

**Authors:** Rex Tse, Steven Luo, Peter Ng, Ronnie Jok

**Group:** St. Paul's Co-educational College Artificial Intelligence Team

**Contact:** [sp20166921@spcc.edu.hk](mailto:sp20166921@spcc.edu.hk) (Rex Tse), [sp20156401@spcc.edu.hk](mailto:sp20156401@spcc.edu.hk) (Steven Luo), [sp20166601@spcc.edu.hk](mailto:sp20166601@spcc.edu.hk) (Peter Ng), [sp20165841@spcc.edu.hk](mailto:sp20165841@spcc.edu.hk) (Ronnie Jok)

**Final Edit:** 2020/08/31

# A New Approach to Eye-to-Face Synthesis

## Abstract

Smart surveillance technology is becoming increasingly prevalent nowadays with the rise of artificial intelligence aiming for more precise and effective security systems, where face recognition is an important part of it. While the technology of identifying faces has been rapidly growing, in some cases, surveillance cameras can only catch some parts of a face, where computer generation of the whole face comes handy. To address this issue, this paper proposes a new Standard Deviation Loss to increase variety of output images, eliminating mode collapse, along with an approach for eye-to-face synthesis by using a generative autoencoder model with feature loss (using the VGG19 model).

## I. Introduction

Face generation has been a popular topic in machine learning for its easy data collection and is very suitable for demonstration of new generative model architectures. Developments such as the CNN [1] architecture, the VAE architecture, the GAN [2] architecture have a profound impact on not only face generation, but on all real-life generation tasks, such as sound generation, text generation, etc.. They show that deep neural networks are capable of understanding complicated real-life patterns which only humans are capable in the past.

Researches on face generation can be generally divided into two categories, unconditional face generation and conditional face generation. Unconditional face generation is comparatively a easier task, in which the model is only required to synthesize a photorealistic face with any prerequisites, such as requirements on skin color, race, etc.. Usually, the model is inputted with a random gaussian noise to ensure the variety of generated faces. The development of unconditional face generation has been rapid in recent years, with projects such as "This Face does not Exist" [3] using a StyleGAN architecture proving the possibility of the generation of detailed super high-resolution faces. In light of maturing technologies in unconditional face generation, projects in unconditional face generation start to appear. They are more useful in real life and can offer better computer-automated solutions to real-life problems such as smart surveillance. However, conditional face generation is much more challenging as it adds complication to the task of face generation, forcing the model to identify and utilize the correlation between faces and inputted information. One major example of conditional face generation is the recovery of faces from partial facial features, such as the generation of a full face from a side-angle face and eye-to-synthesis, in which our project is inspired by.

In the process of researching the topic, we have noticed that GAN seems to be a very popular if not the only solution in face generation. This can be easily understood as adversarial models are more capable of capturing small details which add to a realistic output and can constantly improve itself by the competition of a generator and discriminator model. Rich developments in GAN and the vast variety of GAN models also allow it to be applicable to many generative tasks, not limited to face generation. In our case, conditional GAN models are proven effective to force the model to find the relationships between inputs and outputs, i.e. the inputted eye and output face. Although GAN could be a possible solution to our task, we are dissuaded by its disadvantages. To name one, GANs are unstable as improvements in the generator model comes from the competition between the generator and discriminator, not a direct fit of input and output data. The generator or the discriminator can easily become too powerful and would cause failure as one of the model cannot learn from the other. This means that for the GAN architecture to work, constant optimization of the learning rates and ratio of learning of the generator and discriminator is required.

However, alternatives to face generation are few and are usually inferior to the GAN architecture. Solutions such as L1 loss always result in problems such as blurry images and mode collapse. In replacing GAN, we must be able to solve the two above problems. To increase the precedence of details in the generated face, we are inspired by the use of perceptual loss (i.e. indirect transfer learning of the VGG19 network) which focuses on textures and features but not individual pixels. To tackle the lack of variety in output images, we propose a new loss function, standard

deviation loss, in conjunction with L1 loss when training by each layers of the VGG19 network. Compared to GAN, this methodology is much more stable as it does not has adversarial elements in training, meanwhile preserving the main advantages of GAN, i.e. details in the generated face.

