EE5179 - Image Denoising

Chevvuri Karthik (EE20B026)
A Sudhan (CH20B104)
Mihir singh (CH20B065)

Github Link: Github

U-Net

The U-Net architecture implemented is known for its effective image-to-image translation capabilities. It was originally designed for biomedical image segmentation but has since been widely applied in image restoration tasks.

Architecture Overview
This U-Net model follows the classic encoder-decoder structure, with a contracting (encoding) path that captures context and a symmetric expansive (decoding) path that facilitates precise localization. Each layer in the encoding path has a corresponding layer in the decoding path, and skip connections link the encoding and decoding stages, helping retain spatial details. Here's a breakdown of each component in the architecture:

Encoding Path:
The encoding path consists of five convolutional blocks. Each block contains two convolutional layers, each with a kernel size of 3x3, stride of 1, and padding of 1. ReLU activations are used to introduce non-linearity.
As we progress through the layers, the number of feature channels doubles with each block, starting from 64 up to 1024 channels. This increase allows the model to capture complex patterns and hierarchies of features.
MaxPooling layers (with a kernel size of 2 and stride of 2) follow each block, downsampling the spatial dimensions by half. This downsampling reduces resolution, allowing the model to focus on larger receptive fields.

Bottleneck Layer:
The final layer in the encoder, with 1024 channels, serves as the bottleneck, capturing the most compressed representation of the input image. It contains detailed high-level features, albeit with limited spatial information.

Decoding Path:

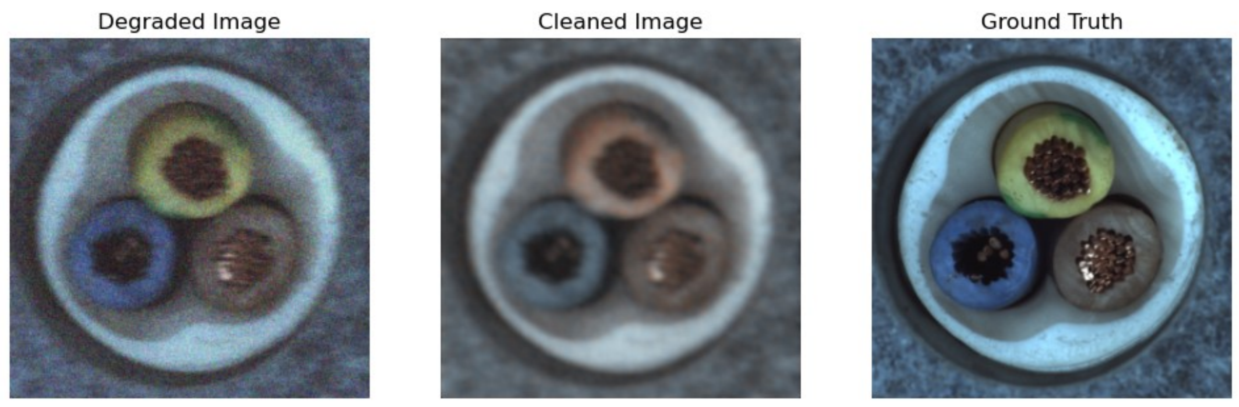
The decoding path uses transpose convolutional layers to upsample the feature maps, gradually reconstructing the image.
Each upsampling layer doubles the spatial dimensions, restoring finer details to the images.
Skip connections concatenate the upsampled feature maps with feature maps from the corresponding encoding layer. This allows the network to utilize both high-level and low-level features from the encoder, enhancing restoration quality by preserving spatial information.
Each decoder block uses two convolutional layers with ReLU activations, similar to the encoding path.

