Image Denoising and Upscaling using Different Types of Deep Learning Models

ASEF JAMIL AJWAD

19121040

asef.jamil.ajwad@g.bracu.ac.bd

SK TAHMED SALIM RAFID

19121028

sk.tahmed.salim.rafid@g.bracu.ac.bd

INTRODUCTION

The ever-expanding human knowledge dictates that security is ensured for protecting the civilizations towering throughout this planet. A subset of the methods used for establishing said security is video surveillance systems. Outdoor environments introduce unwanted noise to the outputs of such systems. Moreover, deep learning techniques such as “Face Recognition” are also quite popular in recent times due to their unmanned usability attributes. Due to this, the demand for algorithms that can make images clearer for facial recognition purposes in increasing.

MOTIVATION

The motivation that drives this project resides in the desire for increasing the efficiency of surveillance systems. This is seen through the vastly superior processing speeds of man-made processors that are far faster than human assessment. Next, human interaction is minimized through the introduction of deep learning architecture. Thirdly, we are aiming for a system that is reliable when it comes to image denoising, so that facial features can be extracted successfully. Lastly, the very backbone of this project rests on the tremendous potential in the sector of image processing itself. The astounding feats of the aforementioned sector are what has motivated us to proceed with this project.

DATABASE INFO

For our training, we have chosen to use the Flickr-Faces-HQ Dataset (FFHQ). The dataset consists of 52,000 high-quality PNG images at 512×512 resolution. There is considerable variation in terms of age, ethnicity and image background. It also has a variety of accessories such as eyeglasses, sunglasses, hats, etc.

METHODOLGY

We decided to trained two different models, one for Gaussian noise and one for upscaling. Each model would be trained using three different architectures. The training was carried out in Kaggle due to lack of computing power on our personal machines. Kaggle provides free access to NVIDIA P100 GPUs, with 16 GB of GPU Memory.

To train the models, datches of 32 images were feed in at a time. Initially we wanted a batch size of 128 images. However, due to memory limitations, we had to settle for 32 images. Our models were trained on 1000 batches, totaling 32,000 images. We ran the training for 3 epochs. We used 3 epochs as we noticed the model mostly converges after

2 epochs and the time per epoch is around 20 minutes.

Before the images were feed in the network, we performed some preprocessing using TensorFlow Pipeline. The image pixel values scaled between 0 and 1. Adam optimizer was used to train and minimize Mean Absolute Error (MAE) loss. We used a TensorFlow custom training loop and GradientTape. For the Gaussian noise training, the input image was made noisy by adding noise, with standard deviation of 0.4. For the upscaling model, the input image resolution was lowered to a resolution of 64x64. The network was then tasked to process the inputs to match the unmodified images.

Performance was evaluated using Peak Signal-to-Noise Ratio (PSNR), Mean Absolute Error (MAE) loss and Structural Similarity Index Measure (SSIM).

NETWORK ARCHITECTERES

The three different architectures we used were inspired from GoogLeNet, ResNet and UNet. From the GoogLeNet, we used to Inception module without the Max Pooling layer. For the ResNet model, we used convolutional layers with residual connections. As for the UNet model, we used add layers in place of concatenate layers.

RESULTS

We analyzed the performance of each network by using 5 test images. The results are as follows.

TABLE I: PNSR FOR NOISY IMAGE TEST

Image	Input	GoogLeNet	ResNet	UNet
1	10.37	24.37	24.77	25.00
2	10.38	24.60	25.40	24.29
3	10.28	25.74	26.67	25.62
4	10.10	25.55	25.95	25.22
5	10.14	27.03	27.73	27.65

TABLE II: MAE FOR NOISY IMAGE TEST

Image	Input	GoogLeNet	ResNet	UNet
1	0.239	0.045	0.042	0.041
2	0.236	0.044	0.040	0.045
3	0.241	0.039	0.034	0.038
4	0.258	0.040	0.038	0.041
5	0.253	0.033	0.031	0.031

