Convex Optimization of Image Denoising using CVXPY layers

Authors:

- **P H V PAVAN KUMAR - CB.EN.P2CEN20020**
- **RAHESH R - CB.EN.P2CEN20021**
- **B. Sachin Aditiya - CB.EN.P2CEN20030**

श्रद्धावान् लभते ज्ञानम्

AMRITA VISHWA VIDYAPEETHAM

DEEMED TO BE UNIVERSITY

Introduction

- Digitized images have become an integral part of our lives in day-to-day applications, which is an enthralling part of Computer Vision and Robotics.
- But due to the influence of environmental constraint, transmission issues, and other factors, images are unavoidably corrupted by noise due to acquisition, compression, and transmission.
- This noise level will vary from low to high range, which leads to image distortion.
- In order to restore the originality of the image, the Image denoising technique is being used.
- The image denoising technique helps in reducing noise.
- But the common problem faced by many researchers is to find an effective convex optimization solver for the denoising technique.

Objective

- In this project, our primary objective is to analyze the performance of Convex Optimization CVXPY layers solver in modified VGG-16 Net for Image Classification of Noisy Image data set.

Literature Survey

Title	Source	Observation
LEAST SQUARE BASED APPROACH FOR IMAGE INPAINTING Aiswarya M; Deepika .N; Sowmya; Dr. Neethu Mahan; Dr. Soman K. P.	Institute of Integrative Omics and Applied Biotechnology, Volume 7, p.44-59 (2016)	This paper presented a least square based approach for image inpainting. .
Image Denoising Based on Weighted Regularized Least Square Method M. Srikanth; K. S. Gokul Krishnan; V. Sowmya; K. P. Soman	2017 International Conference on circuits Power and Computing Technologies [ICCPCT]	Image denoising based on weighted regularized least square method was proposed in this paper.

Literature Survey

Title	Source	Observation
Differentiable Convex Optimization Layers Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, Zico Kolter	NeurIPS 2019	A disciplined parametrized programming, a subset of disciplined convex programming was introduced which helps in introducing new solver solution to a existing problem
Convex Optimization Stephen Boyd Department of Electrical Engineering Stanford University	Textbook	Concept of Optimization

Background Requirements

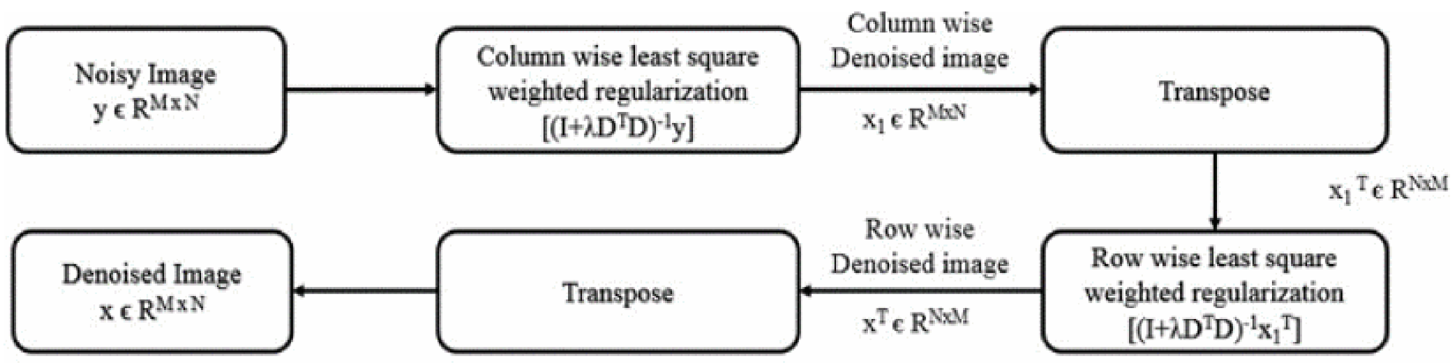
- Least Square Based Image Denoising Concept
- Mnist Dataset
- VGG16 Network
- CVXPY Layers