In the following part of the paper, Section II would introduce some related works to our project and explaining some inspirations and references we have taken when doing this project. Section III would mainly focus our data collection and preparation process. Section IV would be a in-depth explanation and analysis of the methodologies we have incorporated in our project. Finally, Section V would be a demonstration of our results and some evaluation of it.

## II. Related Works

Eyes to face synthesis can be considered as a generalized problem of face generation with partial facial features, for which images are reconstructed from partially available information. One of the popular solutions to this is by using Generative adversarial networks.

### Development of generative networks

When face generation is first introduced, simple neural networks were being used. In [1], dense 3D face and single face image are reconstructed from monocular video, especially on the usage of image-based 3D face reconstruction in face recognition [4], [5] and face animation [6] [7] In early networks of such, L1 loss (Manhattan distance loss) and L2 loss (Euclidian distance loss) were being implied to minimise the error. However, the results of CNN with L1 and L2 losses are still not satisfactory because the images tend to get blurry and fail to grasp details, as well as having artifacts [8]

[2] processed improved networks, by proposing a new network called the Generative Adversarial Nets. Based on a double model involving a generator and a discriminator, as the two trains each other to solve more complex problems, including style transfer [9], super-resolution, [9] and image inpainting [10] However, GANs are unstable in training, leading to poor results and problematic tuning. [11] used the GEOMETRIC METRICS REGULARIZER and the MODE REGULARIZER to solve the problem based on the original GAN model to create MRGAN, and [12] created a new algorithm from the traditional GAN model, by replacing the JS-Divergence with 1-Wasserstein distance to measure the difference between the model and target distributions, leading to better results and more efficient training [12]. [13] used Conditional Generative Adversarial Nets (CGAN), which fed the original input to both the generator and critic, for the discriminator to recognize the relevance between the input and output, avoiding mode collapse and increase correlation between the result and the original data.

[14] designed an end-to-end network based on conditional generative adversarial networks (GAN) to generate the face information based only on the available data of eyes region. Despite having significant improvements, the GANs still have the issue of unstable training process which is rooted in their architecture. To solve the problem [15] used a thorough evaluation of networks of increasing depth using an architecture with very small ( $3 \times 3$ ) convolution filters. Being a totally different architecture, there is no adversarial within the network, show significant impact on eliminating early problems.

### Face Generation

Face generation without condition can be treated as face inpainting with random noises by using training for GANs. [16] produces high resolution (1024x1024) face images with a relatively stable training instance by taking in randomly initialized latent vectors as inputs and constructs a never-before-seen face image. [17] produces state-of-the-art images with great clarity detail and is further implemented in the well-known “this person does not exist” project [18].

Face generation with conditions requires more manual control to integrate computational face generation into real-life applications. [19][9][20] add an extra parameter “condition” to the formula of generation. Conditional GAN, or CGAN, originated from [21]. In [22], faces are being reconstruct from incomplete face images. DeMeshNet reconstructs detailed ID photos that are corrupted by mesh-like lines and watermarks through a 3-step feature extraction process. With breakthroughs in GANs and its variants, [14] reconstructed face from corresponding eyes as an attempt to improve surveillance abilities. [9] proposed a Unet-structured GAN that maps an image with only a pair of eyes to an image of the corresponding face.

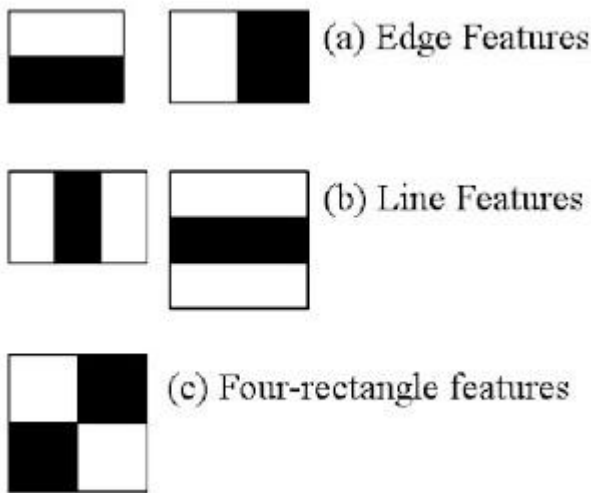
With the rapid development of face recognition and reprinting technology, GANs and its variants are being regarded as typical models for image inpainting and models of such. After our attempt to recreate the architecture in (Chen, X., Qing, L., He, X., Su, J., & Peng, Y. (2018). From eyes to face synthesis: a new approach for human-centered smart surveillance. IEEE access, 6, 14567-14575.), we found GANs are unstable in training [23], [12], [21]. Referring to

