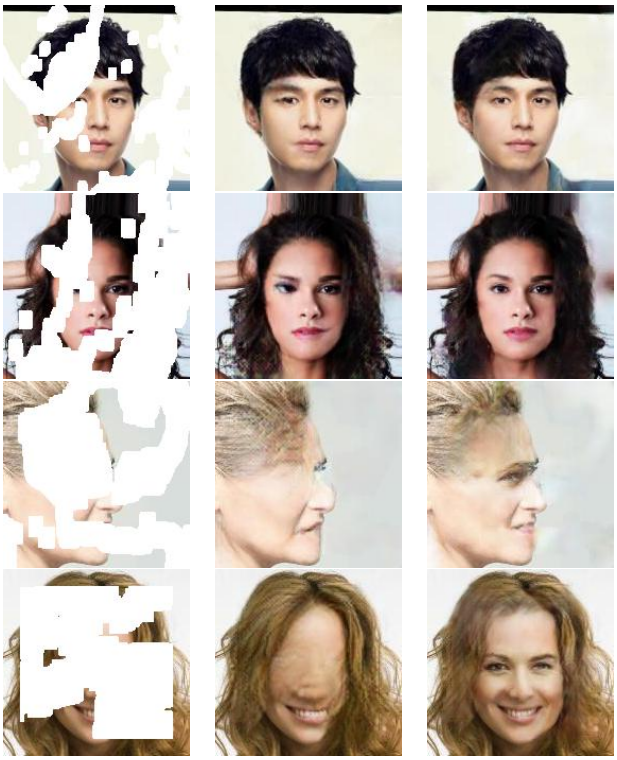

Figure 1: From the left to right are masks images, generated images by the model in EdgeConnect and generated images by our final model.

implicitly during the training stage of our model. Thus our model should have the ability to predict the correct location of different facial structures, and reconstruct facial structure correctly. Our model is trained and evaluated on CelebA dataset and compared with the EdgeConnect model. Our goal of this research is to find out whether adding additional landmarks information implicitly would help to improve the reconstruction of facial structure for image inpainting. Our code can be access on https://github.com/Tim-Tianyu/edge-connect.

2. Related Work

The traditional methods of image inpainting can be classified into two categories (Nazeri et al., 2019), patch-based and diffusion-based. Diffusion-based method (Wei & Liu, 2016; Xue et al., 2017)fills the missing region by propagating the information of the nearby area in a chosen direction. However, it only works well on a small scale (e.g. 64x64) of missing regions with a simple texture. Patch-based method (Muddala et al., 2016; Isogawa et al., 2018) fills the miss-

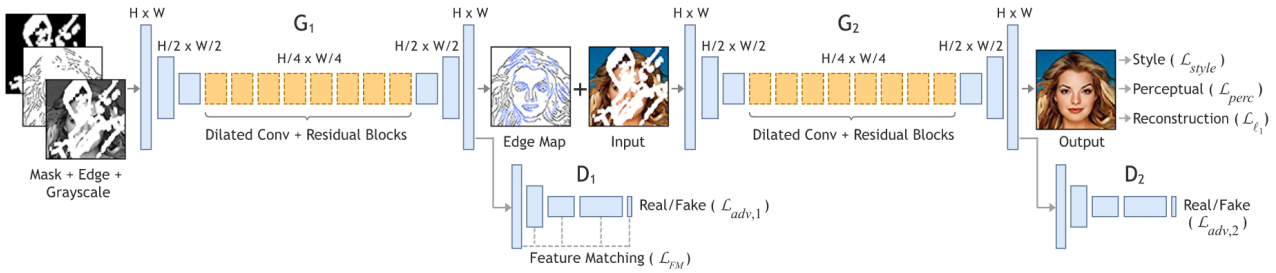

Figure 2: Architectural of model present in EdgeConnect, consisting two GANs, one for generating edge map in the first stage and inpainting in the second stage.

ing region by copying information from the best-matched regions from the same image or a collection of images. It works well for filling the background of images. However, the performance is poor for foreground filling with complex and unique textual. Both traditional approaches do not produce promising results on constructing intricate details like facial information.

