Mini Project Report

Entitled

Image Denoising using Machine Learning

Submitted to the Department of Electronics Engineering in Partial Fulfilment for the Requirements for the Degree of

Bachelor of Technology
(Electronics and Communication)

: Presented & Submitted By :

Avuthu Akash Reddy	U21EC038
Varadi Chandra Sekhar	U21EC099
Gangasani DevendraNatha Reddy	U21EC108

B. TECH. VI (EC), 6th **Semester**

: Guided By :

Dr. Kishor Upla
Associate Professor, SVNIT

(Year: 2023-24)

DEPARTMENT OF ELECTRONICS ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY
Surat-395007, Gujarat, INDIA.

Sardar Vallabhbhai National Institute Of Technology

Surat - 395 007, Gujarat, India

DEPARTMENT OF ELECTRONICS ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project Report entitled "**Image Denoising using Machine Learning**" is presented & submitted by Avuthu Akash Reddy, Varadi Chandra Sekhar, and Gangasani DevendraNatha Reddy, bearing Roll No. U21EC038, U21EC099, and U21EC108, of B.Tech. VI, 6^{th} Semester in the partial fulfillment of the requirement for the award of B.Tech. Degree in Electronics & Communication Engineering for academic year 2023-24. They have successfully and satisfactorily completed their **Mini-Project** in all respects. We, certify that the work is comprehensive, complete and fit for evaluation.

Dr. Kishor Upla
Associate Professor & Project Guide

Abstract

Images are susceptible to various kinds of noises,which corrupt the pictorial information stored in the images. Image de-noising has become an integral part of the image processing workflow. It is used to attenuate the noises and accentuate the specific image information stored within. This report describes the application of machine learning techniques for image denoising specifically targeting Gaussian noise reduction.The approach leverages two complementary methods: dictionary learning and ridge regression.

This report describes the potential of ridge regression, a machine-learning technique, for image denoising. The focus lies on removing Gaussian noise, a common challenge in image processing. The method leverages ridge regression's ability to learn a mapping between noisy image patches and their corresponding clean counterparts. By incorporating a regularization term, ridge regression addresses the issue of overfitting, a potential drawback in denoising tasks.The effectiveness of this approach is evaluted on a set of test images.These de-noisers are compared using PSNR quality assessment metric.

Table of Contents

List of Figures

Chapter 1
Introduction

Digital images are often corrupted by noise during transmission. This noise degrades image quality, obscuring details and hindering further processing tasks. Traditional methods often struggle with complex noise structures or scenarios where noise and image details are intertwined. Deep learning or Machine Learning models, with their exceptional ability to learn intricate relationships from vast amounts of data, can excel at identifying and removing even the most complex noise patterns.

1.1 Types of Noises

The noise classification is done based on its probability distribution function, correlation, nature, and its source. The different types of noise based on pdf are Gaussian, Rayleigh, Uniform, Impulse, Possion, etc. According to the correlation, noise is classified into white and color noise. The white noise has uniform power spectral density and zero autocorrelation, unlike color noise. If an image is corrupted with white noise, it implies that all pixels are uncorrelated with each other. It is additive or multiplicative according to nature. Commonly used noise types are mentioned below [2].

- Gaussian Noise
- Impulse Noise
- Poisson Noise or Photon Noise
- Mixed Noise

1.1.1 Gaussian Noise

It is statistical and additive in nature which follows normal distribution with zero mean and σ standard deviation and affects all the pixels in the image. The cause of its occurrence is sensor temperature fluctuation and environmental illumination variations. It is commonly found in magnetic resonance imaging and confocal laser scanning microscopy imaging. The probability distribution function of Gaussian noise is given by the following equation.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1.1}$$

where x is image pixel value, μ is mean and σ is the standard deviation.

1.1.2 Impulse Noise

It is an additive noise which occurs due to faulty sensors and transmission error. It affects only certain pixels in the entire image, unlike Gaussian noise. It is divided into two parts, i.e., salt and pepper impulse noise (SPIN) and random valued impulse noise (RVIN). In salt and pepper noise corruption, some image pixels take either maximum or minimum value of image dynamic range. Whereas RVIN corruption changes some image pixels with a random value, its detection is more difficult than salt and pepper noise detection. The salt and pepper impulse noise is given by

$$P(x) = \begin{cases} P_a, & \text{for } x = a \\ P_b, & \text{for } x = b \\ 0, & \text{otherwise} \end{cases} \tag{1.2}$$

where a and b are minimum and maximum pixel values of an image dynamic range. P_a and P_b are probabilities are equal for salt and pepper noise.