TABLE III: SSIM FOR NOISY IMAGE TEST

Image	Input	GoogLeNet	ResNet	UNet
1	0.09	0.77	0.79	0.82
2	0.08	0.78	0.80	0.80
3	0.07	0.84	0.87	0.88
4	0.06	0.81	0.82	0.83
5	0.06	0.81	0.87	0.88

TABLE IV: PSNR FOR UPSAMPLE IMAGE TEST

Image	Input	GoogLeNet	ResNet	UNet
1	20.89	23.10	23.05	22.83
2	22.47	24.78	24.62	23.53
3	23.38	26.64	26.45	25.24
4	23.04	26.11	26.09	24.67
5	24.37	27.51	27.41	27.55

TABLE V: MAE FOR UPSAMPLE IMAGE TEST

Image	Input	GoogLeNet	ResNet	UNet
1	0.058	0.044	0.044	0.046
2	0.045	0.035	0.036	0.042
3	0.038	0.027	0.029	0.033
4	0.047	0.032	0.032	0.040
5	0.033	0.024	0.026	0.025

TABLE VI: SSIM FOR UPSAMPLE IMAGE TEST

Image	Input	GoogLeNet	ResNet	UNet
1	0.69	0.76	0.76	0.77
2	0.76	0.80	0.81	0.80
3	0.85	0.88	0.89	0.89
4	0.78	0.84	0.84	0.83
5	0.85	0.89	0.89	0.90

From our tests, we can see that all three models have reduced the noise and somewhat upscaled the image with some clarity and sharpness. For the upscaling, the performance of all three models were similar. As for the denoising models, UNet performed slightly better than the other 2 models. The image outputs for the tests are given below.