Deep learning methods have achieved a much better result in recent years comparing to traditional methods. According to (Elharrouss et al., 2019), the deep learning methods could be classified into two main categories: CNN based and GAN based. Several CNN based methods (Weerasekera et al., 2018; Cai & Song, 2018) or encoder-decoder network based on CNN have been proposed for image inpainting. Those methods have shown excellent results of generating plausible new contents for highly correlated structured images such as faces. However, CNN based methods often create boundary artefacts, distorted structure and blurry texture inconsistent to the surroundings, which is likely due to the ineffectiveness of the CNN based networks modelling the long-term correlation between distance contextual information and the masked regions (Yu et al., 2018).

GAN-Based Methods are methods based on GAN framework. Generative adversarial network (GAN) introduced by (Goodfellow et al., 2014), is a framework which contains two neural networks contest with each other, a generator network G and a discriminator network D. The generator network is trained to capture the data distribution and generate images that are indistinguishable from real images. The discriminator network D is trained to differentiate between real and generated images. GAN based method tends to create realistic images with more exquisite details compare to CNN based method with only encoder-decoder structure, as the discriminator network will easily distinguish blurry generated images from real images. The recent state-of-art methods (Nazeri et al., 2019; Yu et al., 2019; Liu et al., 2018), are all based on GAN framework and have achieved a great result.

EdgeConnect(Nazeri et al., 2019) is a work we mainly focus on; our goal is to improve on their model. In EdgeConnect, the author proposes a two-stage image completion network, and each stage contains a generative adversarial network. In the first stage, the edge generation stage, the generator is trained to generate a complete edge map based on the incomplete grayscale image and the incomplete edge map. In the second stage, image completion stage, the generator is trained to generate a complete image based on the complete edge map and the incomplete image, the structure of their network is showed in figure 2, G1 and D1 are the generator and discriminator for the edge generation, and G2 and D2 are the generator and discriminator for the image completion. The structure of their generator follows from (Johnson et al., 2016), each generator first uses a convolution layer with a kernel size of 7, padding of 3 and out channels of 64, to keep the height and width of the input and enlarge the channel size to 64. It uses encoders to down-sample the feature map, reducing the height and width to one fourth and increase the size of channels from 64 to 256, it uses 8 residual block (He et al., 2016) with dilated convolution, where the first convolution layer in each residual block uses dilation of 2. At last, it uses decoders to up-sample images back to the original size, detail of their code can be seen on https://github.com/knazeri/edge-connect.

The models in edge connect can generate images with fine details, but the edge generator will fail to generate relevant edge map when the missing areas are highly textured and contain rarely seen structure. For example, it would be hard for edge generator to generate facial features like eyes, mouth or nose at correct positions when the face is side-viewed. We can see from figure 1 as an example of such failure.

3. Data set and task

The face dataset we used is "Large-scale CelebFaces Attributes (CelebA) Dataset" (Liu et al., 2015). This dataset is specialized in facial images, and contains about 200K (199,999) celebrity face images with size 218x178, each with annotations, the annotations include facial landmarks, facial attributes and identities. The facial landmarks are useful for our method for predicting facial landmarks based on the corrupted facial image, face landmarks in this dataset are positions of two eyes, position of nose and position of the left and right end of mouth, totally 5 landmarks. The diversities, quantities of this dataset will help us to train a network that generalizes well. The dataset comes with

a Train/Val/Test partition to follow with, and it will split the dataset into a training set with the size of 162770, a validation set of size 19867 and testing set of size 19962. We pre-process each face images by first centre crop each image into the size of 178x178 and then reshape each of them into the size of 176x176 as a factor of 4 to fit in the EdgeConnect network. We also pre-process the facial landmarks that indicate the correct locations of landmarks for processed training images.