1.2 Machine Learning Based Image De-Noising

Ridge Regression

Ridge regression is a statistical regularization technique to reduce errors caused by overfitting on training data. It is also known as L2 regularization. It is one of several types of regularization for linear regression models. Ridge regression specifically corrects for multicollinearity in regression analysis.

Dictionary learning

Dictionary learning is a branch of signal processing and machine learning that aims at finding a frame (called dictionary) in which some training data admits a sparse representation. The sparser the representation, the better the dictionary.

1.3 Scope

The scope of this project includes implementing ridge regression and dictionary learning for image denoising and evaluating their effectivenes with quantitative metrics like PSNR (Peak Signal-to-Noise Ratio) offering valuable insights into the strengths and weaknesses of these techniques,

Chapter 2
Algorithm

2.1 Patch Extraction

Images often contain recurring patterns within local regions. By dividing the image into smaller patches, techniques like dictionary learning can exploit this redundancy. Similar patches usually have similar noise patterns, making it easier to identify and remove the noise. It also breaks down the complex image denoising problem into smaller, more manageable pieces. This allows the algorithms to leverage local information, reduce model complexity, and ultimately achieve better denoising results.

To apply dictionary learning or ridge regression for image denoising , the input image is thus divided into smaller patches and are used as training samples. The size of the patches and the patch extraction strategy are directly related to the denoising performance. The smaller the patches the better the performance but more is the time it takes to handle those patches. Hence, it is important to choose the right size of the patches.

2.2 Model Training

2.2.1 Dictionary Learning

Dictionary learning is an unsupervised machine learning approach that uses data to learn a set of basic functions or atoms. It tries to represent the data as a dictionary formed by a sparse linear combination of these atoms. By iteratively updating the dictionary and sparse codes, the algorithm finds the most compact and informative representation of the data [3].

- The first step is to learn a dictionary from a set of training data that is similar to the images you want to denoise.
- The dictionary consists of a set of basis functions, or atoms, that can represent different image patches.
- Dictionary learning algorithms, such as K-SVD or the method of optimal directions (MOD), are used to iteratively update the dictionary to best fit the training data.

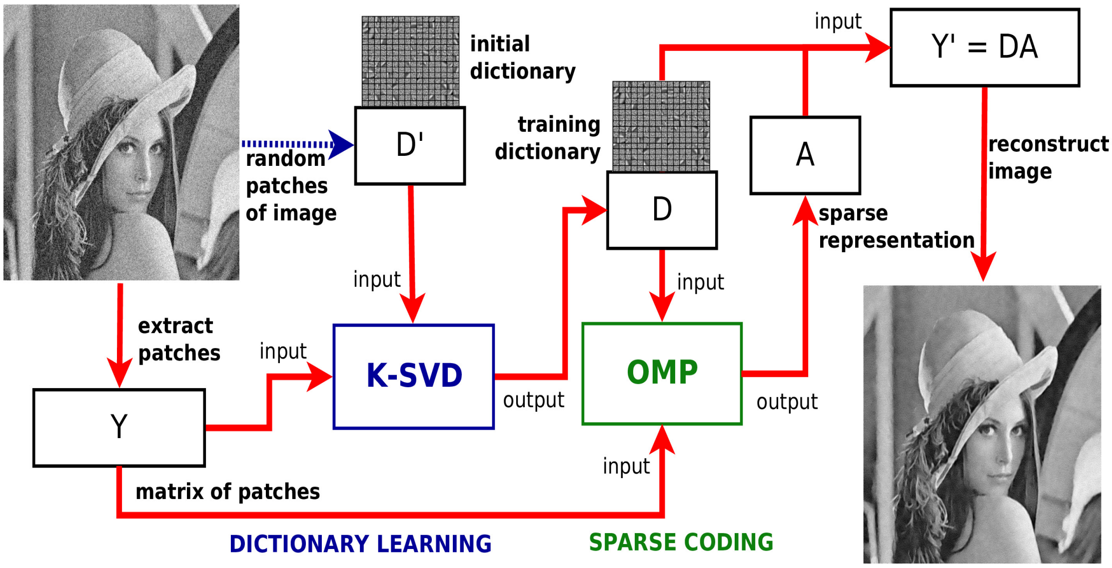

Figure 2.1: Dictionary Learning Approach for Image Denoising. [1]