Least Square Image Denoising

Here we use the least square weighted regularization algorithm for signal denoising. The one dimensional signal denoising using least square approach was proposed by Ivan. W. Selesnick

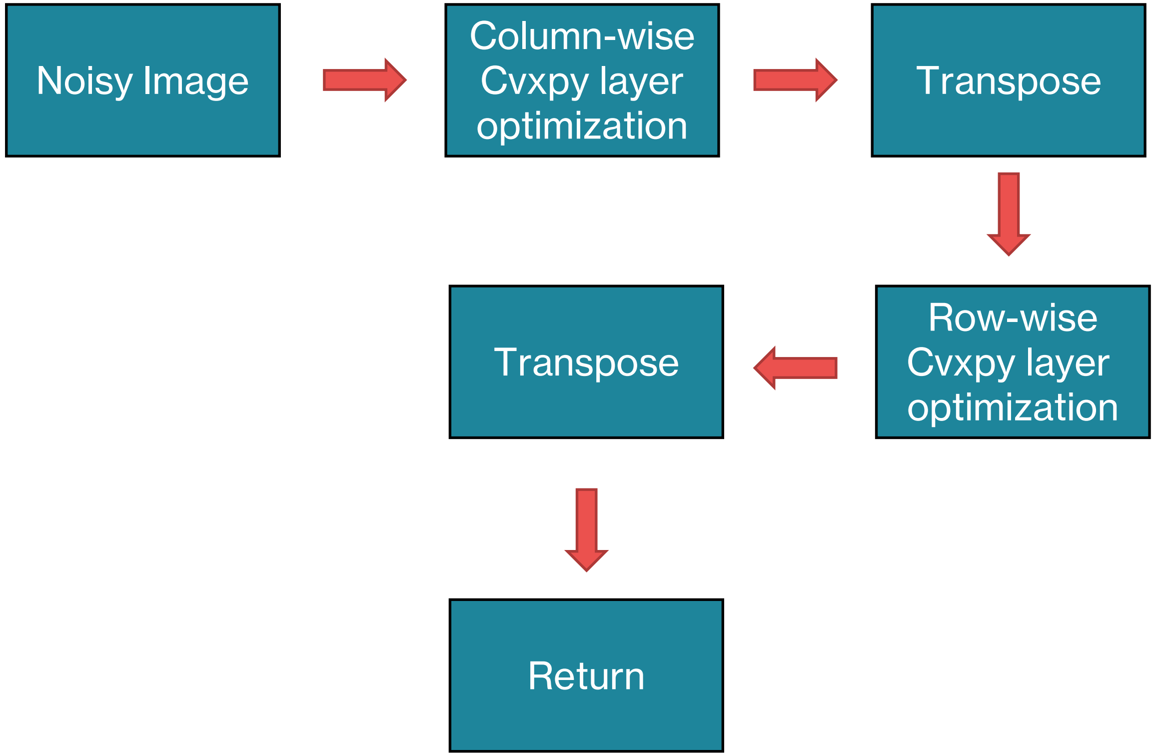
$$min_x \|y - x\|_2^2 + \|\mathbf{D}x\|_2^2$$

Least Square Image Denoising




```python
def call(self, data):
  tf.executing_eagerly()
  denoised = np.zeros(shape=(data.shape[1],data.shape[2]))
  denoised_train = np.ndarray(shape=data.shape)
  img_ix =0
  for img in data:
      for i in range(img.shape[0]):
          sig = img[:,i]
          sig = tf.convert_to_tensor(sig)
          ans, = self.cvxpy_layer(sig)
          denoised[:,i] = ans
      #denoised = denoised
      for i in range(img.shape[1]):
          sig = denoised[i,:]
          sig = tf.convert_to_tensor(sig)
          ans, = self.cvxpy_layer(sig)
          denoised[i,:]= ans
      denoised_train[img_ix,:,:] = denoised
      img_ix+=1
  a,b,c = denoised_train.shape
  return denoised_train.reshape(a,b,c,1)
```