The mask dataset we used is the NVIDIA Irregular Mask Dataset (NVIDIA, 2018). The dataset contains an unprocessed training, size of 55116, image shape of 512x512 and a processed testing set, size of 12000, image shape of 512x512 . We processed the training set in a way as same as the creator of this dataset. We use threshold 0.6 to binaries the masks first and then use from 9 to 49 pixels dilation to randomly dilate the holes and reshape each masks into 176x176. We also use "Random block" and "half" mask in training and evaluation. "Random block" mask is a square mask with height and width half of the size of the image and generates at random position on the image, "Half" mask is a mask that obstacle either all the left half or all the right half of the images. The three types of masks are randomly chosen with equal probability in training time and evaluation time. During the testing time, we only use the testing set from NVIDIA Irregular Mask Dataset.

Our task is to reconstruct complete face images from incomplete face images with missing regions. The incomplete face images are produced by combining mask images with face images, the masked areas in the face images are painted in white. We want to create a model such that by taking a incomplete face image and a mask image as input, the model can filling the masked region in the incomplete face image and generate a realistic face image that is close to the original complete face image. Following the method in EdgeConnect we break this task into two sub-tasks, we want to generate edge maps of the face images as a intermediate product. The ground true edge map of a images is generated using Canny edge detector (Canny, 1986) with sigma equals to 2, and the incomplete edge map is produced from the complete edge map by painting the masked area in black. In the first sub-task we want our model to generate an complete edge map of the faces, filling the masked region of the incomplete edge map, by taking the mask image, the gray-scale incomplete face images, and the incomplete edge map as input. In the second sub-task our model will take a complete edge map and an incomplete face image as input and generate a complete face image, filling the masked region in the incomplete face image. In each sub-task our model also predict face landmarks as side product, .

Finally, our evaluation metric for the generated images are PSNR, SSIM,and FID score.

Peak signal-to-noise ratio (PSNR) is an expression for the ratio that measures the quality between the original and reconstructed image(Instruments, 2019). The higher the PSNR value, the better the quality of the reconstructed image. The mathematical expression for PSNR shown below :

$$PSNR = 20log_{10}(\frac{MAX_f}{\sqrt{MSE}}) \tag{1}$$

where the MSE (mean squared error) is:

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\left\|f(i,j) - g(i,j)\right\|^2 \tag{2}$$

Legend:
f represents the matrix data of our original image.
g represents the matrix data of our reconstructed image.
m represents the number of rows of pixels of the image and **i** represents the index of the row.
n represents the number of columns of pixels of the image and **j** represents the index of the column.
MAX_f represents the maximum possible pixel value of the original image, which is 255 in our case.

Structure similarity (SSIM) is a perceptual metric that measures the perceptual difference between two images (Imatest, 2020) – it cannot judge which of the two is better. The mathematical expression for SSIM for two images x, y shown below:

$$SSIM(x,y) = f(l(x,y), c(x,y), s(x,y)) \tag{3}$$

Legend:
f represents the combination function.
$l(x,y)$ represents the luminosity comparison between two images.
$c(x,y)$ represents the contrast comparison between two images.
$s(x,y)$ represents the structure comparison between two images.
For more details of SSIM equations, see (Wang et al., 2004).

The higher value of SSIM meaning more similar between two images. SSIM(x,y) = 1 if and only image x and image y are same images.