Sparse Coding

Sparse coding is a technique that represents each data point (e.g., an image patch) as a linear combination of a small number of atoms from the learned dictionary. It follows the following steps to find the sparsest representation.In the context of image denoising, dictionary learning follows the following steps:

- Once the dictionary is learned, each image patch is sparsely represented as a linear combination of atoms from the dictionary.
- Sparse coding aims to find a sparse representation that uses a minimal number of atoms to represent the image patch or to reduce the reconstruction error between the original data and its representation using dictionary atoms.
- Techniques like Orthogonal Matching Pursuit (OMP) or Lasso can be employed to find the sparse codes.

2.2.2 Ridge Regression

Traditional linear regression aims to fit a line (or hyperplane in higher dimensions) through a set of data points. However, when dealing with noisy data like images, fitting a model that exactly passes through all data points can be problematic. This can lead to overfitting, where the model captures noise instead of the underlying image structure.

Ridge regression addresses this issue by introducing a regularization term to the cost function used for model training. This regularization term penalizes the complexity of the model by adding a penalty proportional to the squared sum of the coefficients. In simpler terms, it discourages the model from having overly large coefficients, promoting a smoother and more generalized solution.

$$SSE_2 = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.1}$$

2.3 Reconstruction of Image

The noisy image patches are rebuilt using the learned dictionary and the sparse representations in case of Dictionary learning based image denoising. The reconstruction process involves merging atoms from the dictionary based on corresponding sparse codes. Sparse codes are used to regulate how much each atom contributes to the reconstruction, highlighting significant visual elements while suppressing noise components.

For ridge regression the pixels intensity values are smoothened according to the coefficients it estimated during the training process. The regularization term takes care of cases when the image is oversmoothed at the edges causing a blur unlike Linear Regression. As it is much simpler than dictionary learning it is expected to be less performant than dictionary learning but it is much more faster than dictionary learning and gives reasonably good results.

Chapter 3
Implementation

3.1 Ridge Regression

```
import numpy as np
import sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
from skimage.util import random_noise
from sklearn.feature_extraction.image import
extract_patches_2d,reconstruct_from_patches_2d
from sklearn.metrics import mean_squared_error
from skimage.filters import unsharp_mask
from skimage.metrics import peak_signal_noise_ratio
import cv2

image = cv2.imread('test.jpg',cv2.IMREAD_GRAYSCALE)/255.0

image = (
image[::4, ::4]
+ image[1::4, ::4]
+ image[::4, 1::4]
+ image[1::4, 1::4]
)
image /= 4.0
height, width = image.shape

noisy_image = random_noise(image, mode='gaussian', clip=True,
    var=0.003)
noisy_image = random_noise(noisy_image, mode='s&p', amount
    =0.001)

noisy_patches = extract_patches_2d(noisy_image,
                                   patch_size=(7, 7),
                                   random_state=42)
original_patches = extract_patches_2d(image,
                                      patch_size=(7, 7),
                                      random_state=42)

X_train,X_test,y_train,y_test = train_test_split(
                 noisy_patches.reshape(-1, 7*7),
                 original_patches.reshape(-1, 7*7),
```

```
                  test_size=0.2)

model = Ridge(alpha=1)
model.fit(X_train, y_train)

mean_squared_error(y_test, model.predict(X_test))

denoised_patches = model.predict(
                       noisy_patches.reshape(-1, 7*7))
                       .reshape(-1, 7, 7)
denoised_image = reconstruct_from_patches_2d(denoised_patches,
                                             image_size=
                                             (height, width))
sharpened_image=unsharp_mask(denoised_image,radius=1,amount=1)
```