[15], its pretrained, deep convolution layers of the VGG model does a good job on grasping the details of the image, hence feature loss is considered our alternative for image processing. Feature loss functions are used when comparing two different images that look similar, and was being used in [24] for Real-Time Style Transfer and Super-Resolution by a per-pixel loss between the output and ground-truth images. This gives similar qualitative results comparing to the optimization-based method but is three orders of magnitude faster, as well as giving visually pleasing results while a per-pixel loss is being replaced by a perceptual loss. This inspired us to come up with our own Standard Deviation Loss to complete our model.

### III. Data

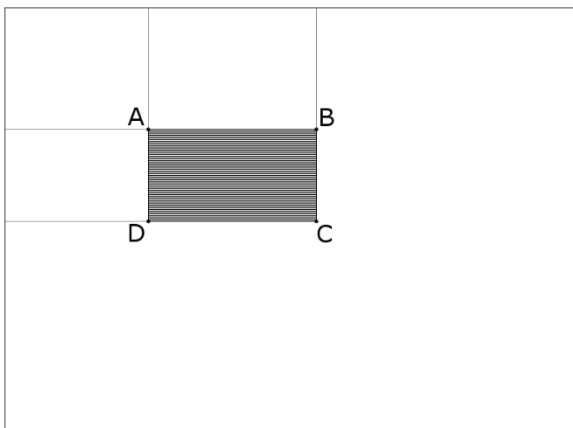
In our project, the first 30000 faces from the CelebA dataset (Liu, Z., Luo, P., Wang, X., & Tang, X. (2018). Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15, 2018.) (in a total of 202599) were used for training. Meanwhile the remaining faces were used to evaluate the model. Faces were cropped from the dataset, and eyes were also being cropped using the cv2 module (Open Source Computer Vision Library), using haar-feature based cascade classifiers [25]. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). IEEE.)

A Haar feature refers to the value representing a section of an image, which indicates the existence (or absence) of certain characteristics (such as edges or changes in texture) by calculating the differences between sections. (Haar-like feature. (2020, May 6). Retrieved from [https://en.wikipedia.org/wiki/Haar-like\\_feature](https://en.wikipedia.org/wiki/Haar-like_feature))

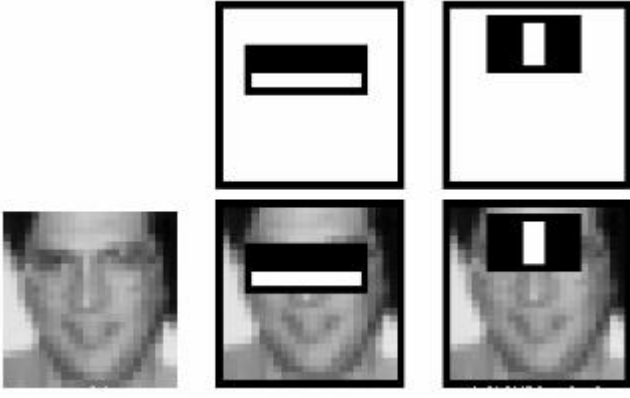


A rectangular Haar feature can be defined as filters that are calculated by taking weighted sums of integrated areas of the images. [25] The equation is as follows:

$$\text{Sum} = I(C) + I(A) - I(B) - I(D)$$



Finding the sum of the shaded rectangular area and to figure out the best features out of all thousands of them, Adaboost was being used, as we applied each and every feature on all the training images. With features of minimum error rate selected, they are sorted to most accurately classify the face and non-face images.



Note that classifiers are also categorized into strong and weak classifiers, being reflected on its error rate. Weak classifiers can be combined into a strong classifier by taking the weighted sum of their results over an identical feature. After the cropping process, faces are resized to 128x128 and eyes are resized to 32x32 by a bilinear interpolation. It works by taking four known neighbouring pixels as reference, then normalize the data from the range of (0,255) to (0,1).