Fréchet Inception Distance (FID) score (Heusel et al., 2017) is also a measurement that measures the quality of generated image. It calculate the distance between two feature vectors of original and generated images. The lower FID score indicates that two images are more similar. The score is calculated firstly by encoding two collections of real and generated images as 2,048 feature vectors using Inception V3 Model (Brownlee, 2019), then use the feature vectors to compute the FID score using the formula:

$$FID = \|mu_1 - mu_2\|^2 + Tr(\Sigma_1 + \Sigma_2 - 2 * \sqrt{\Sigma_1 * \Sigma_2}) \tag{4}$$

Legend
mu_1, mu_2 represents the feature-wise mean of the original and generated images.
Σ_1, Σ_2 represents the covariance matrix of the original and generated images

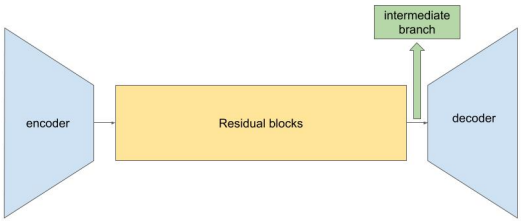

Figure 3: Position of the intermediate branch at the original model.

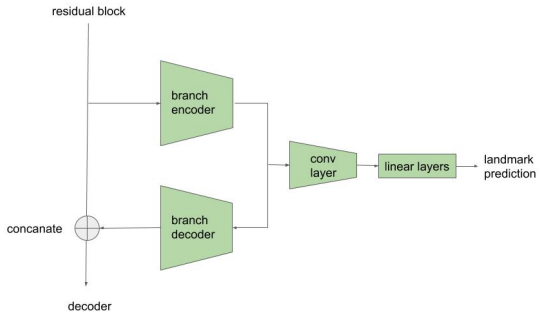

Figure 4: Architectural of intermediate branch, able to predict landmark position.

We do understand that PSNR and FID score has been shown in many papers (Chong & Forsyth, 2019; Shmelkov et al., 2018) that are no best metrics for measuring the generated images. However, we still decide to use it as it is a good comparison to the original model.

4. Methodology

Our model is largely based on the model in EdgeConnect (Nazeri et al., 2019), which we have already described details in section 2. In general, they proposed a "line first, colour next" approach, they use two GAN-based generator networks, one for each stage of the task. The first stage of the network will generate a complete edge map based on an incomplete edge map of the image. The second stage network will generate a complete image based on the incomplete image as input and a complete edge map. Let us name the first stage generator network G1, the second stage generator network G2 for convenience. We modify the structure of both G1 and G2 by adding an intermediate branch after the sequences of residual blocks, as shown in figure 3, to produce network G1' and G2', the modified network can only process image with height and width equals to 176, as the intermediate branch only takes fix-sized input. The basic structure of the intermediate branch is shown in figure 4 is same for both G1' and G2'. It takes an input of shape 44*44*256 from the output of the last residual block and produces two outputs. One of the outputs is the landmark prediction which contains five pairs of normalised cartesian coordinates that ranged in (0,1) indicating five landmarks positions. Another output is produced by the branch decoder, which has a shape of 44*44*256 and it will be concatenated back with the output of residual block as the concatenated block of size 44*44*516. It will be the input of the decoder.

This paragraph described the detail implementation of the intermediate branch showed in figure 3. The "branch encoder" consists of one convolution layer with 256 input channels, 256 output channels, kernel size 4, stride 2 and padding 1, reducing the shape of input from 44*44*256 to 22*22*256. The "branch decoder" take input from "branch encoder", consist of one Transposed convolution layer with 256 input channels, 256 output channels, kernel size 4,

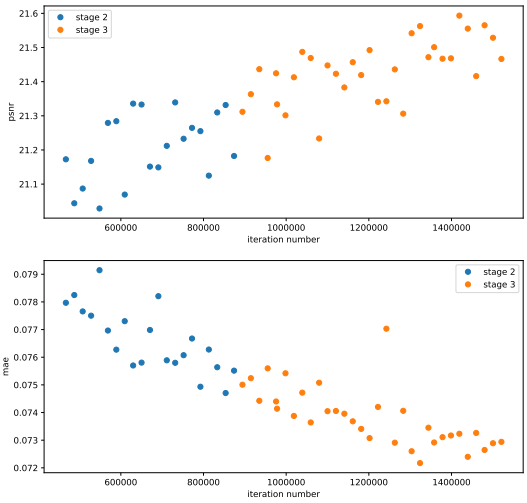

Figure 5: Training curve of our model at stage 2 and stage 3, evaluation metrics are PSNR and MAE.