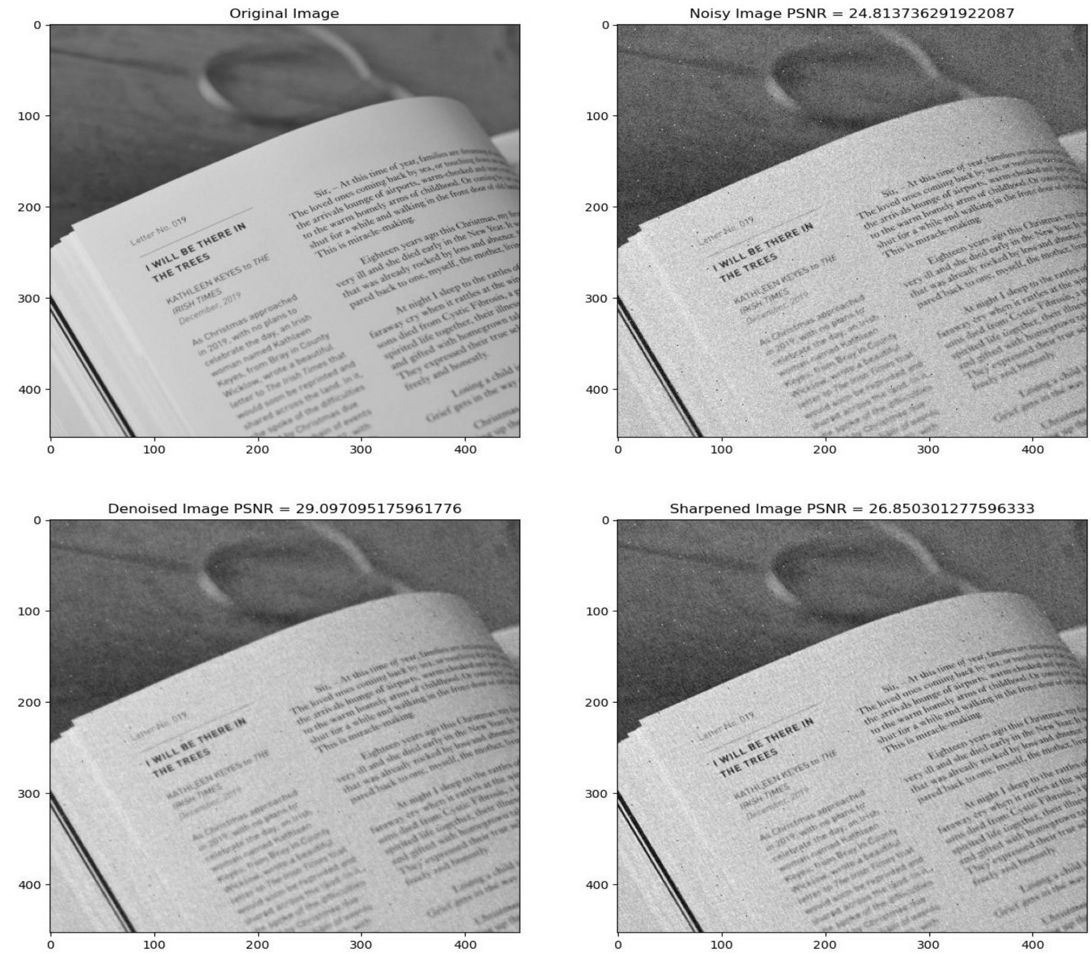

Figure 3.1: Denoising of image using Ridge Regression

3.2 Dictionary Learning

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
from sklearn.feature_extraction.image import extract_patches_2d
    , reconstruct_from_patches_2d
from sklearn.decomposition import MiniBatchDictionaryLearning
from skimage.util import random_noise

image = cv2.imread("test.jpg", cv2.IMREAD_GRAYSCALE)

image = image / 255.0

image = (
    image[::4, ::4]
    + image[1::4, ::4]
    + image[::4, 1::4]
    + image[1::4, 1::4]
)
image /= 4.0
height, width = image.shape

distorted = random_noise(image, mode='gaussian', clip=True, var
    =0.003)
distorted = random_noise(distorted, mode='s&p', amount=0.001)

patch_size = (7, 7)
data = extract_patches_2d(distorted, patch_size)
data = data.reshape(data.shape[0], -1)
data -= np.mean(data, axis=0)
data /= np.std(data, axis=0)

dico = MiniBatchDictionaryLearning(
    n_components=50,
    batch_size=200,
    alpha=1.0,
    max_iter=10,
)

V = dico.fit(data).components_
data = extract_patches_2d(distorted, patch_size)
data = ata.reshape(data.shape[0], -1)
intercept = np.mean(data, axis=0)
```