The normalization process can be summarized by the following:

$$N(r, g, b) = (\frac{r}{255}, \frac{g}{255}, \frac{b}{255})$$

where r,g,b represent the three color channels of a pixel.

## IV. Methodology

The aim of face synthesis from a single eye is to supplement the missing information of a face in the image of an eye. The problem of eye-to-face synthesis can be summarized in a function:

$M(E) = F$ , where M represents the model, E is an eye (the input) and F an image of a face (the output).

Specifically, we used an autoencoder network in our model, combined with feature loss and standard deviation loss during training.

### Autoencoder

Our autoencoder can be divided into two parts, the encoder and the decoder. The aim of the encoder is to convert the image of an eye to a single vector that contains the important information of the eye, while the decoder turns the vector into an image of a face. It can be summarized by the following equations:

$$\begin{aligned} f_e(E) &= \vec{e} \\ f_d(\vec{e}) &= F \end{aligned}$$

An image of an eye of dimensions 32\*32\*3 is inputted, then this image passes through multiple convolutional layers and maxpooling layers until the eye image is reduced to 4\*4. It is then flattened and followed by a fully-connected layer which reduces the vector to a vector of 512 dimension which represents information encoded from the input eye.

In the decoder, the vector, which is the output of the encoder, is first expanded to a vector of dimension 4096 by adding a fully connected layer. The vector represents information of the output eye which the decoder would turned into a face image. Then, the expanded vector passes through consecutive layers of convolutional layer and upsampling repeatedly until the image size reaches 128\*128.

### Standard Deviation Loss

To increase the variety of generated images, we introduced a new loss called standard deviation loss. The objective of the minimization of standard deviation loss is to keep the diversity of the generated batch and the standard batch at a similar level, so as to prevent the generated images being too diverse or too similar. The standard deviation of the output batch and the target batch is identical, standard deviation loss is 0. If the standard deviation of the output batch is too big or too small, the standard deviation loss would be large. Standard deviation loss takes the absolute value of

the difference between the standard deviation of the generated batch and the target batch. The loss function can be summarized by the following equation:

$$\sigma\_loss = |\sigma(y\_true) - \sigma(y\_pred)|, \text{ where:}$$

$$\sigma(B) = \frac{\sum_{k=0}^n \sigma(k)}{n},$$

where  $B$  represents a batch of images,  $k$  represents a pixel,  $n$  represents the total number of pixels in one image

Our current result proves that standard deviation loss is an effective way to prevent the model collapse L1 loss tends to bring, instead of focusing on the absolute error within each pixel, this brings emphasis on the variance faces also have, adding another perspective to the problem of eye-to-face synthesis.

However, there are some items that require special attention when using this loss. Firstly, it is most effective when applied to every layer in the VGG model. This is due to the lack of variance in pixel values in a batch of face images. Considering that hair color and skin color can be strikingly similar, most pixels of a face image can be mostly the same among a batch. However, if features grasped by the VGG model are also added with this loss, it would force variance in facial features. Secondly, a lack of divergence of faces in a training batch should also be avoided as it minimises the effects of the loss function.

## Mean Absolute Error Loss (L1 Loss)

L1 loss is commonly utilized in computer models as it is the simplest most intuitive method of calculating loss. Its objective is to minimize the differences in the pixels between the generated image and the target image. L1 loss takes the mean of the absolute error of the generated image when compared to the target image, without considering their directions. The range of L1 loss is from 0 to  $\infty$ .

The formula for L1 loss is:

$$\text{L1 loss} = \frac{\sum_{i=1}^n |y_i^{real} - y_i^{pred}|}{n}$$

Where  $n$  is the total number of pixels. [26]

However, L1 loss is not used in our model as it creates artifacts and causes the image to be blurry [8]. L1 loss only measures the euclidean distance between the pixels of the generated image and target image, thus it is hard for a model using L1 loss to grasp concepts such as edges and textures. It also causes the problem mode collapse as a model can achieve a low L1 loss even if it generates similar images for different inputs.