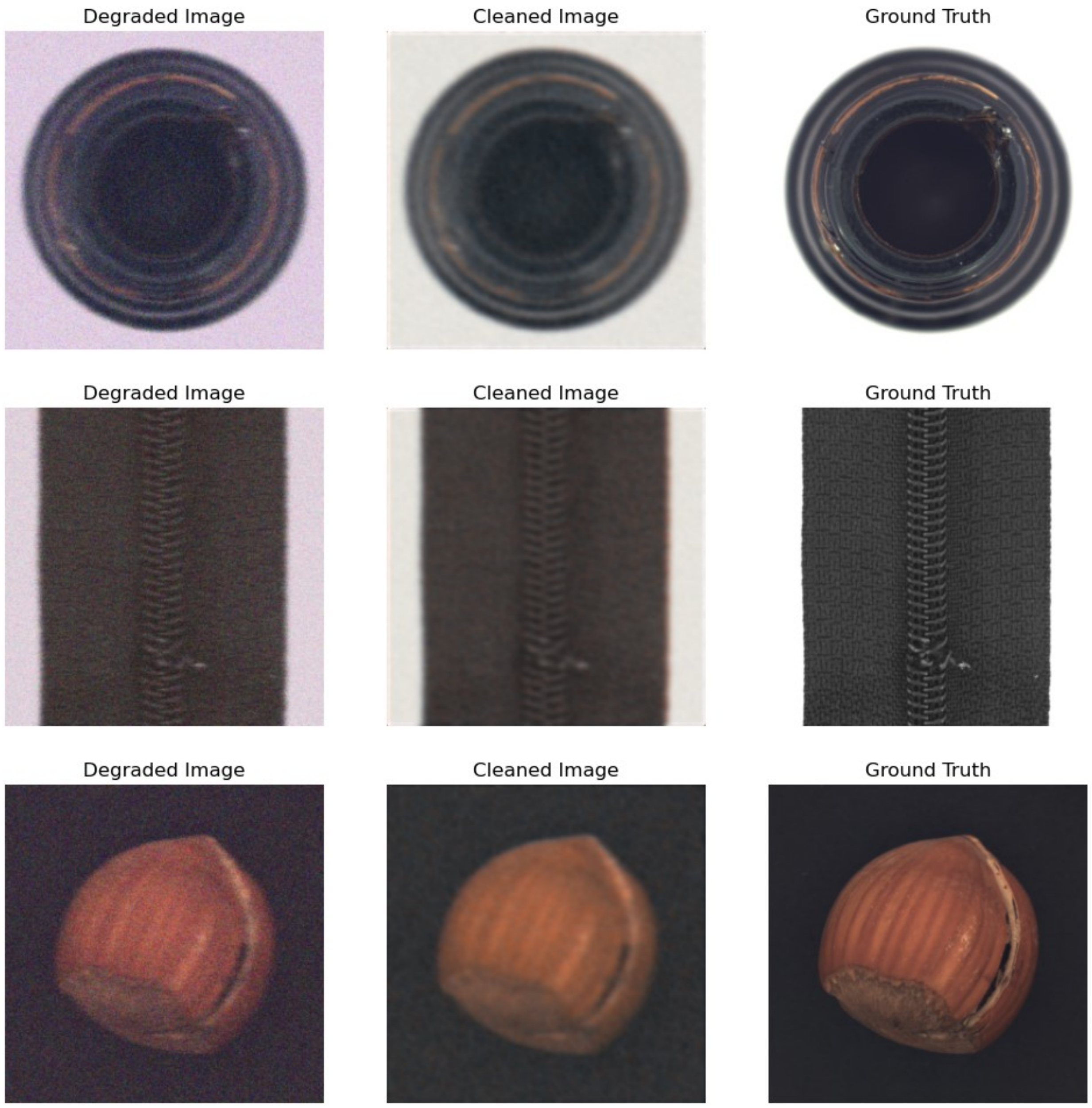
Final Convolution Layer:
The final convolution layer reduces the output channels to 3, matching the RGB channels of the input image. This layer produces the restored image output with the same dimensions as the original input.

Application to Image Restoration
In image restoration, the goal is to reconstruct a clean image from a degraded one. This U-Net is particularly well-suited for this task due to its:

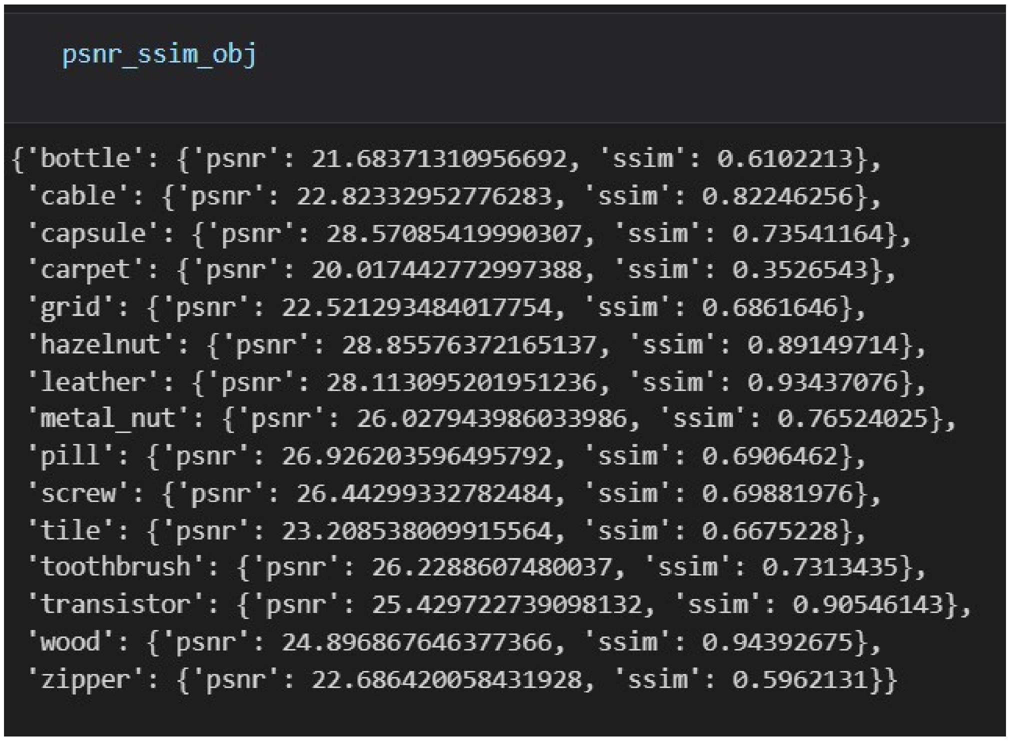
- **Skip Connections**: They enable the network to preserve fine-grained spatial information from earlier layers, crucial for reconstructing sharp and accurate images.
- **Symmetric Encoder-Decoder Structure**: This structure captures both global context (for recognizing larger patterns and objects) and local details (important for reconstructing textures and edges), both essential for denoising and deblurring.
- **Upsampling and Convolutional Blocks**: The network combines upsampling and convolutional blocks, progressively refining and reconstructing details at each stage, ideal for restoring spatial details lost in noisy and blurry images.

Some Output Images:



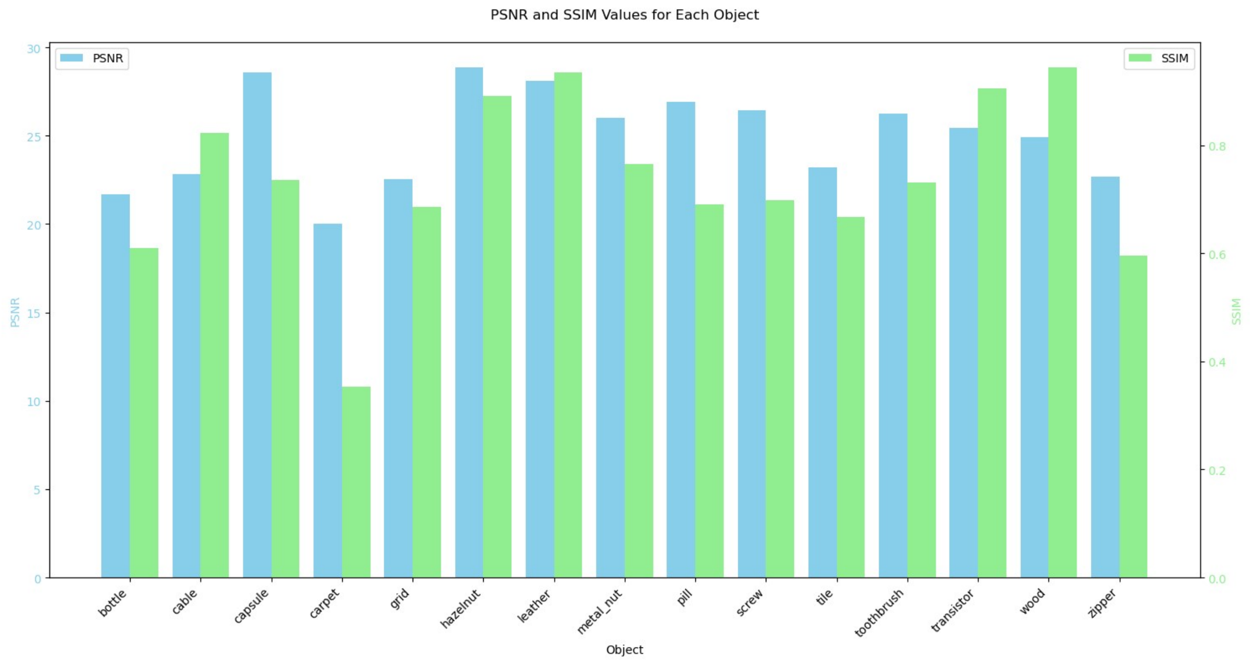


PSNR, SSIM Values for Each object:



```
psnr_ssim_obj
```

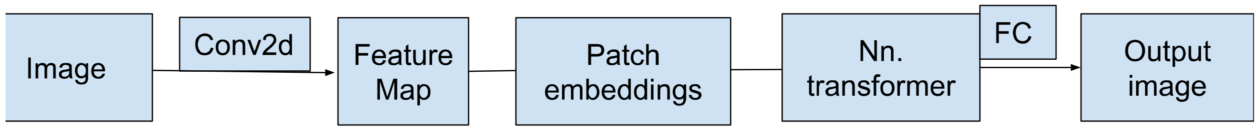
```
{'bottle': {'psnr': 21.68371310956692, 'ssim': 0.6102213},
 'cable': {'psnr': 22.82332952776283, 'ssim': 0.82246256},
 'capsule': {'psnr': 28.57085419990307, 'ssim': 0.73541164},
 'carpet': {'psnr': 20.017442772997388, 'ssim': 0.3526543},
 'grid': {'psnr': 22.521293484017754, 'ssim': 0.6861646},
 'hazelnut': {'psnr': 28.85576372165137, 'ssim': 0.89149714},
 'leather': {'psnr': 28.113095201951236, 'ssim': 0.93437076},
 'metal_nut': {'psnr': 26.027943986033986, 'ssim': 0.76524025},
 'pill': {'psnr': 26.926203596495792, 'ssim': 0.6906462},
 'screw': {'psnr': 26.44299332782484, 'ssim': 0.69881976},
 'tile': {'psnr': 23.208538009915564, 'ssim': 0.6675228},
 'toothbrush': {'psnr': 26.2288607480037, 'ssim': 0.7313435},
 'transistor': {'psnr': 25.429722739098132, 'ssim': 0.90546143},
 'wood': {'psnr': 24.896867646377366, 'ssim': 0.94392675},
 'zipper': {'psnr': 22.686420058431928, 'ssim': 0.5962131}}
```

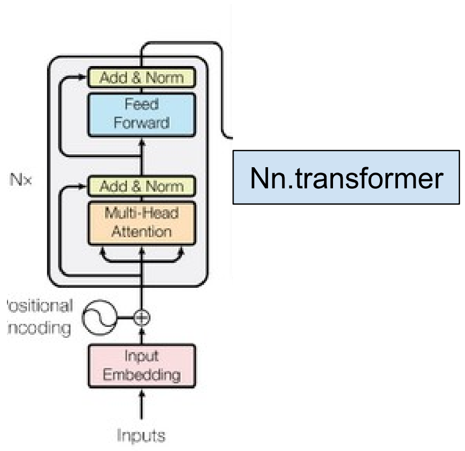


Denoising the image based on vision transformer:

Architecture:

Similar to denoising auto encoder for the encoder block we replaced the encoder with nn. transformer(vision transformer) for feature extraction and fc for reconstruction of the outputs





Architecture Overview:

Multi headed attention:

Multi-head attention allows the model to focus on different parts of the sequence by computing attention scores multiple times (each as a separate "head"). Each head independently captures unique relationships within the data, which are then concatenated and linearly transformed.

Positional Encoding

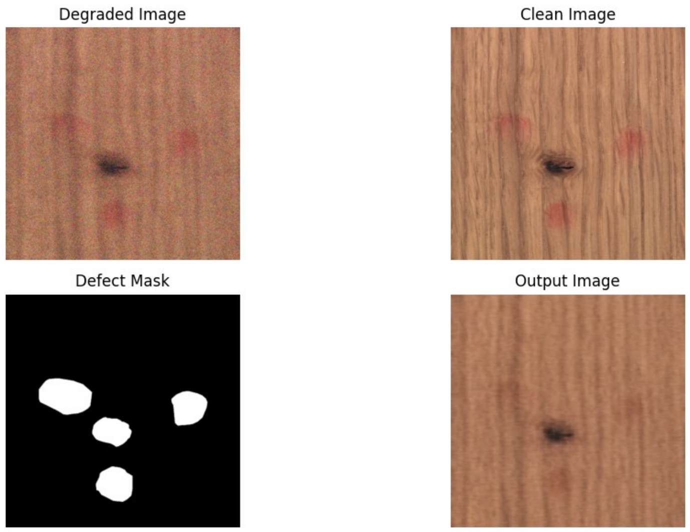
The Transformer lacks a sequential nature, so positional encodings are added to embeddings to provide order information. Positional encodings are sinusoidal or learned values that are unique for each position in the sequence.