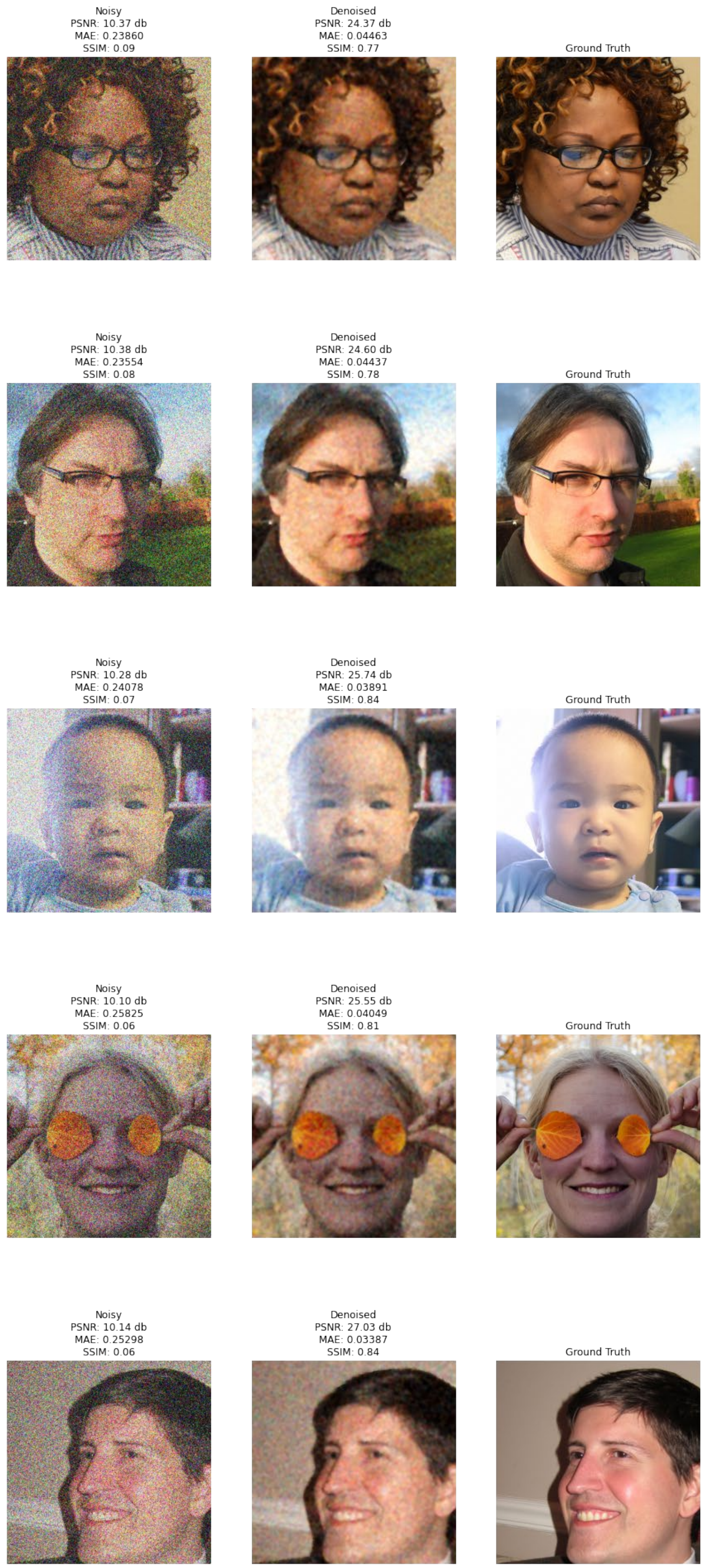
REFERENCES

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

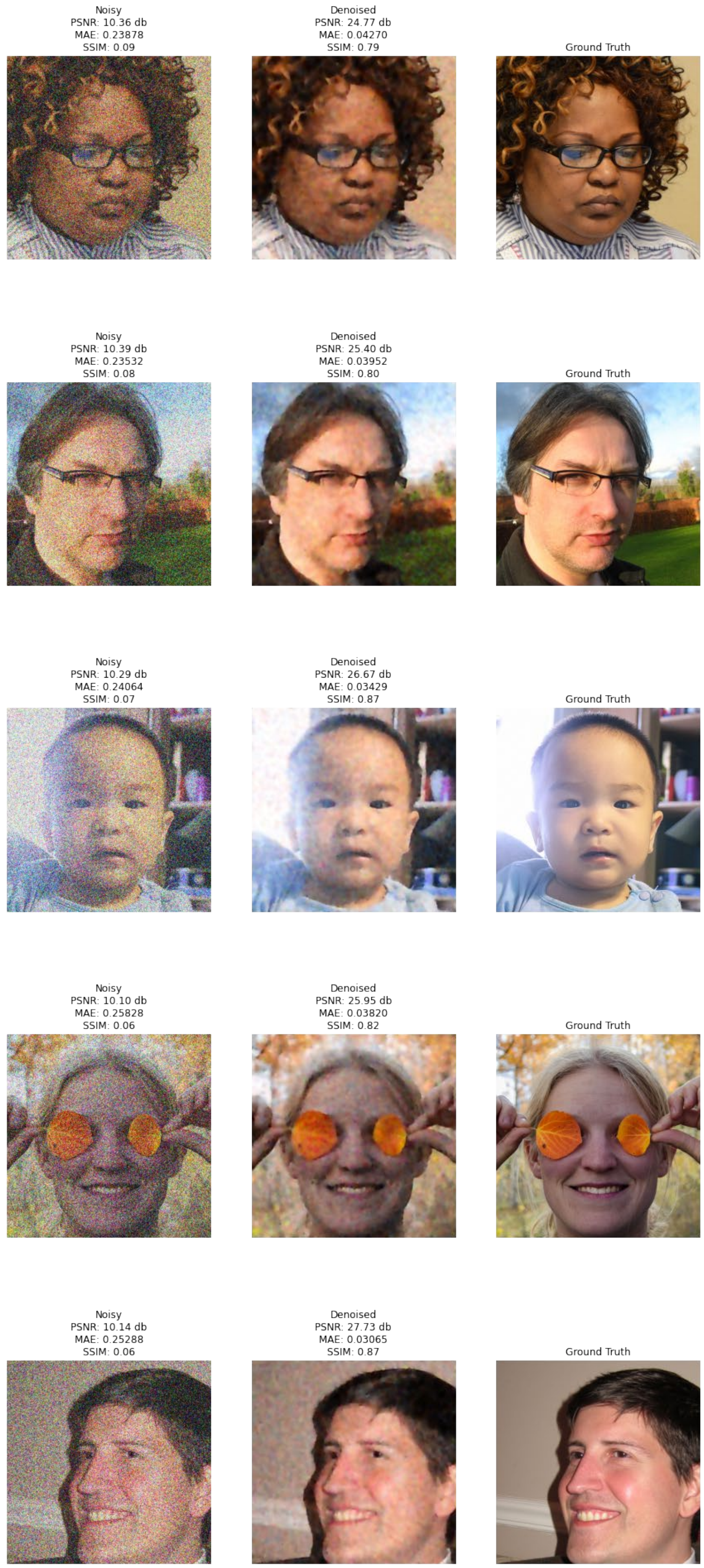
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

