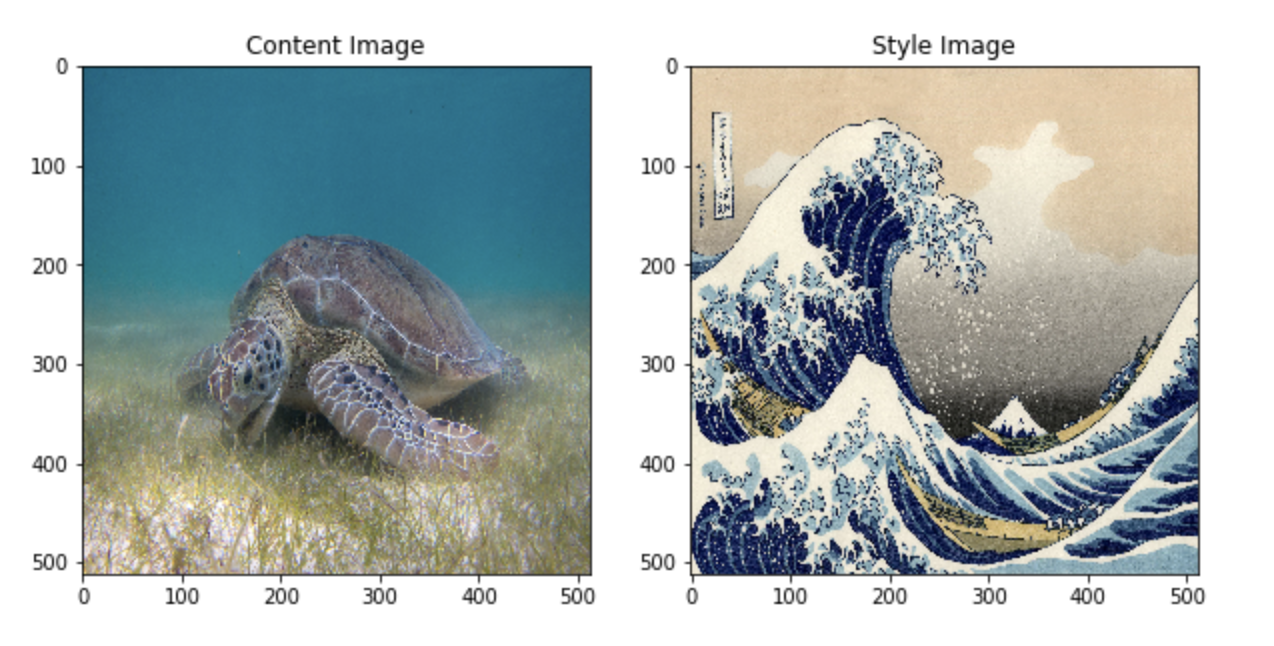
***Content for the presentation:***

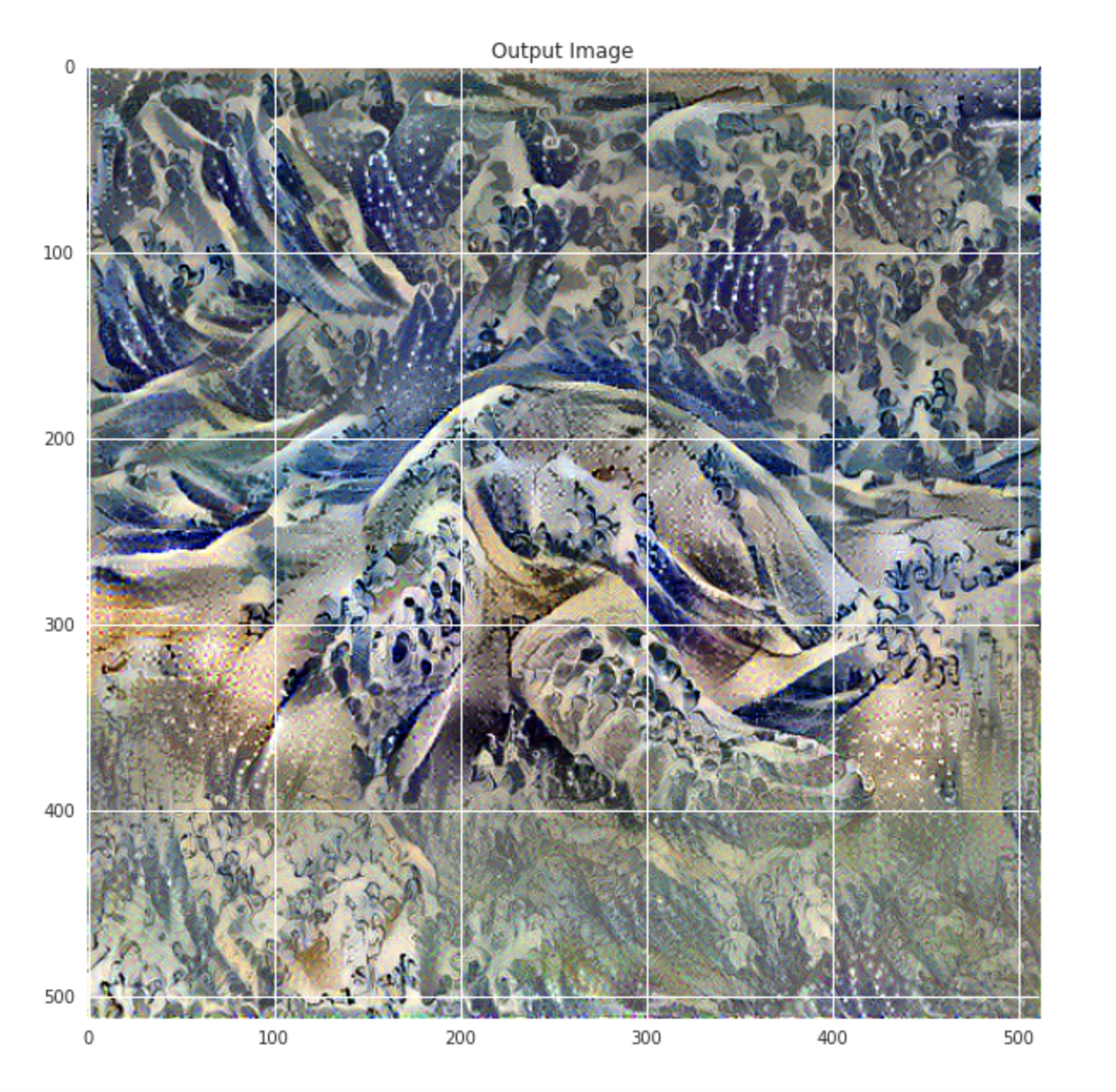
**Let me connect the dots from my freind’s presentations**

**ML2 Presentation - Style Transfer / Style GAN 2019**

There are some prerequisites for the paper to start before I would like to tell some of the terminologies which will help you to stay with the paper in the following slides.

**Style Transfer** - It is a ML technique that takes 3 images as input. a **content** image, a **style reference** image, and the image you want to style. It includes the process of blending content image and style image such that you transform your content image with textures of style image.





Disentanglement - It means separating the features at latent space. Remember the part of the encoder decoder where you get a compact representation of the input. At that level the author tried to separate the features.

