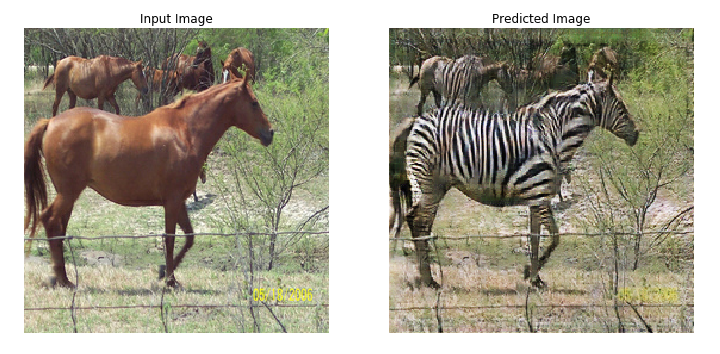
***Project: image to image translation***

**Objective:**

In this project, we are using artificial intelligence for transforming an input image into a synthetic image. We are taking test sets with pictures of horses and zebras, and after the training, our model can replace the horse in images with zebras.



**Model:**

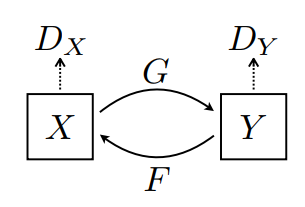
The model demonstrates unpaired image-to-image translation using Cycle GANs. Here, we have used the ‘Tensorflow” deep learning library.

Using TensorFlow we imported the dataset which contained the required images. Then we performed Image Augmentation to avoid overfitting i.e very high accuracy with training set results and low accuracy of the test set results) of the training set. The image augmentation includes jittering (resizing the images and cropping) and monitoring (images are flipped horizontally). This is done for better training.

After that, we have plotted one original image and the image after applying random jittering.

In the cycle GAN model, we have two generators (G and F) and two discriminators (X and Y).

* Generator G learns to transform image X to image Y. (G:X−>Y)
* Generator F learns to transform image Y to image X. (F: Y−>X)
* Discriminator D\_X learns to differentiate between image X and generated image X (F(Y)).
* Discriminator D\_Y learns to differentiate between image Y and generated image Y (G(X)).



One of the generators generates images of zebras, and the other generates images of horses. One discriminator learns to differentiate between real and fake zebra images, and the other model learns to distinguish between real and fake horse images.

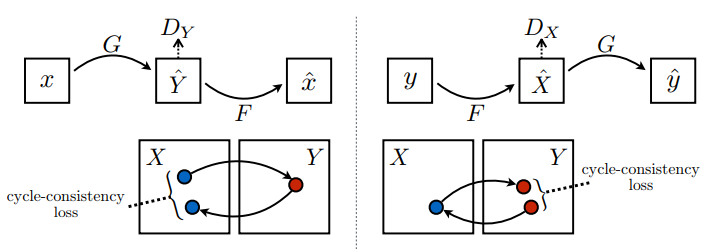
To enforce that the model makes correct predictions, we have used cycle consistency loss as the loss function, which calculates the errors and backpropagates them into the network, and takes required steps to predict accurate results.

In cycle consistency loss,

* Image X is passed via generator G that yields generated image Yˆ.
* Generated image Yˆ is passed via generator F that yields cycled image X^.
* The mean absolute error is calculated between X and Xˆ

forward cycle consistency loss: X−>G(X)−>F(G(X))∼Xˆ

backward cycle consistency loss: Y−>F(Y)−>G(F(Y))∼Yˆ.



Now the training of the model starts. Training consists of four basic steps: getting the predictions, calculating the loss, calculating the gradients using backpropagation, and applying gradients to the optimizer. Now, the training is completed. We can visualize the test set results and see the image transformation from horses to zebras and vice versa.

**Results**

****

The above image is the result when the model runs for 50 epochs.



This image is the result when the model runs for 90 epochs.

The accuracy of the model goes on increasing with the number of epochs.