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.

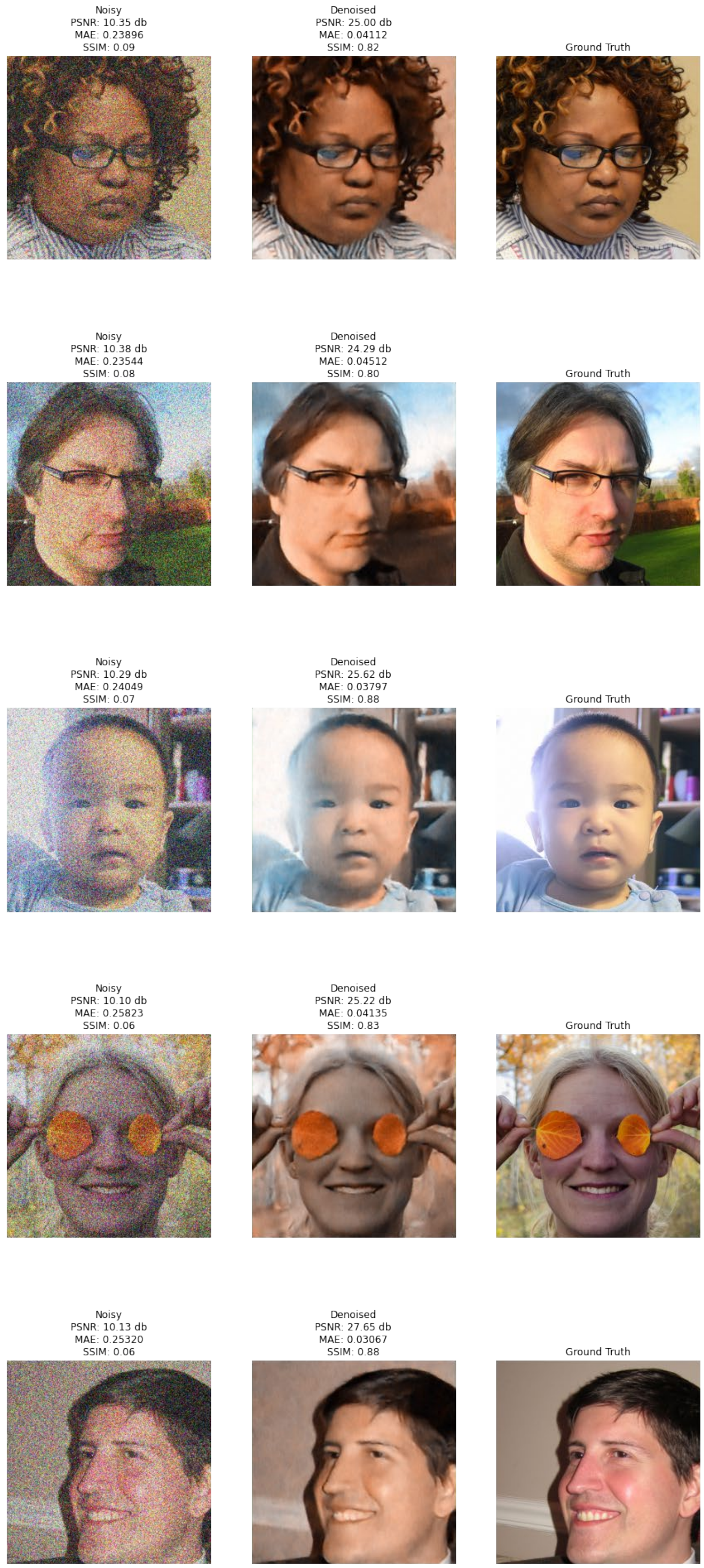
Results for denoising using Inception model



