Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

Goals and Objectives

Motivation:

- Our motivation for doing this project is to denoise the image using the advanced neural - networks models instead of using all the existing filters.
- The goal is to denoise the color images using blind noise.
- The main goal of this project is to denoise the image using deep learning architectures rather than using any conventional techniques, like filters on the image. Since I want to denoise the images as closely as possible to the real world, even if it takes some time, there is no latency restriction.
- Loss of Pixels equals Loss of Data is the driving principle behind the development of this image d enoising technique.
- By employing conventional strategies like filters, we are wasting more data where it is unnecess ary to do so.
- They used to function fairly well for images with moderate noise levels.
- However, using such effects would produce a picture that was specific to that field.
- Additionally, if the image is very fuzzy, many of the item's essential parts will be hidden by the fu zziness of the final product.
- As a result, we are avidly investigating completely novel ways to defeat this strategy.

Significance :

- With existing filters, we may lose some important data to avoid having to use different models like Restormer model, DnCNN.
- We increase the Accuracy of our model to denoise the images
- In this project we have implemented the pure neural network models for extracting the features without any loss of information of the images
- Image denoising is a key component in many important applications. such image processing, segmentation, classification, and picture restoration. For all of these, we require real image pixels or data in order to perform effectively. Therefore, it's crucial to extract the image with no

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

noise for precise outcomes in all digital applications.

Objectives:

- Retrieve the Blur or distorted images.
- Get the Denoised image using the pretrained models and architectures.
- Compare the difference between the various CNN models
- For a variety of applications, such as image segmentation, visual tracking, and image registration, it is essential to acquire images that are free of noise.
- In order to avoid pixel loss and image data loss
- to prevent textured images that are fuzzy and blurry after applying filters

Related Work :

Spatial domain filtering is a traditional de-noising method. Further divisions into linear and non-linear filters are made. By employing weighted-median and median filtering, noise is reduced in non-linear filters. For producing photos with no noise, bilateral filtering is also frequently utilized.

The drawback of linear filtering is that image textures are not preserved. Gaussian noise reduction is then applied, but the images are oversmoothed as a result. Wiener filtering is suggested as a solution, but it produces extremely blurred sharp edges. Gone through various research papers and analyzed which models best.

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

Model Architecture:

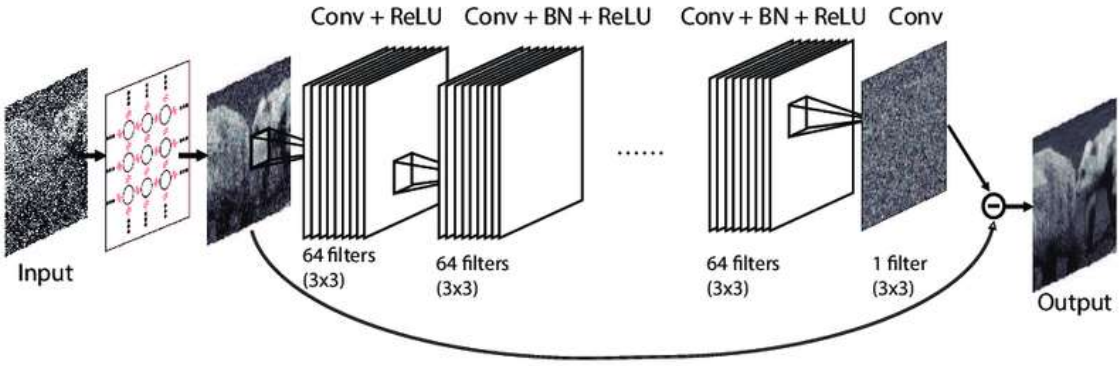

Fig: DnCNN Architecture.

We input the original and noise images to the neural network layer. Our architecture mainly describes how noisy images are filtered. In the above diagram the noisy images are converted to the matrix values and then it assigns the noisy image and original to different classes, using the functions we acquire all the deviating values that are affecting the naturality of the picture. These images are processed as patches instead of processing the whole image, to retrieve the accurate reconstructed pixels of images. These are divided into a number of batches.

This helps our model to work swiftly, without losing the useful and highly intense information. We pass these patches through nearly 64 filters with Convolutional and Relu Activation to acquire the positive information by rejecting the negative information. Every image patch is 3*3 matrix. After this these images will be normalized using the Batch Normalization. This process is repeated until the neural layers achieve the best accurate value.