Least Square Image Denoising




```python
def call(self, data):
  tf.executing_eagerly()
  denoised = np.zeros(shape=(data.shape[1],data.shape[2]))
  denoised_train = np.ndarray(shape=data.shape)
  img_ix =0
  for img in data:
      for i in range(img.shape[0]):
          sig = img[:,i]
          sig = tf.convert_to_tensor(sig)
          ans, = self.cvxpy_layer(sig)
          denoised[:,i] = ans
      #denoised = denoised
      for i in range(img.shape[1]):
          sig = denoised[i,:]
          sig = tf.convert_to_tensor(sig)
          ans, = self.cvxpy_layer(sig)
          denoised[i,:]= ans
      denoised_train[img_ix,:,:] = denoised
      img_ix+=1
  a,b,c = denoised_train.shape
  return denoised_train.reshape(a,b,c,1)
```

MNIST Dataset

MNIST dataset contains handwritten images ranging from 0-9.

MNIST dataset has a training set of 60000 images and testing set of 10000 images

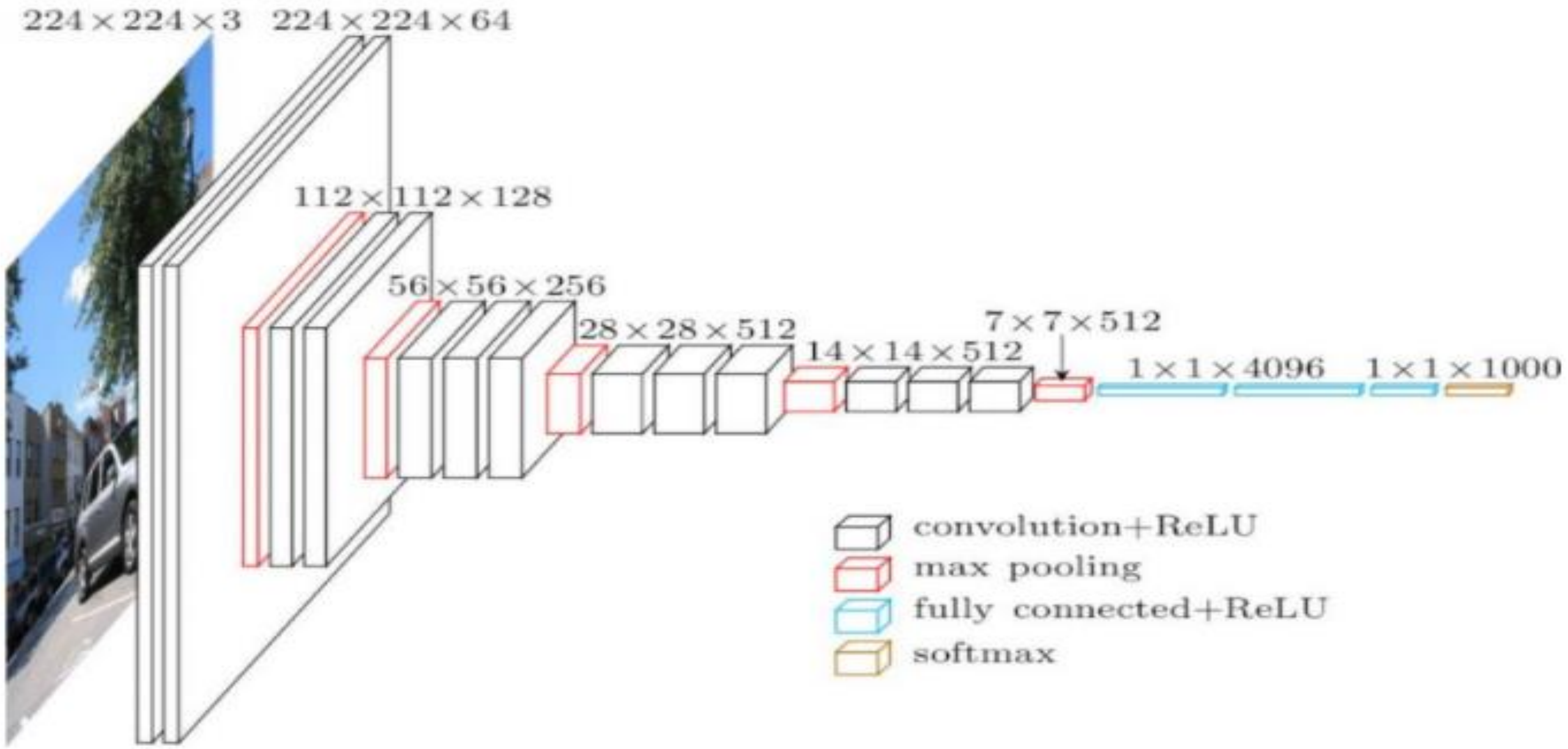
Images in the dataset 28 x 28 pixels.

VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”.

The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

VGG16 Architecture



CVXPY Layers

- CVXPY is a Python-embedded modeling language for convex optimization problems.
- It allows you to express your problem in a natural way that follows the math, rather than in the restrictive standard form required by solvers.
- CVXPY began as a Stanford University research project. It is now developed by many people in different versions, across many institutions and countries.
- cvxpylayers is a Python library for constructing differentiable convex optimization layers in PyTorch, JAX, and TensorFlow using CVXPY.

CVXPY Layers

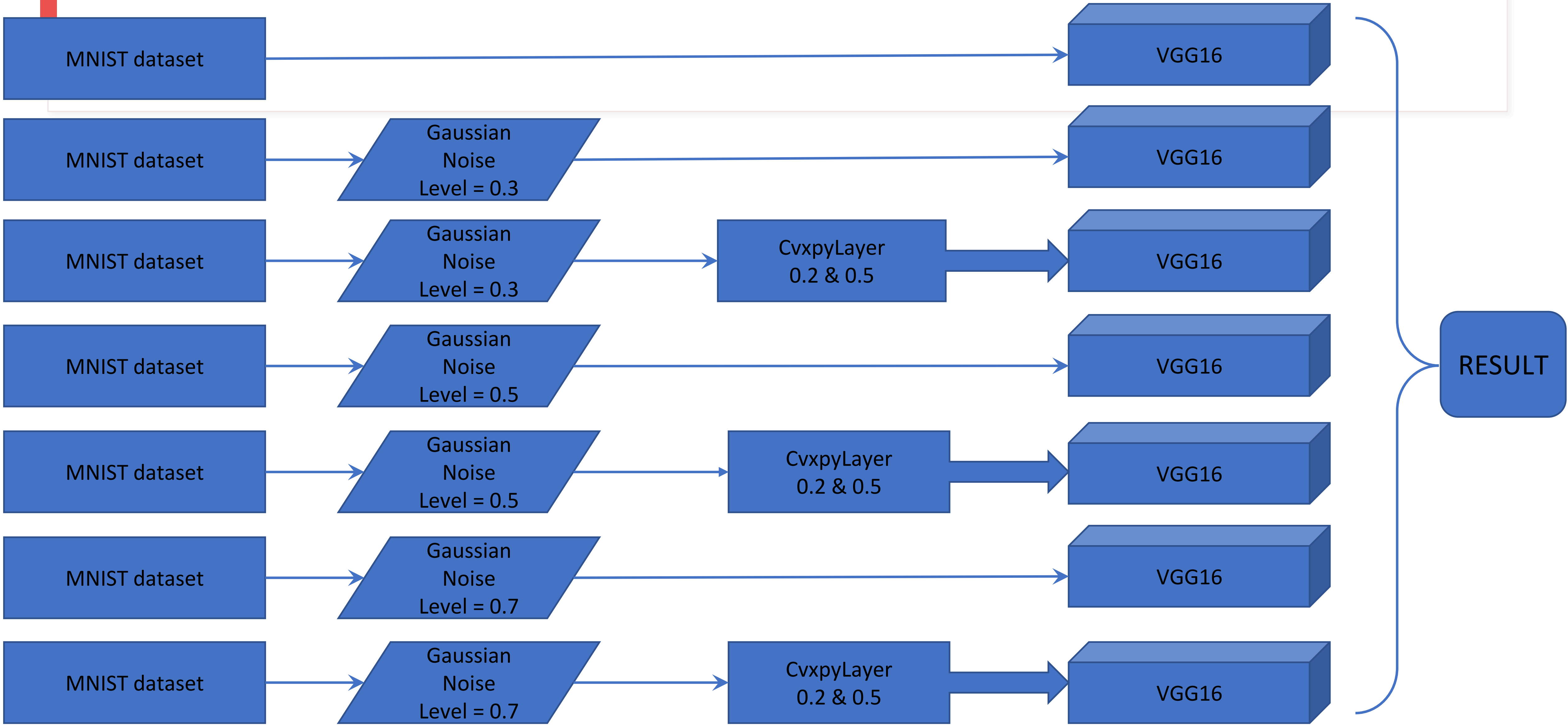
- Expressions:
  - Variables
  - Constants
  - Parameters
- Constraints
- Problem

```
from cvxpylayers.tensorflow import CvxpyLayer
import cvxpy as cp
import tensorflow as tf
from tensorflow.keras import layers
import numpy as np
tf.compat.v1.enable_eager_execution()
tf.config.run_functions_eagerly(True)

class Denoise(layers.Layer):
  def __init__(self, input_shape, lam_val,**kwargs):
    super(Denoise, self).__init__(**kwargs)
    tf.executing_eagerly()
    self.input_dim,_ = input_shape
    self.y_parm = cp.Parameter(self.input_dim)
    self.lambda_parm = cp.Constant(value=lam_val)
    self.x_var = cp.Variable((self.input_dim))
    self.objective = cp.sum_squares(self.y_parm-self.x_var) + self.lambda_parm*cp.sum_squares(cp.diff(self.x_var,2))
    self.constrains = [self.x_var<=self.y_parm,self.x_var>=0]
    self.problem = cp.Problem(cp.Minimize(self.objective),constraints=self.constrains)
    self.cvxpy_layer = CvxpyLayer(problem=self.problem, parameters = [self.y_parm], variables=[self.x_var])
```