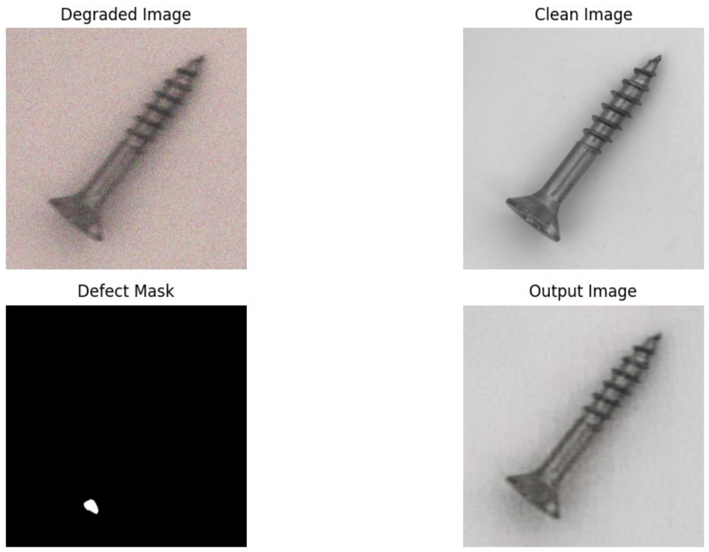
Overall Flow

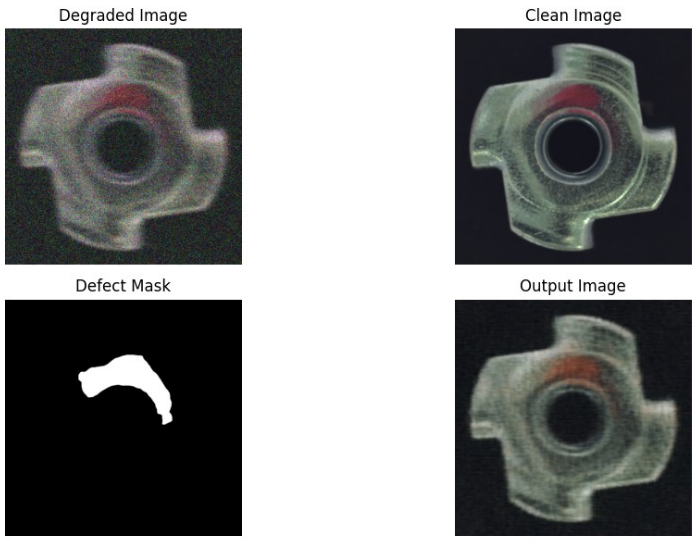
1. **Input** is embedded and positional encodings are added.
2. **Encoder Layers** process the embedded input through attention and feedforward layers, creating high-level representations.
3. **Decoder Layers** attend to both previous decoder states and encoder outputs to generate output sequences one token at a time.
4. **Output image** is generated by applying a linear layer and softmax function over the decoder output to obtain the probability of each token in the vocabulary.

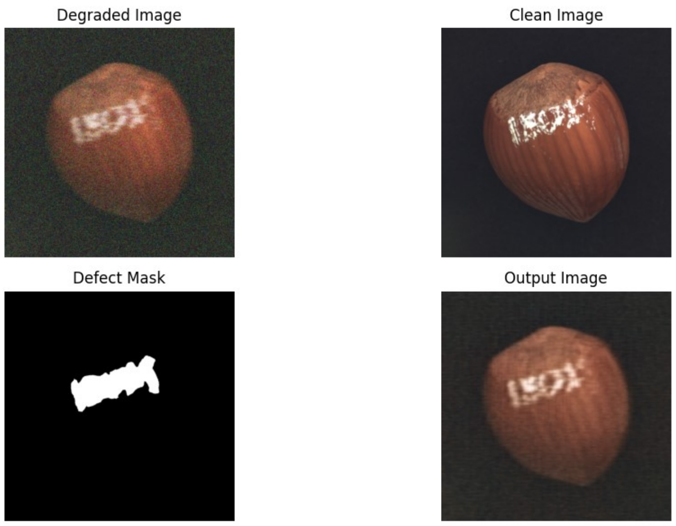
Results:
Average psnr score for validation dataset: 28.100744
Average SSIM index for validation dataset: 0.617031