Interpolation - Interpolation is a very important term wrt to ML. Because essentially you always envision maximum interpolation between classes. It says that you want to classify the input data into clusters such that you maximize the inter class distances and minimize the intra class distances. For example half space classification.

The paper targets the generator part specifically and takes the rest of the GAN architecture as same. The author claims following:

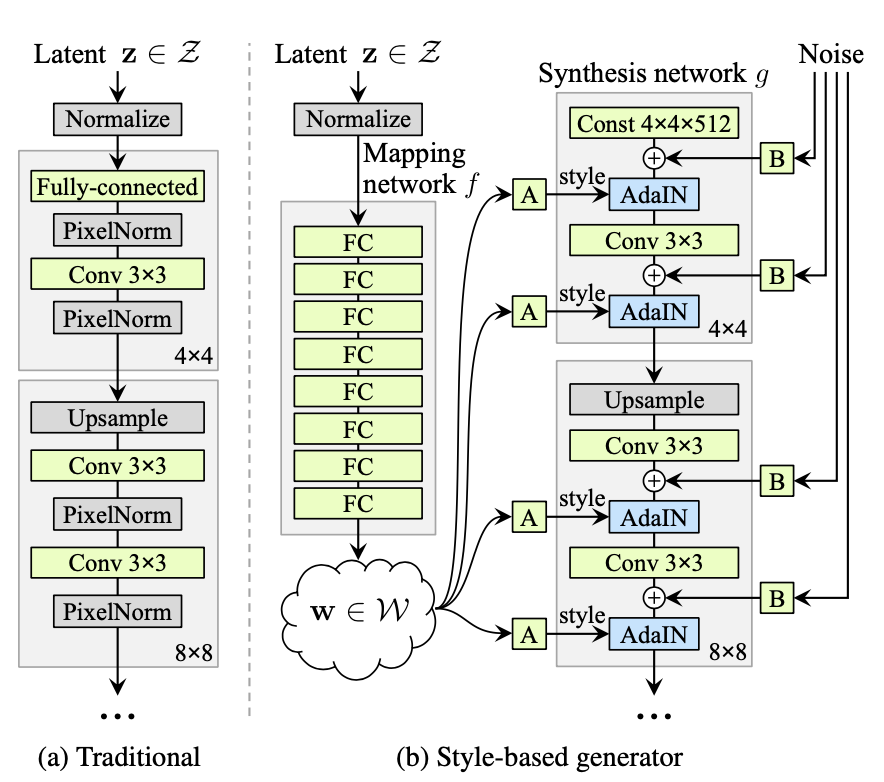
1. A kind of unsupervised learning develops inherently in the architecture such that it is able to separate the latent factors / features from the compact representations.
2. Better Interpolation and disentanglement between features.