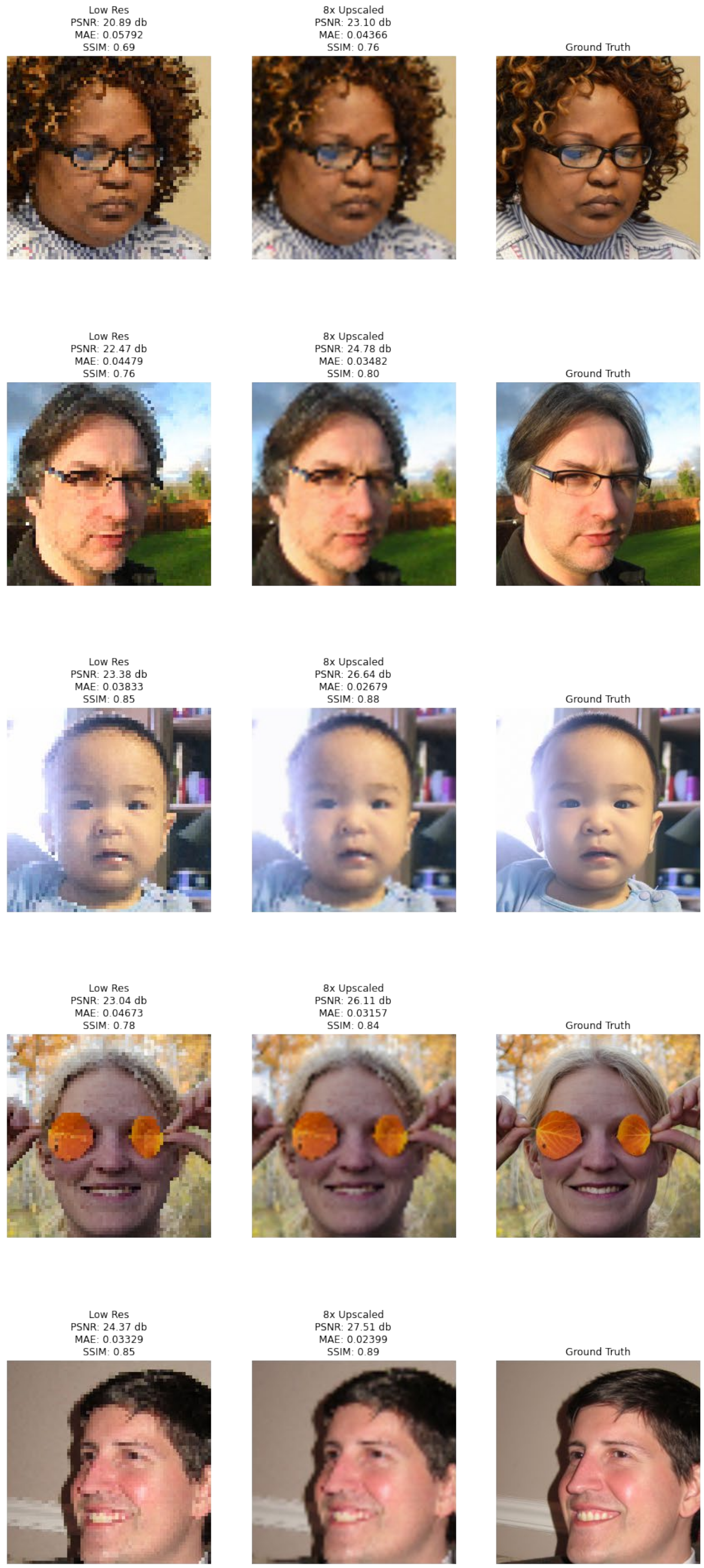
Results for denoising using ResNet model



