~:Summary for StyleGAN (Karras et al 2019):~

For a quick recap of SG

The fundamental idea is to grab features from latent code and reduce the load on the generator, which now focuses on constant latent code (W) rather than input space of latent Z. The SG (Karras et al. 2019) paper targets the generator part specifically and takes the rest of the GAN architecture as the same. The author claims the following: 1. A kind of unsupervised learning develops inherently in the architecture such that it is able to separate the latent factors / features from the compact representations. 2. Better Interpolation and disentanglement between features. These both claims are able to prove better than previous SOTA Architecture. And gets SOTA performance. The principle idea is to reduce the working load of the generator G. Generator sees the input after it is completed with style transfer. The idea is to throw an input to the generator which is already processed to understand the latent features more precisely. This enables directly controlling the image features such as pose, identity in the humans and minute features like hair, neck, ear etc.This brings the learning of the specific to very minute features which looks realistic than previous architectures. The model assumes each image as a collection of features which it needs to separate. The Nvidia corp was able to come up with a new tool which is able to tune the strength of each style you apply to the image.

This new generator is able to understand how the latent features are represented in the network. Previous architectures used to follow a technique which tells that the new generated outputs should be as close to the original input images. The input latent space must follow the training data. However the author claims that this brings unavoidable constraints and entanglement. To overcome this they have allowed disentanglement. Basically in simpler words, the author found that in older techniques the features were entangled when represented in compact representation or the code (recall code from encoder decoder.) This entanglement does not allow to study or learn specific style related features presented in the face input image. This way the author is trying to break the correlation developed inherently between the GAN networks. To achieve this and solve the problem of entanglement, they propose 2 algorithms Perceptual Path length and Linear separability. They have also released a high quality image dataset as well named as FFHQ (Flickr faces HQ).

The fundamental Vanilla GAN architecture is to give density estimation such that probability(observation) == probability(synthetic observation) recall auto-encoder (where we tried to equate X and X^cap) We have also seen that we throw encoder part and give randomly generated vector to decoder. So we have seen that the Generator (Gr) model is able to improve by taking loss from Discriminator (Dr) so that the Discriminative model fails in the next iteration. And Dr will improve from Gr loss. There is a fight between generators and discriminators to classify fake and real images and to generate more realistic images. The basic idea is Gr tries to fool Dr and Dr tries not to be fooled. We try to achieve Nash equilibrium. **This is the problem where the entanglement gets started**. Again, the Nash equilibrium is the state where my P data or the P input real data is equal to P of the synthetic data which the decoder generates. When both the states are equal then we say that decoder is trained well or nash equilibrium. In

VAE also we tried to minimise the KL and reconstruction loss or class identity. Along with interpolation. Where we wanted to match two probability distributions. (Synthetic and Real images).The basic idea is to divide the features and collect them to a space and add those features to the way we like. So if you want to perform ageing to a face or add a beard to a face then you can.

Authors came up with a new style based generator architecture which removes the dependency of input and omits the input layer. Now the subsequent layers just learn from a constant W. A new mapping is then created at early layers along with AdaIN operations and found that the network no longer benefits from feeding the latent code to the first conv layer. By adding the mapping network and AdaIN operations, surprisingly observation showed that the network no longer benefits from feeding the latent code into the first conv layer. Thus mapping network is introduced in the Style Gan which extracts the features already for the generator. This way it tries to learn specific features from the image as well. The latent vector W which is also called an intermediate controls generator with Adaptive Instance Normalisation.The AdaIN brings the relative importance of features for the subsequent conv layer at the same time being independent of original images, since normalised. This way each style is taken into account for one convolutional block unit of the generator.

So collecting everything, in simpler terms, The architecture starts with z input, then it's passed to FC layers where specific features are extracted. Those features are passed to the generator. Remember we are not passing images directly, we are passing the latent code. The styles “Y” is a tuple of styles collected that controls adaptive instance normalisation. This is because each feature map Xi is normalised separately with corresponding Yi from the tuple Y (that is collection of styles.) Basically normalising the feature map wrt that particular style. Then secondly, a dedicated noise is then added to the generator phase where it is passed to each of the feature maps. This noise image is specific for the specific learned feature maps. (Please refer the architecture in the paper)

Style Mixing: On getting successful results from one latent code representation, the author further improved and augmented another latent code to display the mixing of styles. This created amazing results where a single image is realized from two types of latent codes w1 and w2 from z1 and z2. This helps to overcome the biggest challenge in what they are addressing is that it prevents the network to memorise the features that are correlated. That is for example, a child does not have a beard when regularised with such a AdaIN and Noise vector, but style gan is able to develop such images as well.

Stochastic Variation: It is simply a variation of minute features of the image. Earlier methods tried to target specific layers which were computing the respective features so that it can be normalised, however this is a very tedious task to pick a layer and then normalise it wrt that particular feature always. Since we want everything automatically. Author proposed that adding per pixel noise will help in better understanding of the minute textures in the images. For example hair, eye brows, etc. Different noises are realised to create different patterns for the

stochastic variations. This is a clever feature which most of the time limited human observers do not notice.

Disentanglement Studies: Perceptual Path Length and Linear Separability.
The techniques measure the disentanglement. Since the author uses baseline (ProGAN) which eliminates the auto-encoder or compact representational encoder which maps images to compact latent code part, the author proposes 2 new disentangling techniques, such that it is possible to remove the intra-features dependency of images. Authors claim that the generator architecture described here is that where intermediate latent space W is a collection of extracted features, does not support sampling on fixed distributions. Since mapping is learned in smaller steps as we have seen earlier, in progressive manner we can get the original features in W space rather than input space, However the main purpose is to extract the W space to different latent factors which contain learned attributes in the process before. As an agreement, it is true that if we had latent codes of each of the face features we would be able to control the features in the image and come up with a completely different representation.

Perceptual Path Length: This will find the distance between feature maps visually. If separating the images far off and we observe a drastic change, this means the feature maps are entangled. Suppose removing the hair feature removes ears also, then we need to take care of it as removing hair shall not remove ears. Or vice versa. It measures the difference between consecutive images (their VGG16 embeddings) when interpolating between two random inputs. Drastic changes mean that multiple features have changed together and that they might be entangled.

Linear Separability: This brings the binary classification for the inputs. It should be high as we want to separate features. The ability to classify inputs into binary classes, such as male and female. The better the classification the more separable the features. (SVM was used for this with prediction probabilities)