$$\boxed{0 \le x \le y}$$

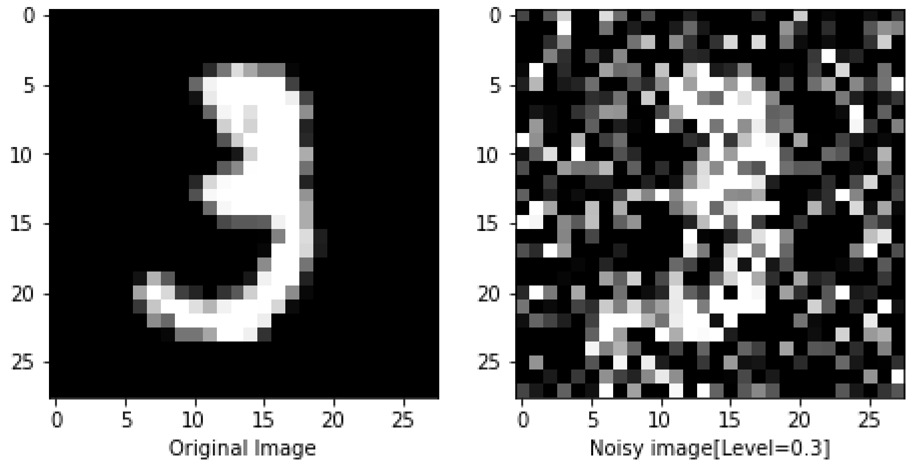
Methodology



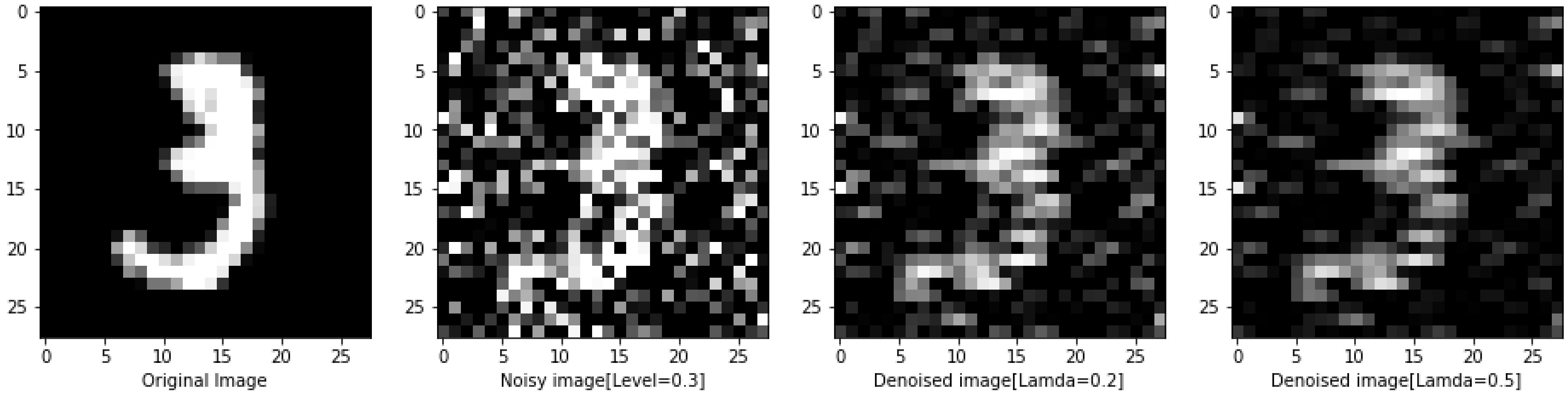
Noise(Gaussian noise)

- Used skimage library for adding Gaussian noise.

```
import skimage
noise_lvl_1 = 0.3
noise_x_train_1 = np.array([skimage.util.random_noise(x, mode='gaussian', seed=42, clip=True, var = noise_lvl_1)  for x in x_train])
noise_x_test_1 = np.array([skimage.util.random_noise(x, mode='gaussian', seed=42, clip=True, var = noise_lvl_1) for x in x_test])
```



RESULTS



0
5
10
15
20
25
Original Image
Noisy image[Level=0.3]
Denoised image[Lamda=0.2]
Denoised image[Lamda=0.5]

RESULTS(cont)

Observations of Train and Test accuracy:

NOCVXPY + VGG16			CVXPY + VGG16		
Noise	train accuracy	test acc	Lamda	train accuracy	test acc
0	0.9747	0.97	-	-	-
0.3	0.9676	0.95	0.1	0.9236	0.93
			0.2	0.954	0.93
			0.3	0.8425	0.75
			0.5	0.9204	0.88
0.5	0.9642	0.91	0.1	0.8902	0.65
			0.2	0.9148	0.74
			0.3	0.7704	0.79
			0.5	0.7858	0.61
0.7	0.9639	0.91	0.1	0.9568	0.9
			0.2	0.885	0.89
			0.3	0.8808	0.8
			0.5	0.7559	0.63

Observations of PSNR values:

NOCVXPY + VGG16		CVXPY + VGG16	
Noise	PSNR	Lamda	PSNR
0.3	8.926987	0.1	11.91903
		0.2	12.63295
		0.3	12.86479
		0.5	12.94257
0.5	7.591076	0.1	10.57404
		0.2	11.81571
		0.3	12.13295
		0.5	12.27896
0.7	6.863497	0.1	9.801258
		0.2	10.63991
		0.3	11.00036
		0.5	11.2757