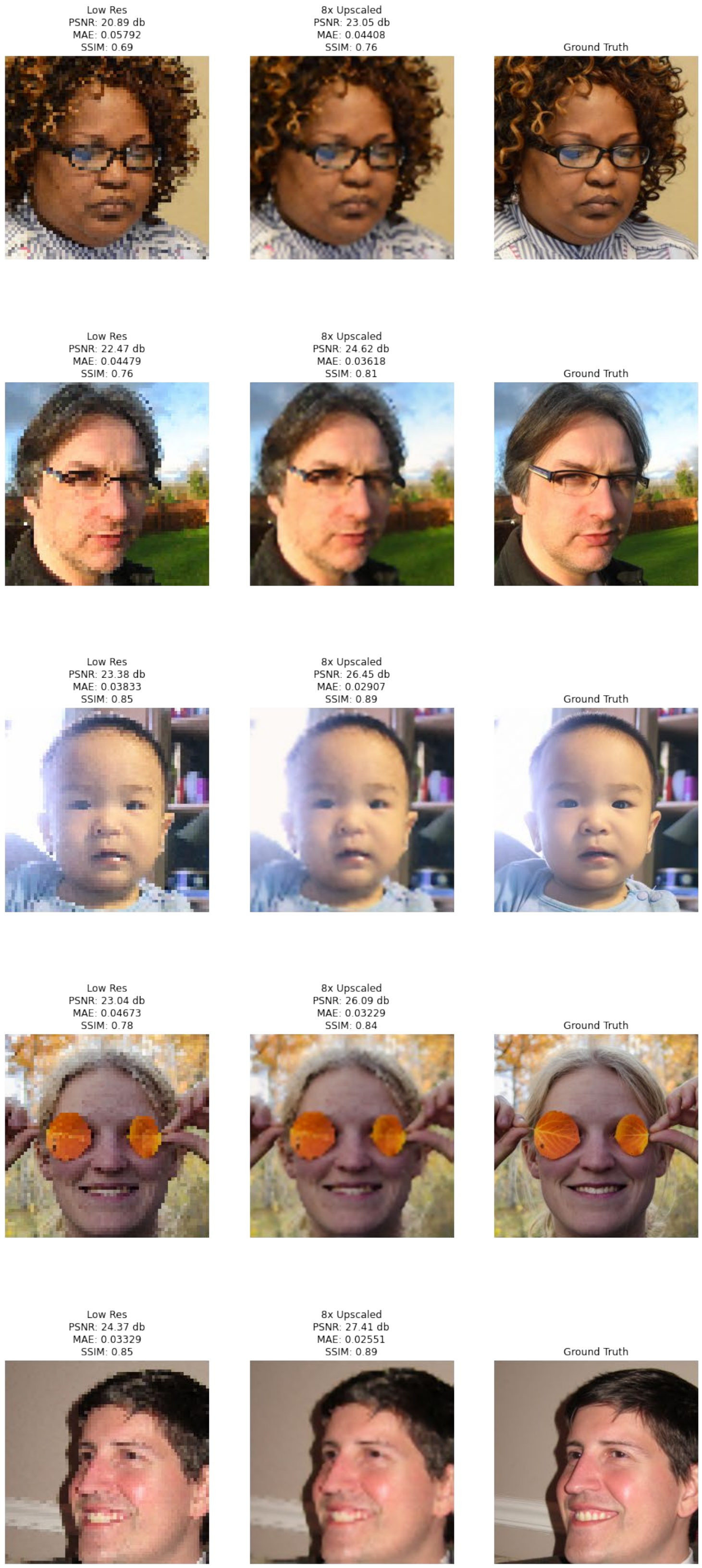
Results for denoising using UNet model



Results for 8x upscaling using Inception model



Results for 8x upscaling using ResNet model



Results for 8x upscaling using UNet model