stride 2 and padding 1, increasing the shape of input from 22*22*256 to 44*44*256. The "conv layer" showed in figure 3 is consist of a max-pooling layer with kernel size 2, reducing the shape from 22*22*256 to 11*11*256, and a convolution layer with 256 input channels, 256 output channels, kernel size 3, stride 3 and padding 1, further reducing the shape from 11*11*256 to 4*4*256. The "linear layer" showed in figure 3 consist of two successive linear layers, reducing the size of input from 4096 to 256 to 10. Finally, the data pass through a Sigmoid layer which constrained the output, the landmark prediction, in the range of (0,1).

Our network is trained similarly as the original network trained in EdgeConnect, but with fewer epochs due to the time and budget limitation. The training process involves three stages. In the first stage, G1' and G2' are trained separately. The edge model G1' is trained on the ground truth edge maps. The inpainting model G2' takes the ground truth edge maps as part of the inputs, so there is no connection between G1' and G2'. In the second stage, we

Original	Masked	(G1,G2)	(G1',G2)	(G1,G2')	(G1',G2')
					
Original -	Masked -	PSNR=22.92 SSIM=0.7517	PSNR=23.52 SSIM=0.7618	PSNR=21.88 SSIM=0.7371	PSNR=22.66 SSIM=0.7611
					
Original -	Masked -	PSNR=21.49 SSIM=0.7501	PSNR=23.07 SSIM=0.7625	PSNR=21.55 SSIM=0.6920	PSNR=22.10 SSIM=0.7632
					
Original -	Masked -	PSNR=25.02 SSIM=0.8615	PSNR=26.10 SSIM=0.8676	PSNR=24.16 SSIM=0.8430	PSNR=25.33 SSIM=0.8585
					
Original -	Masked -	PSNR=21.98 SSIM=0.7274	PSNR=22.18 SSIM=0.7389	PSNR=20.69 SSIM=0.6989	PSNR=20.94 SSIM=0.7179
					
Original -	Masked -	PSNR=22.30 SSIM=0.7320	PSNR=22.67 SSIM=0.7476	PSNR=20.21 SSIM=0.6814	PSNR=22.05 SSIM=0.7592

Table 1: Generated images of different models

only train G2', but this time, the edge map in the inputs is generated by G1'. In the third stage, two models are trained as a whole system. The errors in G2' is backpropagated all the way to G1', aggregating with the error of between the ground truth edge map and predicted edge map produced by G1'. We are using the same of hyper-parameters as EdgeConnect. Under batch size of 8, we trained our model for about 447612 iterations (21 epochs) in stage one, 447612 iterations (21 epochs) in stage two, 610380 iterations (30 epochs) in stage three. We can see in the result training curve 5 in stage 2 and stage 3, our network is still not converged, due to the time and budget limitation, we can not train our model for any more iterations, so we chose the latest model as our final model.

5. Experiments

In the experiment section, we showed the testing result of different combinations of 4 trained models, G1, G2, G1', G2' on the same mask and face image test set, which enable us to see the effect of adding intermediate branch at a different stage of the task. There are four possible combinations of these models: (G1, G2), which is the original pre-trained model from EdgeConnect, (G1, G2'), (G1', G2) and (G1', G2'). In table 2, we can see that (G1', G2) has the best performance, and (G1', G2') has better performance than (G1, G2') does, both indicating that G1' is having better performance than G1 as an edge model. We can also see that (G1, G2') has worse performance than (G1, G2) and (G1', G2') has worse performance than (G1', G2), both indicating G2' is having worse performance than G2 as an inpatient model.