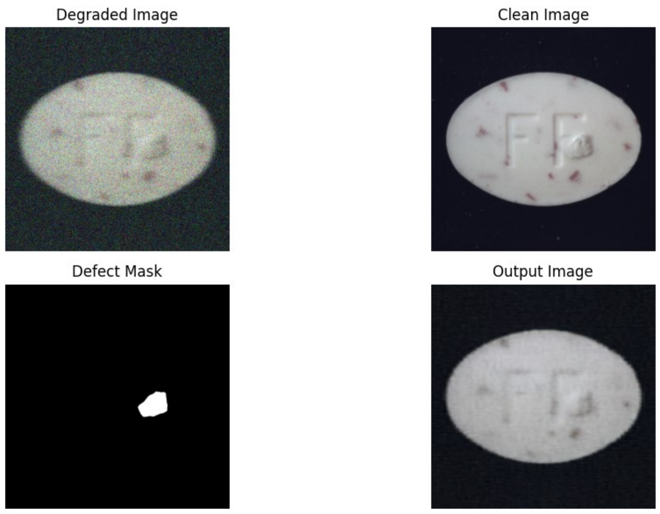
Sample Images for each class:

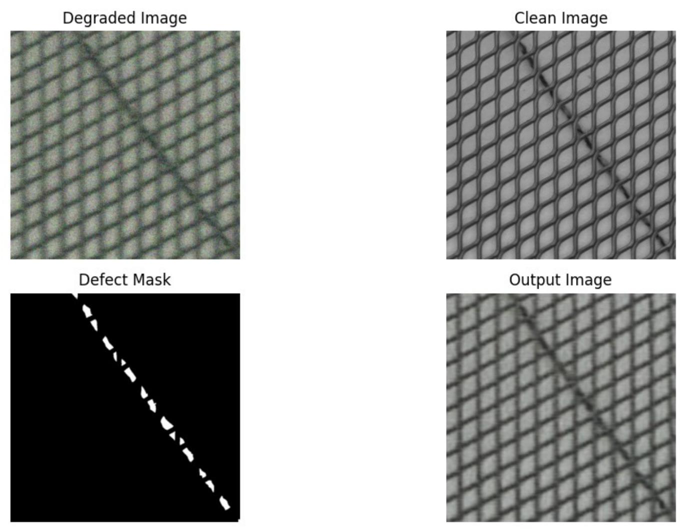


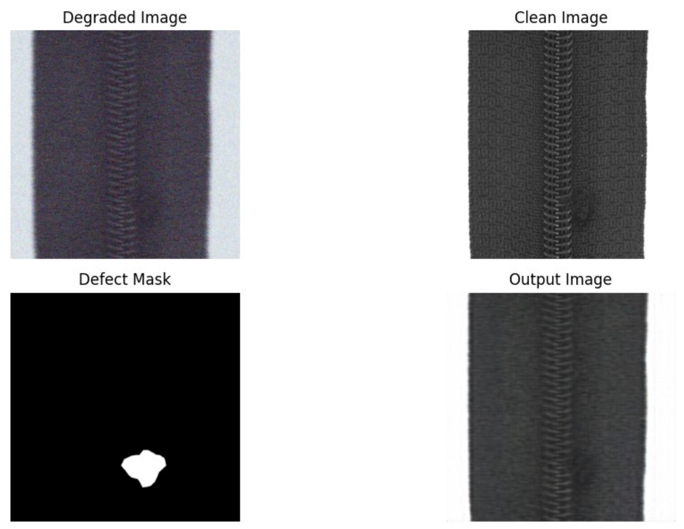



Degraded Image
Clean Image
Defect Mask
Output Image


Degraded Image
Clean Image
Defect Mask
Output Image


Degraded Image
Clean Image
Defect Mask
Output Image





Conclusion:

Compared to unet architecture, the vision transformer based denoiser is working slightly better because of the ability of the vision transformer to extract and understand global and complex features because multi headed attention. The vision transformer needs more data for better performance and the vision transformer based denoising auto encoder can be made better by adding more transformer blocks in both encoder and decoder.