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

Entire filtered images are passed through the convolutional layer and we retrieve a Residual image, which is the variance values between the real and noisy image. Internally all these images are introduced to orientation, translation, illumination changes. Using the functions, we subtracted all those residuals from the noisy and reconstructed the original transformed image. We used the reformer to construct these noisy images to its normal more efficiently. Therefore, we constructed the original by denoising.

Dataset:

Detailed description of Dataset & Detail design of Features with diagram:

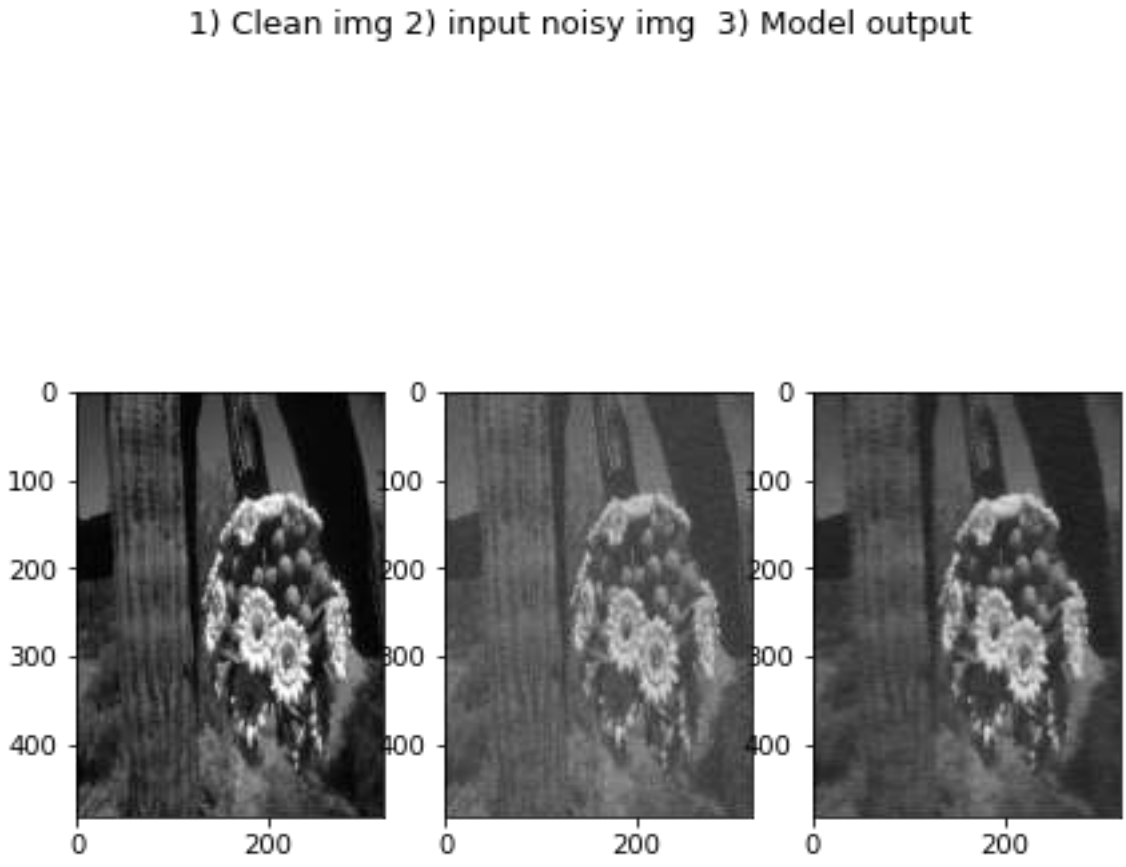

Fig: Sample Dataset

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

Used the SIDD image dataset, which has 160 images of data in that it contains all the normal images with Ground truth images and Noisy images. The Ground truth images are nothing but the real images.