However, when we go through the actual generated images, we found that (G1', G2') are continually producing the most visually plausible results, especially when a large portion of faces have been masked. Table 1 has shown some of these examples, we can see that (G1', G2') is

Model	PSNR	SSIM	FID
(G1,G2)	27.3914 ± 6.1734	0.9165 ± 0.0922	2.4333
(G1',G2)	**27.6684 ± 6.1429**	0.9240 ± 0.0849	**1.9151**
(G1,G2')	26.6643 ± 5.9379	0.9062 ± 0.1039	4.1512
(G1',G2')	27.2469 ± 5.8904	**0.9256 ± 0.0806**	2.1373

Table 2: different models' performance measured by PSNR,SSIM and FID using original and generated images (4 d.p.). PSNR, SSIM: higer is better; FID: lower is better

Figure 6: From the left to right are masks images, generated images by the model in EdgeConnect and generated images by our final model, showing our model failed to generate plausible faces.

producing most visually plausible faces, but the PSNR and SSIM of these images are lower than images produced by (G1, G2) or (G1', G2). The reason is that when the masked region is large and covers most of the face, model G2 tends to produce blurred images. Whereas (G1', G2') tend to produce images with finer details and face features. The difference resulting in (G1', G2') producing a face that is visually plausible but largely different from the original face, and a largely different face compared with the original face will yield a lower PSNR/SSIM score than a blurred face compared with the original face.

6. Discussion

Both (G1', G2) and (G1', G2') outperformed the original model in EdgeConnect, which indicates that adding implicit information of face landmarks do helps the original model generate facial structures at the correct location. It is especially helpful to edge model. We deduce that, as the edge model is the most import part for generating complex structure, the edge model would generate much better edges for facial structures based on the implicit information of face landmarks.

We have also seen that (G1', G2') is worse than (G1', G2) in PSNR and FID score, but the actual images produced by (G1', G2') are more visually plausible, we have explained the reason in 5. In short, the statistical difference between two different faces is more significant than the statistical difference between a face and a blurred face. As $PSNR \propto \log(\frac{1}{\sqrt{MSE}})$, a lower Mean Square Error (MSE) yields higher PSNR. The ordinal relationship between generated images would be the same in both MSE and PSNR, so we think PSNR is not a suitable metric in terms of image inpainting, especially for face images. Similarly, FID score is also not suitable as it calculates the distance between two feature vectors. The new image generated would have more different features, although it is plausible. We would still conclude that (G1', G2') is the actual best model in support of SSIM and visual evidence.

One strange phenomenon in our experiment is, (G1, G2') almost always had the worst performance among four models, both statically and visually. We infer the reason for it is such that the model G2' is trained to co-operate with model G1', a better edge model compare to G1, and a generator model would perform much worse when co-operating with a worse edge model, as this generator model does not have the robustness to deal with worse edges. Additionally, as G2 is having a more complex task than G1, it requires much more epochs to train until convergence, we know our model is not converged, so it might be that G1' is nearly converged, but G2' is still far from convergence.

Our final model (G1', G2') is still not perfect, we can find many examples where our model is having poor performance, we can see in figure 6. This usually happens when the most of face is masked. However, as we can see from figure 6, our model is still producing a more plausible image than (G1,G2). This is strong evidence supporting that our model is strictly better than the original one, in the term of generating more plausible images. We could conclude with confidence that adding implicit information of face landmarks improves the performance of inpainting.

7. Conclusions & Further work

The most important thing we have learned in this work is that we have proved our hypothesis is correct – adding additional implicit information of face landmarks indeed improves the performance of inpainting task, in the term of generating more plausible images. It is important because this methodology could apply to similar tasks to improve the model's performance. Also, we have learned that metrics such as PSNR and FID may not be representative enough for the quality of generated images in tasks such as facial inpainting. Better quality metrics comply with human perception should be found for facial inpainting tasks.