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.

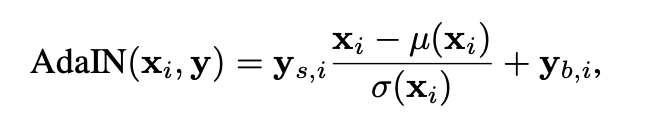
This new generator is able to understand how the latent features are represented in the network. Previous architectures used to follow a technique which tells that the new generated outputs should be as close to the original input images. The input latent space must follow the training data. However 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).

Now I will try to tell you how the basic GAN architecture we have studied in class have a problem, The fundamental target of that architecture is to give density estimation such that probability (observation) == probability(synthetic observation) recall autoencoder (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. And the author’s idea to overcome this entanglement is what I am going to explain. 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 minimize 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 aging to a face or add a beard to a face then you can.

**Style Based Generator Architecture and Properties:**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 con 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 as intermediate controls generator with Adaptive Instance Normalization. (represented as A in the figure).



The AdaIN brings the relative importance of features for the subsequent conv layer at the same time independent of original images, since normalized. 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 in an array that controls adaptive instance normalization. This is because each feature map Xi is normalized separately with corresponding Yi from the tuple Y (that is collection of styles.) Basically normalizing 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.

**Quality of generated Images:** redesigning does not compromise the quality. Authors used baseline as progressive GAN from which they imported all hyperparameters. The baseline was improved by adding AdaIN and Noise (Google adversaries which are actually the styles) They devised a novel mixing regularization which helps interpolation and generates new images in a controlled manner with high quality.

**Comparison with ProGAN:** Progressive Gan or progessive training of GAN improves the image quality. So they started with a small dim of images and improved it progressively. The demerit there in ProGAN was the features are entangled and there is lack of freedom to choose the features of an image. Say for face its nose, eyes, wrinkles etc, The attempt to tweak anything changes the whole distribution of the image quality. Here when comparing ProGAN and Style GAN the main difference is that in Style GAN we learn an intermediate latent representation of the input whose different outputs are different features (Using the idea that early layers of the Neural network learns very small features and bottom layers learns the class specific features) Thus different outputs from the intermediate layers we can control them.

**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 memorize the features that are correlated. That is for example, a child does not have a beard when regularized 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 normalized, however this is a very tedious task to pick a layer and then normalize 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 realized to create different patterns for the stochastic variations. This is a clever feature which most of the human observers do not notice. essentially Time limited humans do not notice.

Really recommend to watch this video:

<https://youtu.be/kSLJriaOumA>

**Disentanglement Studies: Perceptual Path Length and Linear Separability.**

The techniques measure the disentanglement. Since the author uses baseline (ProGAN) which eliminates the autoencoder 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-dependency between features 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 agreeing point, 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.

1. **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. Measure 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.
2. **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)

The study of these disentanglement makes a claim that features are more easily separable in W space than in Z space.

**Details:**

StyleGAN was trained on the CelebA-HQ and FFHQ datasets for one week using 8 Tesla V100 GPUs. It is implemented in TensorFlow and will be open-sourced.

**Conclusion:**

StyleGAN is an amazing paper that not only produces high-quality and realistic images but also allows for superior control and understanding of generated features in the images, making it even easier than before to generate believable fake images. The techniques presented in StyleGAN, especially the Mapping Network and the Adaptive Normalization (AdaIN), are the basis for many future innovations in GANs.

**Advantages:** This is an enhancement of ProGAN. Now you can easily manipulate entangled features. High quality and realistic images are generated.

**Limitations:**