## Feature Loss

Inspired by the paper *Perceptual Loss for Real-Time Style Transfer and Super-Resolution* [24], we decide to use feature loss to replace GAN loss and L1 loss in order to compensate for the shortcomings of them. By using feature loss, our model is more able to generate an image that is more similar to the target image, instead of constructing a whole new face. When compared to other kinds of losses, feature loss is more able to identify the details of features in an image, thus creating a more realistic image that is similar to the target image.

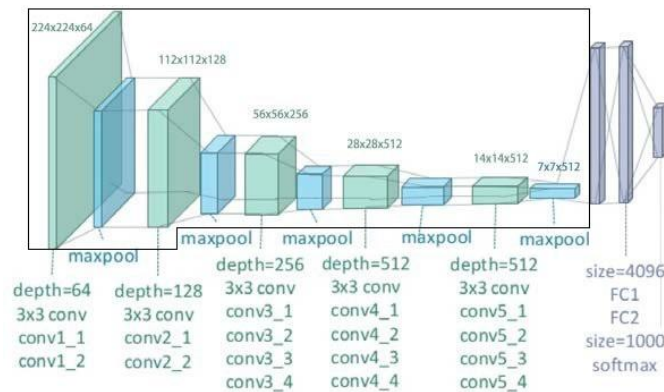


Image IV.1 The architecture of VGG neural network, the layers included in the box is the part we utilized in our model (Modified from: [https://www.researchgate.net/figure/illustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means\\_fig2\\_325137356](https://www.researchgate.net/figure/illustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means_fig2_325137356))



In our model, we construct a model combining our generator and the VGG model. In training, the output of each layers of the VGG model in the combined model would be compared with the actual output when the target image is feed to the VGG model by the L1 loss. The output of early layers of a VGG model are details concerning the local features, which are very useful for reconstructing the image of a human face. In contrast, the latter layers of a VGG model provides us with more global features, which are less useful for generating human faces, as the global features of all human faces are generally very similar, for example, the nose of all people are placed in the centre of our faces. Therefore, we would like to place more importance on local features, and the loss weights of the early layers of the VGG model is set to be larger. By doing so, our model can generate images with high similarity to the target images as it actually reconstructs a face from existing features instead of generating a new one. Overall speaking, our feature loss can be summarized into the following equation:

$$\text{Feature Loss} = \sum_{k=1}^l w_k \left[ 0.75 \times \frac{\sum_{i=1}^n |y_i^{real} - y_i^{pred}|}{n} + 0.25 \times |\sigma(y_{real}) - \sigma(y_{pred})| \right]$$









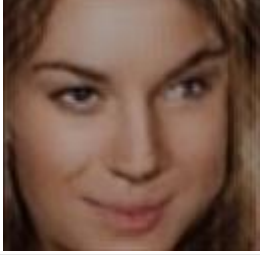



$$w_k = \left( \frac{ar - a}{r^l - 1} \right) r^k, \sum_{k=1}^l w_k = a$$

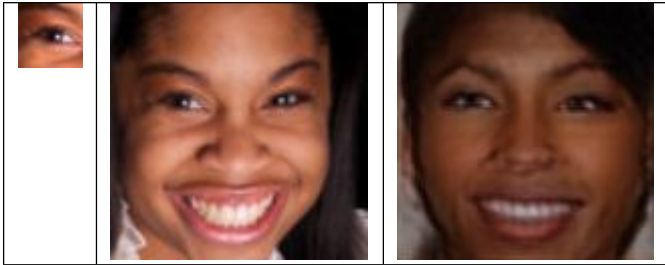
Where n is the total number of VGG layers used, k in the layer used, r is the rate of regression.

## V. Results

### Training and Results

Our model was trained using the Adam optimizer using the suggested learning rate ( $10^{-3}$ ) [27]. To prevent overfitting and increase the training efficiency, batch normalization is used [28]. Our training set consists of around 6000 pairs of eyes and faces, which are feeded into the model with a batch size of 64 during training. Some results are demonstrated as follows:

Eye	Target Face	Generated Face
		
		
		
		




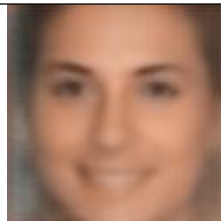







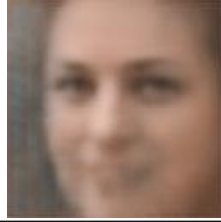


### Evaluation

The overall quality is satisfactory, the faces generated are very clear. the wrinkles are visible and obvious. Skin colours are very accurate, however the brightness of the generated portraits is relatively dark. Men generated usually have mustache. Age might not be always accurate, with young faces being generated as an older person due to the lack of young people’s face data in CelebA. Our L2 loss averages about 0.08.