For further work, we would like to investigate adding new face attributes information provided in Celeba Dataset, together with facial landmarks, into the model. The attributes including information such as whether the person is attractive, wearing glasses and the gender of the person in the image. We think those additional attributes would improve the performance of the inpainting as landmarks did. We also would like to develop or adapting an existed metric to a metric more comply with human perception for inpainting.

References

Brownlee, Jason. How to implement the frechet inception distance (fid) for evaluating gans, 08 2019. URL https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/.

Cai, Xiuxia and Song, Bin. Semantic object removal with convolutional neural network feature-based inpainting approach. *Multimedia Systems*, 24(5):597–609, 2018.

Canny, John. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

Chong, Min Jin and Forsyth, David. Effectively unbiased fid and inception score and where to find them. *arXiv preprint arXiv:1911.07023*, 2019.

Elharrouss, Omar, Almaadeed, Noor, Al-Maadeed, Somaya, and Akbari, Younes. Image inpainting: A review. *Neural Processing Letters*, pp. 1–22, 2019.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Imatest. Ssim: Structural similarity index | imatest, 01 2020. URL https://www.imatest.com/docs/ssim/.

Instruments, National. Peak signal-to-noise ratio as an image quality metric - national instruments, 03 2019. URL https://www.ni.com/en-gb/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html.

Isogawa, Mariko, Mikami, Dan, Iwai, Daisuke, Kimata, Hideaki, and Sato, Kosuke. Mask optimization for image inpainting. *IEEE Access*, 6:69728–69741, 2018.

Johnson, Justin, Alahi, Alexandre, and Fei-Fei, Li. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Liu, Guilin, Reda, Fitsum A, Shih, Kevin J, Wang, Ting-Chun, Tao, Andrew, and Catanzaro, Bryan. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 85–100, 2018.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Muddala, Suryanarayana M, Olsson, Roger, and Sjöström, Mårten. Spatio-temporal consistent depth-image-based rendering using layered depth image and inpainting. *EURASIP Journal on Image and Video Processing*, 2016 (1):9, 2016.

Nazeri, Kamyar, Ng, Eric, Joseph, Tony, Qureshi, Faisal Z, and Ebrahimi, Mehran. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

NVIDIA. Nvidia irregular mask dataset. In *https://nv-adlr.github.io/publication/partialconv-inpainting*, 2018.

Pathak, Deepak, Krähenbühl, Philipp, Donahue, Jeff, Darrell, Trevor, and Efros, Alexei. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Shmelkov, Konstantin, Schmid, Cordelia, and Alahari, Karteek. How good is my gan? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229, 2018.

Wang, Zhou, Bovik, Alan C, Sheikh, Hamid R, and Simoncelli, Eero P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Weerasekera, Chamara Saroj, Dharmasiri, Thanuja, Garg, Ravi, Drummond, Tom, and Reid, Ian. Just-in-time reconstruction: Inpainting sparse maps using single view depth predictors as priors. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9. IEEE, 2018.

Wei, Yinwei and Liu, Shiguang. Domain-based structure-aware image inpainting. *Signal, Image and Video Processing*, 10(5):911–919, 2016.

Xiong, Wei, Yu, Jiahui, Lin, Zhe, Yang, Jimei, Lu, Xin, Barnes, Connelly, and Luo, Jiebo. Foreground-aware image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5840–5848, 2019.

Xue, Hongyang, Zhang, Shengming, and Cai, Deng. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *IEEE Transactions on Image Processing*, 26(9):4311–4320, 2017.

Yu, Jiahui, Lin, Zhe, Yang, Jimei, Shen, Xiaohui, Lu, Xin, and Huang, Thomas S. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Yu, Jiahui, Lin, Zhe, Yang, Jimei, Shen, Xiaohui, Lu, Xin, and Huang, Thomas S. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4471–4480, 2019.

Zhou, Bolei, Lapedriza, Agata, Khosla, Aditya, Oliva, Aude, and Torralba, Antonio. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.