```
data -= intercept

reconstructions = {}

for i in range(2):
    reconstructions[i] = image.copy()
    dico.set_params(transform_algorithm="omp",
        transform_n_nonzero_coefs=i+1)
    code = dico.transform(data)
    patches = np.dot(code, V)
    patches += intercept
    patches = patches.reshape(len(data), *patch_size)
    reconstructions[i] = reconstruct_from_patches_2d(patches, (
        height, width))

from skimage.metrics import peak_signal_noise_ratio
peak_signal_noise_ratio(image, reconstructions[0])
peak_signal_noise_ratio(image, reconstructions[1])
```

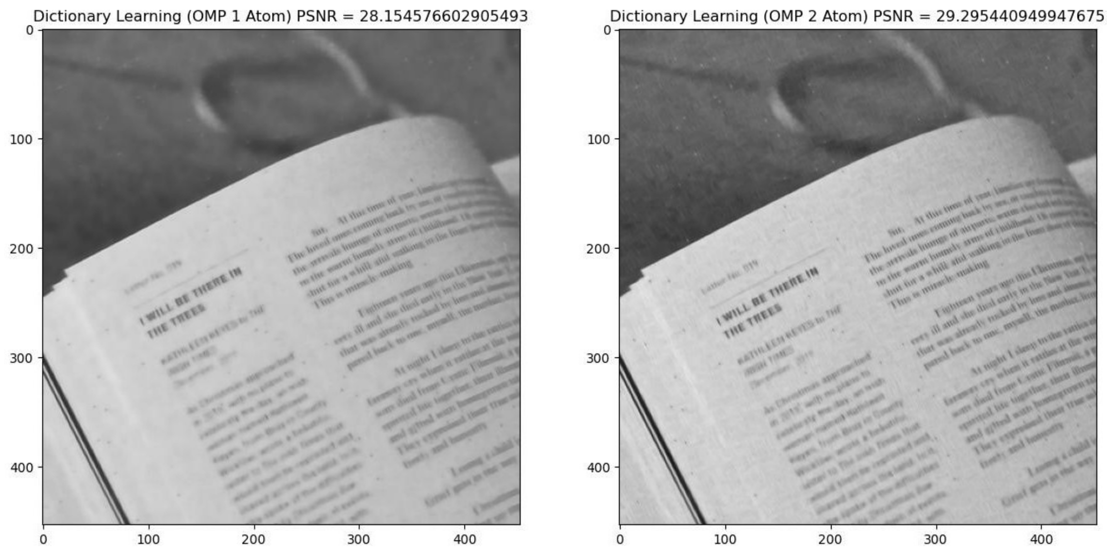

Figure 3.2: Denoising of image using Dictionary Learning OMP Algorithm with 1 and 2 Atoms

Conclusion

Machine learning algorithms work by manipulating the pixel values within each patch to remove noise. Replacing noisy pixel values with cleaner values estimated from the learned patterns and also using sparse representations might exclude noisy elements when reconstructing the patch.

This report describes the effectiveness of machine learning for image denoising, specifically comparing ridge regression and dictionary learning with sparse coding using patch extraction. Both methods were evaluated based on their ability to remove noise from a test image and the resulting Peak signal-to-noise ratio (PSNR).

Ridge regression provided a baseline for denoising, achieving a PSNR of 29.09 . Dictionary learning with sparse coding also achieved a PSNR of with OMP 1 atom 28.15 and with OMP 2 atom 29.29.

References

[1] L. J. Fuentes Perez, L. A. Romero Calla, and A. A. Montenegro, "A dictionary learning approach in gpu for image denoising," Poster presented at NVIDIA GPU Technology Conference, 2016. [Online]. Available: https://on-demand-gtc.gputechconf.com/gtcnew/sessionview.php?sessionName=p6294-a+dictionary+learning+approach+in+gpu+for+image+denoising

[2] R. S. Thakur, S. Chatterjee, R. N. Yadav, and L. Gupta, "Image de-noising with machine learning: A review," *IEEE Access*, vol. 9, pp. 93 338–93 363, 2021.

[3] (2023) Image-denoising. Accessed on 29th Oct 2023. [Online]. Available: https://www.geeksforgeeks.org/image-denoising-using-dictionary-learning-in-scikit-learn/