On another note, we can notice that the training time is much shorter than GAN, with only about 1.5 day of training time and 150 epochs for the above resultchs. It’s because the GAN model requires separate trainings of both the discriminator and generator for each batch. This shows that within the same time period, the above model can be trained a lot more times and hence ensuring better result.

### Comparison with L1 loss

Eye	Target Face	Current	L1
			
			
			

Being reflected by our model with L1 loss, it is obvious to us that most of the faces are pixelized, with no clear face shape. The model also couldn’t accurately grasp the specific facial features, as the generated portraits are usually blurry. The faces are all similar in general, the global features of faces are very similar, for example the features of the eye, as well as the same eye colours, showing the huge problem of mode collapsing. The shadowing of the generated images is also very poor, with some of the dark areas covering up the face, alongside with unclear wrinkles and facial lines, as well as jawlines. The faces also look flat and foggy, without any 3-D features.

Comparing with our model, L1 loss is generally inferior. Benefiting from our outstanding Standard Deviation loss, our model shows much more variation in facial features, for instance, eyelashes, jawlines, wrinkles, as well as accurate skin colours and better shadowing. The over picture is smoother with more significant differences between colours, as well as a 3-D face structure. The above difference in performance is mainly due to the disadvantages in the L1 loss function, namely the loss function only cares about the absolute error in pixel values, not the relations between pixels or facial features.

## VI. Conclusion

Although the GAN network might be the most dominating methodology towards face generation, our model provides a possible alternative for replacing GAN networks in image generation. Furthermore, we may try to use adversarial loss, or other kinds of face recognition networks to replace the VGG network used in our model. In the future, we may create models that can generate a human face from other partial facial features, e.g. mouth and nose, and our ultimate goal is to create a model that can generate an image of a human face from any partial face features provided to it.



## References

1. Guo, Y., Cai, J., Jiang, B., & Zheng, J. (2018). Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE transactions on pattern analysis and machine intelligence*, 41(6), 1294-1307.
2. Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.
3. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8110-8119).
4. Tuan Tran, A., Hassner, T., Masi, I., & Medioni, G. (2017). Regressing robust and discriminative 3D morphable models with a very deep neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5163-5172).
5. Blanz, V., & Vetter, T. (2003). Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9), 1063-1074.
6. Ichim, A. E., Bouaziz, S., & Pauly, M. (2015). Dynamic 3D avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4), 1-14.
7. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2387-2395).
8. Zhao, H., Gallo, O., Frosio, I., & Kautz, J. (2016). Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1), 47-57.
9. Lu, Y., Tai, Y. W., & Tang, C. K. (2018). Attribute-guided face generation using conditional cyclegan. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 282-297).
10. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536-2544).
11. Che, T., Li, Y., Jacob, A. P., Bengio, Y., & Li, W. (2016). Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*.
12. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.
13. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
14. Chen, X., Qing, L., He, X., Su, J., & Peng, Y. (2018). From eyes to face synthesis: a new approach for human-centered smart surveillance. *IEEE access*, 6, 14567-14575.
15. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
16. Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
17. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
18. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8110-8119).
19. Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, 2014(5)*, 2.
20. Di, X., Sindagi, V. A., & Patel, V. M. (2018, August). Gp-gan: Gender preserving gan for synthesizing faces from landmarks. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 1079-1084). IEEE.
21. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

22. Zhang, S., He, R., Sun, Z., & Tan, T. (2017). Demeshnet: Blind face inpainting for deep meshface verification. *IEEE Transactions on Information Forensics and Security*, 13(3), 637-647.
23. Che, T., Li, Y., Jacob, A. P., Bengio, Y., & Li, W. (2016). Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*.
24. Johnson, J., Alahi, A., & Fei-Fei, L. (2016, October). Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision* (pp. 694-711). Springer, Cham.
25. Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.
26. Grover, P. (2020, May 27). 5 Regression Loss Functions All Machine Learners Should Know. Medium. <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>
27. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
28. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03*