- SIDD-Small Dataset: This dataset contains 160 image pairs (which consists of dependent and independent images or we can say noisy and ground-truth images). Both noisy image and ground-truth image are Gamma corrected which are without tone mapping.
- Also, SSID has the 3 as one so combined large data set.
- It has real images and noisy images including all the blurred and distorted images.
- Our Dataset helps to make the model understand and learn the original images and noisy images.
- We have images of animals, humans, things, nature photos.
- This dataset has specific characteristics for each, which enables the DncNN model to learn more deeply about constructing the Residual images.

Features:

- Using the Model DnCNN which is Convolutional neural network used for denoising the images with restore model.
- Classifying the image data into different groups.
- Extract the main features of the picture and reduce the noise.
- Internally the restorer compares the noisy and Ground truth images with help of graphs and again neural network layers draws a conclusive residual image graph to map and reshape the data to acquire the reformed original image.
- Training and testing the truth and distorted images.

Analysis of data:

Data Pre - processing:

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

- We selected the data which already preprocessed and usually this data used for experimenting the new custom and pretrained models.
- We arranged the data and spitted the data into train and test. These Images are divided into batches with size of 32.
- Used the activation called “RELU” with an optimizer to enhance the working of the functions.
- Used the Learning Rate of: 0.0001, Power : 0.9.
- Used Feed Forward and Feed Backward Layers to work on retrieving the residual images.

Graph model with explanation:

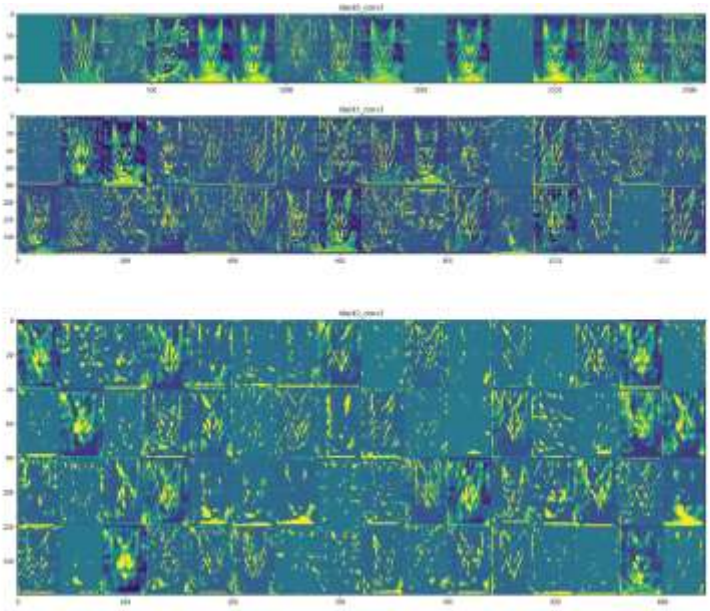

Fig: Spectrogram for original Noisy data

- We compared the original images residual images and restored images in the above representation.
- Different patches of images are aligned and compared the reconstructed images using the spectrum.

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

Implementation :

DnCNN :

DnCNN is Convolutional neural network to denoise the images. This helped us Identify and reduce all the residuals values in the noisy image. It helped us reduce the residual by comparing it to the original images to noisy images. We applied max pooling layers to the images to retrieve the actual information from the original image and validated the values to values of the noisy pictures.

Pooling the actual information gave a few vectors of the features and we sampled these vectors using the neural networks internally to multiply the vectors of the neural filter to the noisy images to retrieve the originality of the picture.

Explanation of implementation :

DnCNN is used in the Restormer model which helps various types of image filtering and image classification. The restormer model helps the system to reform the structured real image. Before implementing the Restormer model with DnCNN, we read all the files of the pictures and mapped them specified channels where we use functions to differentiate the We trained and tested the data .We decoded all images and then processed the images by allocating these pixel values to different channels to clean the images by applying mathematical functions internally by assigning the depth as 5. Here we have chosen the learning rate 0.0001 with beta value of 0.9, batch size is 32. After processing and cleaning, reshape and reconstruct the images using the filtered Vectors.

