Problem Statement

DETECT PIXELATED IMAGES AND CORRECT IT

Unique Idea Brief (Solution)

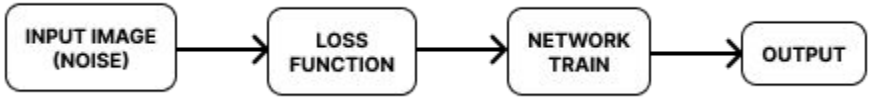
The model does not require a dataset for training for recovery of pixelated images. It is able to detect images that have blur, pixelation, noise and if they have other graphics superimposed onto them(e.g. shapes and text). The model uses inpainting to recover the part of the image that is lost.

Features Offered

- Detects all forms of noise such as pixelation, superimposed graphics
- Graphical representation of error with respect to the data
- Offers a NRMSE(Normalized Root Mean Squared Error) of 10%-20%

Process flow

- Input Image has gaps and noise
- Loss Function used to compare with output
- NRMSE calculated by comparing each pixel with original input image
- Input is mapped to the output for approximating the image to the true value
- Output



Architecture Diagram

Network architecture design(Encoder decoder)
Layers and Filters (Convolutional layer)
Activation Function (Non - Linear Activation)
Random Initialization Loss Function
No pre training or optimisation required

Technologies used

Language used - Python

Libraries used -

- NumPy
- Matplotlib
- OpenCV
- Scikit-Image

Team members and contribution:

G Dhvani Iyer - Worked on image recovery

Akshat - Worked on noise/pixelation detection

Conclusion

The approach using Deep Image Prior (DIP) and CNNs demonstrates significant potential for JPEG restoration, effectively reconstructing high-quality images from noisy inputs without extensive training datasets. The methodology harnesses the implicit regularization properties of CNNs, making it particularly effective for image denoising, inpainting, and super-resolution tasks. The implementation in Python, leveraging libraries such as PyTorch, ensures efficient neural network construction and training.