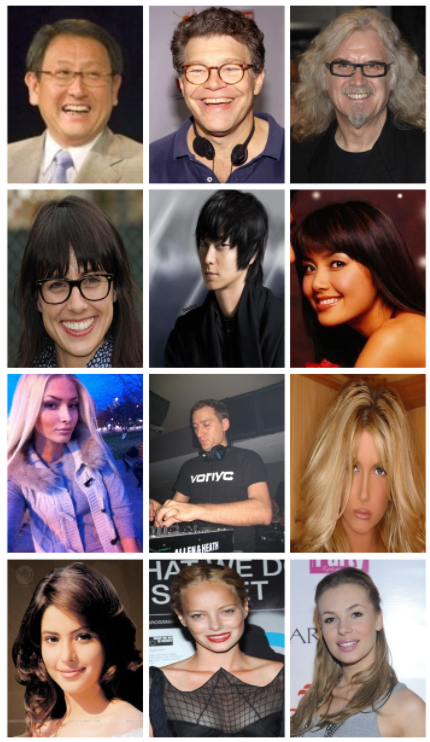
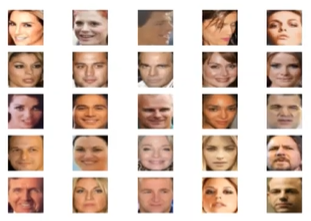
**Final Project – Machine Learning Course – Progressive GAN**  
  
**Students :**  
Noor Abu Abeid - 207755695  
Tali Amit - 312146749  
Saher Saadi - 208046383  
Eyal Romano - 311158448  
  
**Overview**  
  
Progressive Growing GAN is an extension to the GAN training process that allows for the stable training of generator models that can output large high-quality images.  
It involves starting with a very small image and incrementally adding blocks of layers that increase the output size of the generator model and the input size of the discriminator model until the desired image size is achieved.  
Progressive growing GAN models are capable of generating photorealistic synthetic faces and objects at high resolution that are remarkably realistic.  
  
**Input**   
  
Firstly, we take a lot of pictures (with background etc.) of celebrities from CelebA dataset.  
After some preparations we get a “clean” face pictures of all of those celebrities.  
  
   
  
**PG GAN Main Layers**  
  
1. Progressive Growing & Smoothing in of Higher Resolution Layers  
2. Minibatch standard deviation  
3. Equalized Learning Rate  
4. Pixel-wise Feature Normalization



Data preparation

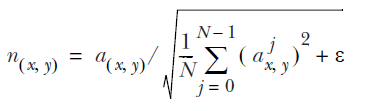
**1. Progressive Growing & Smoothing in of Higher Resolution Layers**

Used to control the weighted sum of the old and new layers during a growth phase.  
It uses a variable called *alpha* that controls how much to weight the first and second inputs

weighted sum = ((1.0 – α) \* input1) + (α \* input2) , 0 α 1   
  
**2. Minibatch standard deviation**  
  
We compute the standard deviation across all the images in the batch, across all the remaining channels-height, width, and color. We then get a single image with standard deviations for each pixel and each channel.

We average the standard deviations across all channels-to get a single feature map or matrix of standard deviations for that pixel.

We average the standard deviations for all pixels within the preceding matrix to get a single scalar value.  
  
**3. Equalized Learning Rate**  
  
The idea behind equalized learning rate is to scale the weights at each layer with a constant such that the updated weight w' is scaled to be w' = w /c, where c is a constant at each layer. This is done during training to keep the weights in the network at a similar range.  
  
**4. Pixel-wise Feature Normalization**Used to normalize activation maps in the generator model to unite the length which means that we use it when the training data and testing data does not come from the exact same distribution.

  
  
  
**PG\_GAN And CNN:**

N – number of feature maps  
(x,y) – vector  
Epsilon –

Progressive GAN is based on CNN architecture. In the generator \ discriminator we have a 1×1 convolutional layer. The output block involves a MinibatchStdev, 3×3, and 4×4 convolutional layers, and a fully connected layer that outputs a prediction. Leaky ReLU activation functions are used after all layers and the output layers use a linear activation function.

**Output**   
  
We have generated new faces that are not in our dataset of CelebA. Those pictures are fake people that don’t exist.  
  
A collage of a person's face

Description automatically generated