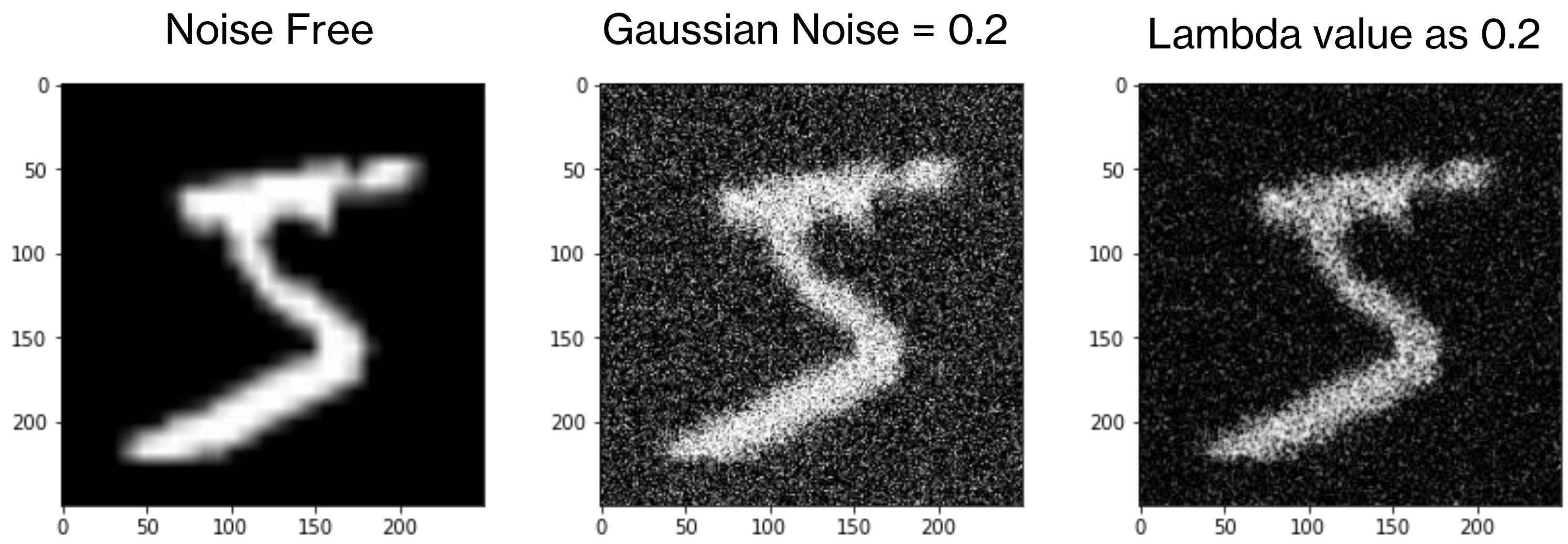
Conclusion:

- This approach reduces the noise in the images through convex optimization.
- The accuracy increases for the denoised data in some cases.
- PSNR improves after denoising
- For better results the noise should be less and the control parameter should be optimum.
- We believe that this approach works well with higher dimension images with less noise inherited with it.

Conclusion:

Limitations:

- The time required for attaining the convex optimisation increases as the input size and constrains increases.



Future Work:

- This methodology can be applied for higher dimension images and other datasets too.
- This can be applied to other architectures.

References

1. Image Denoising Based on Weighted Regularized Least Square Method M. Srikanth, K.S. Gokul Krishnan, V. Sowmya, and K.P. Soman

2. Least Squares with Examples in Signal Processing1 Ivan Selesnick

3. LEAST SQUARE BASED APPROACH FOR IMAGE INPAINTING Aiswarya M; Deepika .N; Sowmya; Dr. Neethu Mahan; Dr. Soman K. P.

4. https://github.com/cvxpy/cvxpy

5. https://github.com/cvxgrp/cvxpylayers