We input the original and noise images to the neural network layer. Our architecture mainly describes how noisy images are filtered. In the above diagram the noisy images are converted to the matrix values and then it assigns the noisy image and original to different classes, using the functions we acquire all the deviating values that are affecting the naturality of the picture. These images are processed as patches instead of processing the whole image, to retrieve the accurate reconstructed pixels of images. These are divided into a number of batches. This helps our model to work swiftly, without losing the useful and highly intense information. We pass these patches through nearly 64 filters with Convolutional and Relu Activation to acquire the positive information by rejecting the negative information. Every image patch is 3*3 matrix. After this these images will be normalized using the Batch Normalization. This process is repeated until the neural layers achieve the best accurate value. Entire filtered images are passed through the convolutional layer, and we retrieve a Residual image, which is the variance values between the real and noisy image. Internally all these images are introduced to orientation, translation, illumination changes. Using the functions, we subtracted all those residuals from

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

the noisy and reconstructed the original transformed image. We used the reformer to construct these noisy images to its normal more efficiently. Therefore, we constructed the original by denoising. We got best results for these models compared to previous models (RedNet,MwCNN etc.) which worked in the increment 1.

Results:

Noisy and Blurred Image

Denoised and Deblurred Image

Noisy and Blurred Image

Denoised and Deblurred Image

Preliminary Results:

There are many models which only work on gaussian noise, their models achieved better results, but

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

here we have achieved the better results for various noisy images with help of the Restormer and DnCNN model.

Project Management:

Work completed:

Designed and worked on the DnCNN and Restormer Model. In this the Entire filtered images are passed through the convolutional layer and we retrieve a Residual image, which is the variance values between the real and noisy image. Internally all these images are introduced to orientation, translation, illumination changes. Using the functions, we subtracted all those residuals from the noisy and reconstructed the original transformed image. We used the reformer to construct these noisy images to its normal more efficiently.

Therefore, We constructed the original image by denoising through DnCNN and Restormer with High Resolution.

Responsibility (Task, Person):

Data Preprocessing and SIDD Dataset: Roshan Sah
Data Preprocessing and DnCNN Model Implementation :Saisri Teja Pepeti
Restormer Model for denoising and Deblurring : Yamuna Bollepalli
SIDD dataset preprocessing and Restormer Inference: Guduru Charan chand

Contributions (members/percentage) :

Roshan Shah - 25**%**
Saisri Teja Pepeti - 25%
Yamuna Bollepalli- 25%
Guduru Charan chand - 25%

Responsibilities(Task,Person) :

Data Preprocessing and Model Implementation and Report Writing :Roshan Sah
Data Preprocessing and Model Implementation and Report Writing :Saisri Teja Pepeti

Project Title: Image Denoising

Increment -2

Git : https://github.com/ErRsah/CSCE5222_Image-De-noising-Using-Deep-Learning.git

Team Members:

Roshan Sah [11574385]
Saisri Teja Pepeti [11555656]
Yamuna Bollepalli [11552426]
Guduru Charan Chand [11559174]

Data Preprocessing and Model Implementation and Report Writing :Yamuna Bollepalli
Data Preprocessing and Model Implementation and Report Writing: Guduru Charan chand

References

1. *https://medium.com/@polanitzer/building-a-cnn-based-autoencoder-with-denoising-in-python-on-gray-scale-images-of-hand-drawn-digits-61131ec492e4*
2. *https://www.analyticsvidhya.com/blog/2021/07/image-denoising-using-autoencoders-a-beginners-guide-to-deep-learning-project/*
3. *https://arxiv.org/abs/2107.14803*
4. *https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3791105*
5. *https://github.com/cszn/DnCNN (Trail and Error)*
6. *https://openaccess.thecvf.com/content_cvpr_2018/papers/Lefkimmiatis_Universal_Denoising_Networks_CVPR_2018_paper.pdf*
7. *https://sh-tsang.medium.com/review-dncnn-residual-learning-of-deep-cnn-image-denoising-super-resolution-jpeg-deblocking-cbf464b03130*
8. *https://vciba.springeropen.com/articles/10.1186/s42492-019-0016-7*
9. *https://uwaterloo.ca/vision-image-processing-lab/research-demos/image-denoising#:~:text=Therefore%2C%20image%20denoising%20plays%20an,is%20crucial%20for%20strong%20performance*
10. *https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/*
11. *https://commons.wikimedia.org/wiki/Natural_Image_Noise_Dataset#Tools*
12. *https://www.eecs.yorku.ca/~kamel/sidd/dataset.php*
13. *https://adrianbarburesearch.blogspot.com/p/renoir-dataset.html*

Final project video link: