# 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 denoising technique.
* By employing conventional strategies like filters, we are wasting more data where it is unnecessary 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 fuzziness 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 RedNet Model, 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 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.

# Dataset :

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
* RENOIR: The noise levels in the example photographs range widely, making it challenging for a noise removal method to choose the right hyperparameters for each object.
* NIND: To encourage us to submit pictures for a more extensive and diversified crowdsourcing project, we will take 62 pairs of noisy and ground truth shots using the Fujifilm X-T1 and XF18-55mm.

# Features:

* Using the Model REDNet which is Convolutional neural network auto- encoding architecture.
* Classifying the image data into different groups.
* Extract the main features of the picture and reduce the noise.
* Comparing the noisy and Ground truth images with help of graphs
* Training the truth and distorted images.

# Analysis of Implementation:

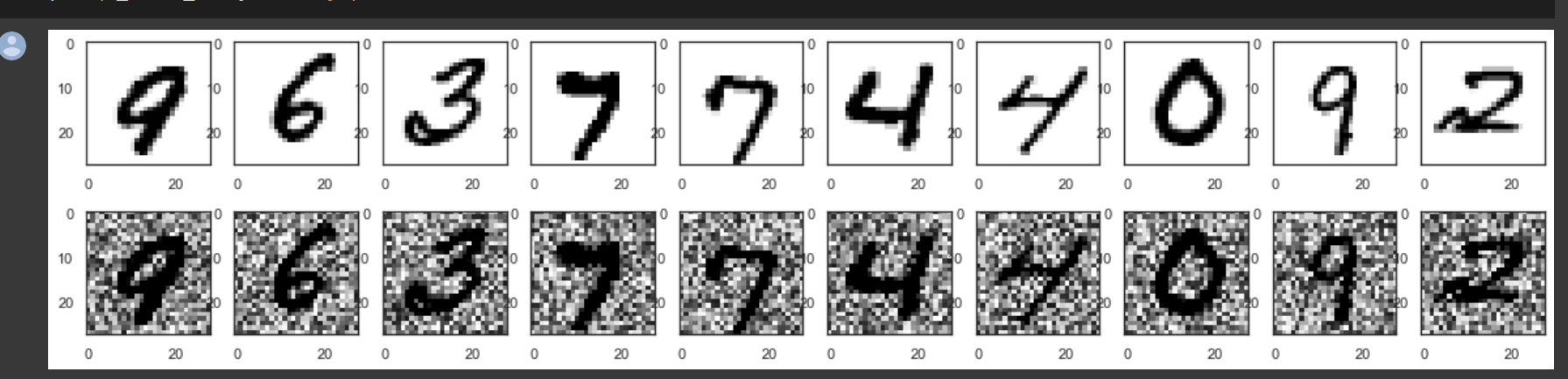
To Tackle the problem of noisy images we have Selected the specific type of dataset(SSID) for which we can use all the advanced Neural network architectures. We have U-Net, MWCNN, PRIDE-NET, Red-Net, and DnCNN Architectures. In that we found DnCnn is very efficient model to implement , Therefore we can achieve the best accuracy. But to understand the Basic working of DnCnn, we have to work on the Red-Net Model but while working on it we get know that it takes lot of time also less accurate comparatively.

The Advanced Red-Net Architectural Model is interlinked to the working principle of Autoencoders. Autoencoders transform the real time data or information to machine interpretable language and is also used to denoise the images. So, for better understanding we worked on the autoencoder denoising model, to further implement the Red-Net and Dncnn model. After all the analysis of the models, we came to the conclusion that Dncnn is far better in achieving the precise outcome compared to PRIDE-NET,MWCNN.

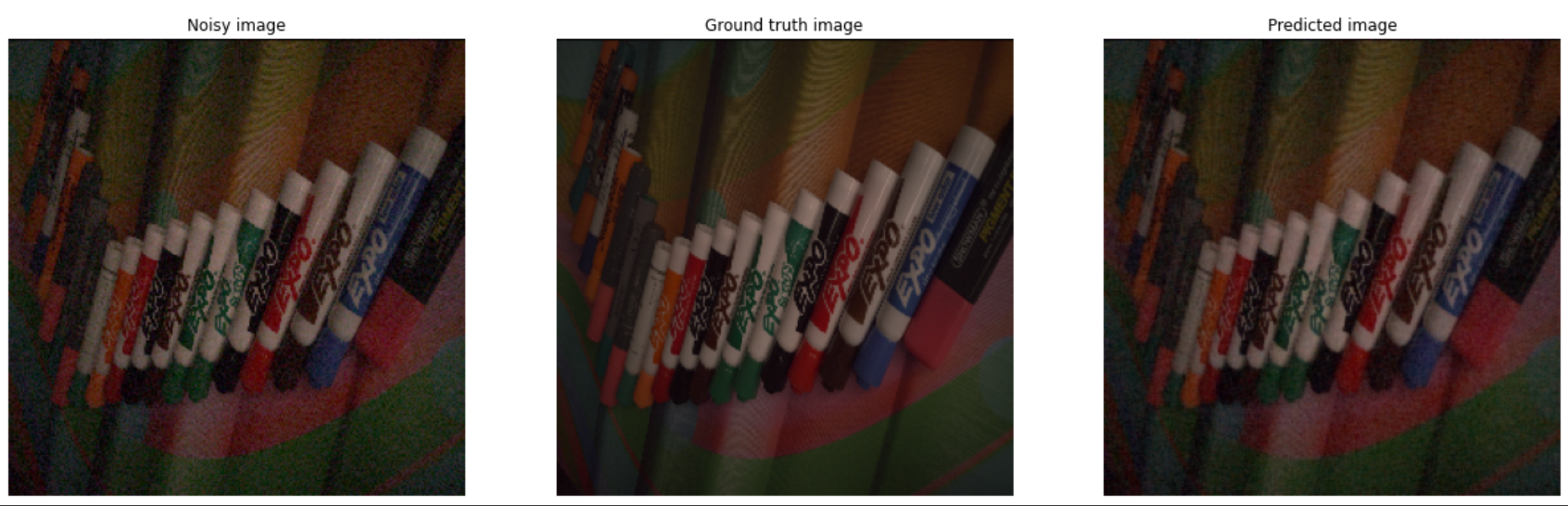
# Implementation:

* Data preparation is the major part in succeeding the task of denoising as we may not be able to understand what is happening internally without the correct dataset.
* We've imported all the necessary libraries like NumPy, pandas, seaborn, sklearn, train\_test\_split and imported the sequential model to fit the model and arrange the data in sequential order to train and test. mainly we use the utils models for all neural networks to do the image processing task with ease.
* We’ve imported the ignore warnings to avoid confusion in understanding the Outcome.
* loaded the train and test data files (format .csv)
* After we assigned a Label to the variable for training, to understand the data we plotted the count plot with various colors.
* Created the arrays and initialized the train and test split with test of 0.2.
* Created the train, test split for the noisy images, using the random samples and clipped the data of Digits. Plotted the train and train noisy images.
* Designed a model to implement the neural network layers, with dense and hidden layers, we used an activation function called relu and SoftMax for extracting the information. SoftMax extracts the data by pooling technique, it labels the pixel as a class, which makes the model to compare the difference between the true image and the noisy image. Our model is a Sequential model with 1,62538 total parameters with 4 deep dense Layers.
* We calculated the loss and Accuracy of the model and we used auto encoder using the sequential neural networks and achieved 97%. As it is a base Model we can’t work with the large datasets with autoencoding Sequential models and expect precise results.
* We have imported OpenCV and patchify for processing the images and creating the various patches of images for training and testing
* The Convolution Neural Networks that serve as the foundation for REDNet: Residual Encoder-Decoder Networks share the same auto-encoder design but lack connections. Each Convolution encoder and Deconvolution decoder will have six layers.
* Multi-level Wavelet CNN With the exception of down- and up-sampling, convolutional neural networks have an architecture identical to that of U-Net. Discrete Wavelet Transform (DTW) and Inverse Wavelet Transform will be used in this case (IWT).
* Pyramid Real Image Denoising Network, or PRISNet, will be the pinnacle of deep learning technology. The models will be divided into three primary modules, the first of which is the Channel Attention Module, which will add attention to each input channel. The Multi-Scale Feature Extraction Module, the primary module of each architecture, is the second module. This calls for image pooling and down sampling.
* We’ve converted the image files to pickle format for importing the pickle library and we added the peak signal to noise ratio, optimizers, Adam and structural similarity to validate the frequency of the train-test images.
* Extracting the image paths for our SIDD Dataset. Created a folder for ground truth and noisy images, then appended the GT and Noisy Images to the folder.
* Have to Read the data frame and label the data accordingly.
* Reading and locating the data to get everything in shape and specific format.
* Use train and test split, then resize the image by adding RGB channel and add that to the list, then converted to arrays.
* printed the noisy and ground truth image, to analyze.
* We have defined the transmission ,illumination, orientation of the images and label them as per the requirement.
* Then We augmented the data and added all the transmission, illumination, and orientation of images.
* Have verified the images of the dataset which is created for image filtering purposes.
* For the RED-NET Model, we have designed a model with few convolutional layers and deconvolutional Layers using a transpose function. So, the image transforms into a new set of arrays and again it will be passed to the deconvolutional layers to bring it back to the original shape of the image.
* For that we plotted our model
* We have steps for each epoch training and validation and after that we call back all the checkpoints and assigned the value loss function with a patience level of 3 and 10 for different conditions
* We complied our model using MSE with an optimizer called ADAM and fitted our model
* To test the image well we ran the epochs to verify the loss and then we save the model
* After that we validate our model called Interference with noisy images and we predicted output and compared with Noisy and Ground Truth Image
* Then by using SSIM(which shows the loss in the quality) and PSNR (Comparison of 2 images) we evaluated the quality of the predicted image quality to quantity measurement.

**Results**:







**Preliminary Results:**

As we know that image denoising process can be done by the people with pretrained models thinking that the existing Rednet Models accuracy is detrimental , but if we train well with the right data set like SIDD with exact ratio of Ground Truth and the Noisy Images can give the similar results of pretrained models.

**Project Management:**

**Work completed :**

We have implemented the base model of image denoising which is Auto Encoder with the Sequential Neural Network Model. We have created a Neural Network model for Red Net and Pride Net that has Convolutional and deconvolutional layers, helpful for training noisy images and compared with the Ground truth Images. Then we converted the image to layers which helped to compare the variations in the channels of the image thoroughly. Then after we will send both the images to the CNN then internally it converts by passing to the convolutional layers and again it deconvolutes the image to re-track the original shape of the image. Then by adding loss functions and optimizers and delta values to the minimum level by defining the patience level. Then finally evaluated the Images taken: Noisy image to the Ground Truth Image to the Predicted Image. While working on RedNet we get know that it takes lot of time also less accurate comparatively to the DNCNN model hence this works faster and gives a great accuracy in reasonable time.

Then by using SSIM(which shows the loss in the quality) and PSNR (Comparison of 2 images) we evaluated the quality of the predicted image quality to quantity measurement.

**Responsibility (Task, Person):**

Data Preprocessing and Auto Encoder using MNIST dataset: Roshan Sah

Data Preprocessing and Red Net Model Implementation :Saisri Teja Pepeti

Red Net Model and the Auto Encoder using MNIST dataset: Yamuna Bollepalli

SIDD dataset preprocessing and Inference: Guduru Charan chand

# Contributions (members/percentage) :

Roshan Shah - 25**%**

Saisri Teja Pepeti - 25%

Yamuna Bollepalli- 25%

Guduru Charan chand - 25%

**Work to be completed :**

In the upcoming increment we are planning to implement the best DNCNN model which will give the best denoising accuracy compared to which we used in this increment which is RedNet and the Auto Encoder.

After thoroughly analyzing all of the models, we came to the conclusion that Dncnn performs significantly better than PRIDE-NET and MWCNN in terms of producing precise results.

# Responsibilities(Task,Person) :

Data Preprocessing and Model Implementation and Report Writing :Roshan Sah

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

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*](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/*](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*](https://arxiv.org/abs/2107.14803)
4. [*https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3791105*](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3791105)
5. [*https://github.com/cszn/DnCNN*](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*](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*](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*](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*](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/*](https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/)
11. [*https://commons.wikimedia.org/wiki/Natural\_Image\_Noise\_Dataset#Tools*](https://commons.wikimedia.org/wiki/Natural_Image_Noise_Dataset#Tools)
12. [*https://www.eecs.yorku.ca/~kamel/sidd/dataset.php*](https://www.eecs.yorku.ca/~kamel/sidd/dataset.php)
13. [*https://adrianbarburesearch.blogspot.com/p/renoir-dataset.html*](https://adrianbarburesearch.blogspot.com/p/renoir-dataset.html)

**Recording Link :** [***https://www.youtube.com/***](https://www.youtube.com/)