ADV. APP. MATH CONC. MACH. LEARN. CRN-15175-202201

Final Term Project

Submitted in partial fulfillment of the requirements of the program

Applied A.I. Solutions Development

by

1. Varun Khanna 101415041
2. Aman Gupta 101420524
3. Alex Vecchiettini 101439446
4. Mehul Sharma 101414996
5. Manasi Rane 101414495
6. Belle Pensamiento 101398784
7. Kirti Takrani 101423492

Supervisor:

Prof. Moe Fadaee

DEPARTMENT OF COMPUTER TECHNOLOGY

GEORGE BROWN COLLEGE- CASA LOMA CAMPUS

(2022-23)

Table of Content

LIST OF FIGURES AND TABLES

0. Task and Responsibilities

Student Name	Tasks
Varun Khanna	Built SVM, KNN and DT models and benchmarking in part-2
Aman Gupta	Coded initial phase for part-1, Built LR model in part-2 Report for part-2 Presentation of LR model
Alex Vecchiettini	Part 1 - Design of GUI Part 1 - Development of GUI and restructuring of initial code Part 1 - Presentation of problem statement and results Part 2 - Dataset selection
Mehul Sharma	Built RF model in part-2
Manasi Rane	Frequency Domain, FFT, Edge detection, HPF in part-1, Report for part 1 Design of GUI for part 1
Belle Pensamiento	Coded pre-processing phase for part-2
Kirti Takrani	Coded LPF, Blurring in part-1

Part 1

1.1. INTRODUCTION

1.1.1. Problem Statement

Create software for image enhancement using Fourier transformation to obtain the image's frequency domain, in particular, implement the following:

- Sharpening
- Blurring
- Edge detection
- Noise suppression

1.1.2. Dataset

The chosen dataset encompasses a variety of colored images in landscape & portrait mode. We can check the results for image transformations for following types of images:

1. HD coloured Image
2. RGB Colored image
3. Black and white image
4. Noisy image

1.2. METHODOLOGY

1.2.1. Workflow

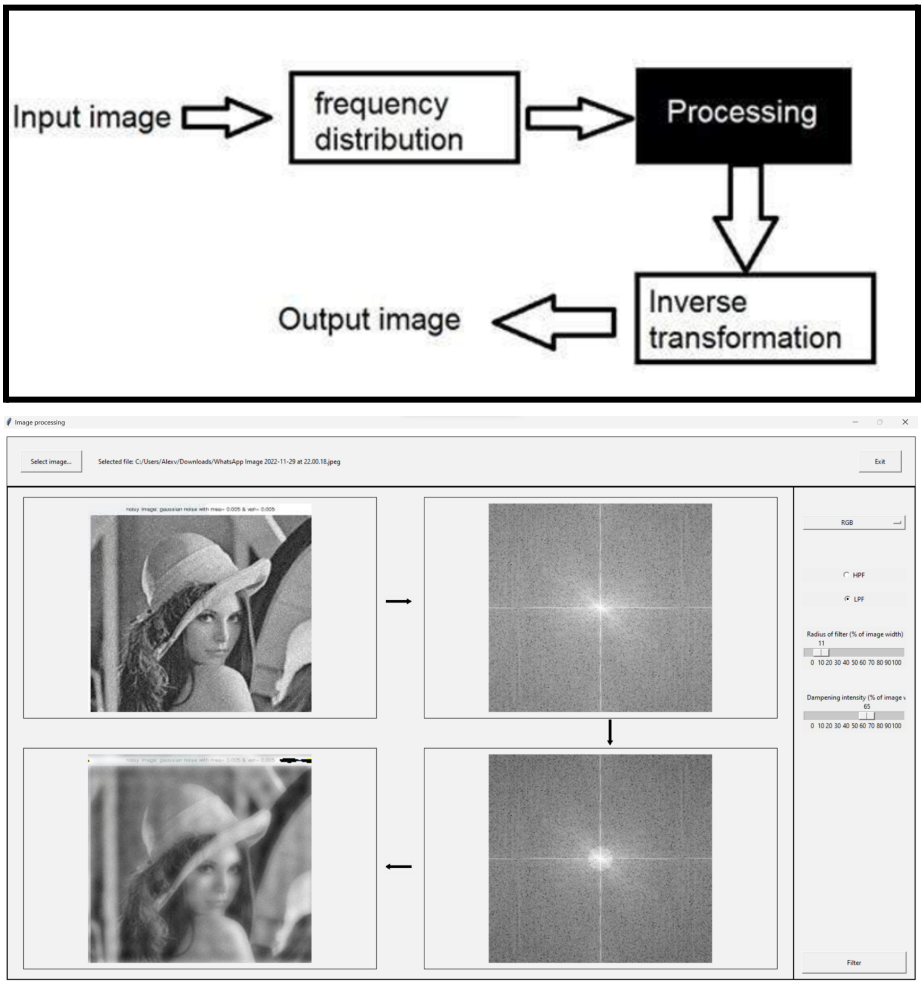

Fig no: 01 Frequency Domain steps

We first transform the image to its frequency distribution. Then we apply a circular mask to filter either high or low frequency, depending on the selected filtering technique. The result of this process is the modified frequency domain of the original image. After performing inverse transformation, it is converted into an image which is then viewed in spatial domain.

Frequency Domain:

It is the rate at which the pixel values are changing in the spatial domain.

A signal can be converted from spatial domain into frequency domain using mathematical operators called transforms.

It can be noted that:

- High frequency components in an image correspond to drastic variation of color, edges are an example of that. A high pass filter, consequently, is the basis for most sharpening methods.
- Low frequency components in an image correspond to smooth regions. A low pass filter is the basis for most smoothing and blurring methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels (see Smoothing an Image for more information)

Fast Fourier transform (FFT):

The Fast Fourier Transform is a particularly efficient algorithm to evaluate the Discrete Fourier Transformation of a function.

The input image is in the spatial domain and the output of the transformation represents the Fourier Transform of the image, or the or frequency domain. The result of the FFT has following parts:

- MAGNITUDE indicates "how much" of a certain frequency component is present
- PHASE indicates "where" the frequency component is in the image.

FT treats image as a part of a periodically replicated array of identical images extending horizontally and vertically to infinity. The FFT method preserves all original data. It fully transforms images into the frequency domain, unlike time-frequency or wavelet transforms.

As in most implementations, the resulting image of the FT is shifted such that the image *F(0,0)* is displayed in the center of the image. The further away an image point is from the center, the higher is its corresponding frequency.

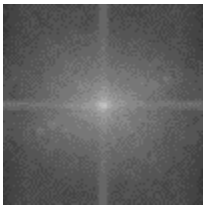

Fig no: 02 Sample FFT

The result shows that the image contains components of all frequencies, but that their magnitude gets smaller for higher frequencies. Therefore, low frequencies contain more image information than the higher ones.

1.2.2. Image Processing Techniques

Low pass Filters (LPF) and High Pass Filters (HPF):

LPF and HPF can be applied to the frequency domain to obtain blurring, sharpening, edge detection, and noise filtering. Different parameters yield different results. In general:

- Blurring is obtained with a LPF that dampens the high frequencies.
- Sharpening can be obtained by applying an HPF to reduce the values of the low frequencies
- Edge detection is achieved by increasing the effect of the HPF and bringing the low frequencies to values close to 0
- Noise filtering, similar to blurring, is achieved through an LPF with a stronger effect on the high frequencies.

1.2.3. Models for processing

i. Sharpening : The simplest way to sharpen an image is to high pass filter it (without the normalization stretch) and then blend it with the original image.

ii. Blurring :

The effect of blurring is to attenuate the high spatial frequencies. The mathematical operation of convolution corresponds to multiplying the Fourier transform of the original image with that of the convolution kernel; this product is the transform of the blurred image. So convolution (in ordinary space) corresponds to multiplying the various frequency components of the image by a filter function (in frequency space). So the filter function of the blurring is the ratio of the Fourier transforms of the output and input images, as a function of spatial frequency

iii. Edge Detection : Edges in an image are usually made of High frequencies. So we need to take a FFT (Fast Fourier Transform) of an image, then apply a High Frequency Pass Filter to this FFT transformed image. This filter would in turn block all low frequencies and only allow high frequencies to go through. Then we take an inverse FFT on this filter applied image, you should see some distinct edge features in the original image.

iv. Noise suppression : To remove noise/ suppress noise we need to manually mask (or notch) out the dots or lines in the magnitude image. We do this by transforming to the frequency domain, create a grayscale version of the spectrum, mask the dots or lines, threshold it, multiply the binary mask image with the magnitude image and then transform back to the spatial domain.

1.2.4. Challenges and Comprehension while drafting solution

i. cv2 interprets the image as BGR rather than RGB. Pyplot displays results in inverted colors.

ii. The imaginary portion of the FFT result and the magnitude in both FFT and inverse FFT were taken into consideration, and

Solution for FFT:

flags=cv2.DFT_COMPLEX_OUTPUT

Solution for IFFT:

flags=cv2.DFT_SCALE | cv2.DFT_REAL_OUTPUT

iii. To scale the frequency back to the spatial domain, the DFT SCALE option, which is exclusively in the IFFT, is required. Accordingly, concluding it means evaluating the magnitude of the vectors described by the real and imaginary parts.

iv. Basic LPF and HPF don't operate well on colored images.

v. It is necessary to resize images and adjust them in accordance with space available on the scree while creating a GUI interface.

vi. We decided to force the main window to maximize to avoid dealing with the rescaling of all the widgets on screen every time the window is resized.

vi. Changed to the 2x2 disposition since it drastically restricts the width of the photos for 4 images in a row, requiring the images to be "super skinny."

vii. Filtering colored images created a lot of distorsion

viii. Testing the GUI on different screen and OS it appeared evident that designing flexible solutions is not easy.

1.3. RESULTS

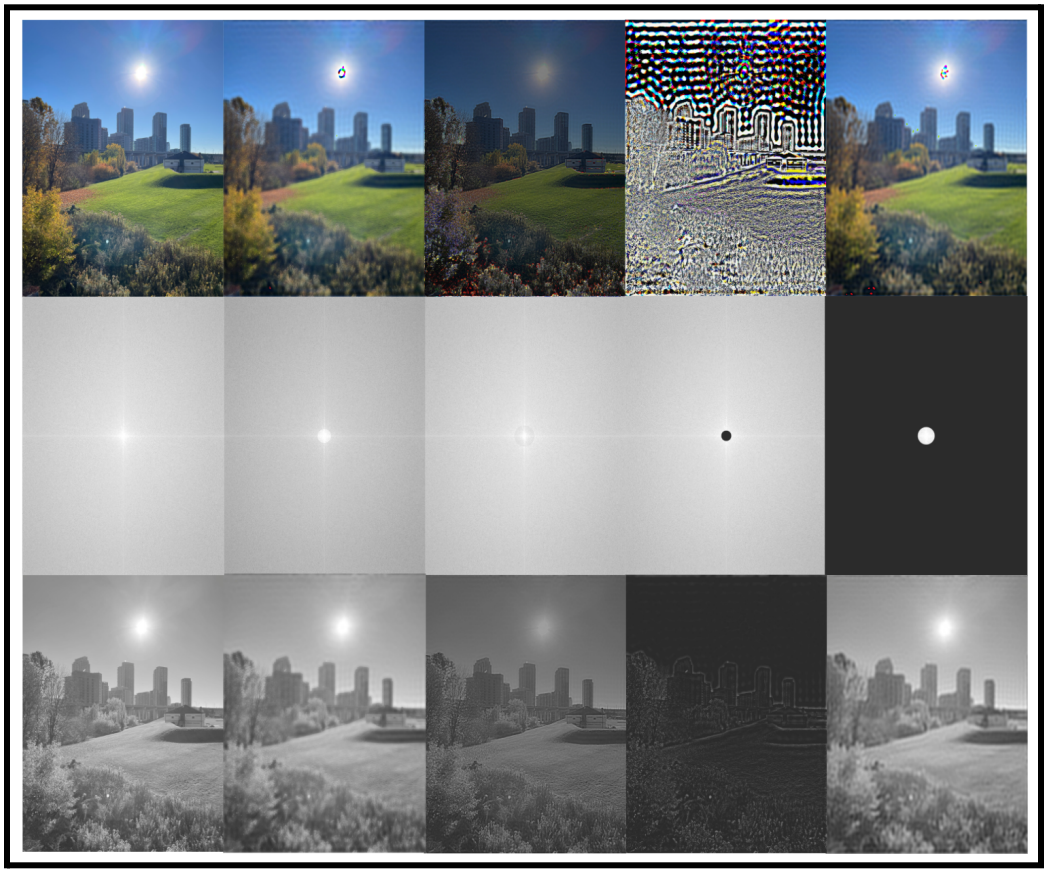

Fig no 03 Part 1 results

Basic LPF and HPF struggle to give great results on colored images whereas they perform fairly well (once fine tuned) on grayscale images.

The first coloured image can be resolved partially by fine-tuning the filter to the image, to obtain marginally better results there are better algorithms in the space domain to achieve the required image enhancements, such as:

- Sobel filter for edge detection
- Gaussian, Mean, Median, and Bilateral filters can be used to deal with different kinds of noise.

Part 2:

2.1. INTRODUCTION

2.1.1. Problem Statement: Landscape Based Image classification

In addition to bringing in tourists and investment, high-quality landscapes can boost local economies. Farming, forestry, and horticulture are other few of the key productive activities supported by rural landscapes. Various landscapes have different meanings to different people.

In the field of remote sensing, land cover classification has grown to be an intriguing research topic. What remote sensing science provides and what domain specialists anticipate from remote sensing data are sometimes incompatible to understand. The field of classifying land cover has seen considerable progress with a variety of applications of machine learning techniques. It is suggested by a number of premises that solving ecological problems in the landscape would be possible using AI-based methods.

We have worked on Landscape classifications based on images.

2.1.2. Data:

https://www.kaggle.com/datasets/puneet6060/intel-image-classification?resource=download

The selected dataset is a collection of global natural images. About 25k images, each measuring 150x150, are included in this dataset and are divided into 6 categories. The categories are buildings, forests, glaciers, sea, mountain, street. We evaluated 2 classes, 'Forest' and 'Buildings' . Building denoted as '0' and Forests as '1'.

2.2. METHODOLOGY

2.2.1. Workflow

The main goal of Data Preprocessing is to reduce dimensionality, problem complexity and computation time. Data preprocessing starts as soon as data is being handled, downloaded, or moved around.

Data Pipeline:

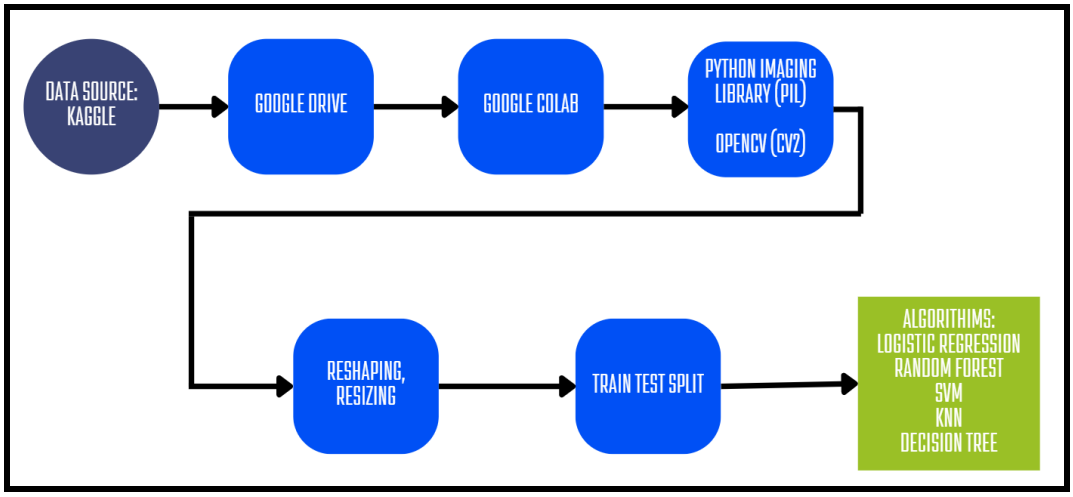

Fig no: 04 Data Pipeline

After downloading our data off of Kaggle, we uploaded it to a google drive in order to be able to connect to it through Google Colab. This facilitated better teamwork as we could all access the data without having to download, but also being able to download it if some preferred to work on their computer. Once we had a working environment, we used Python Imaging Library (PIL) and OpenCV to interpret the data. We reshaped and resized until we got matrices that worked well with our algorithms.

Data Preprocessing:

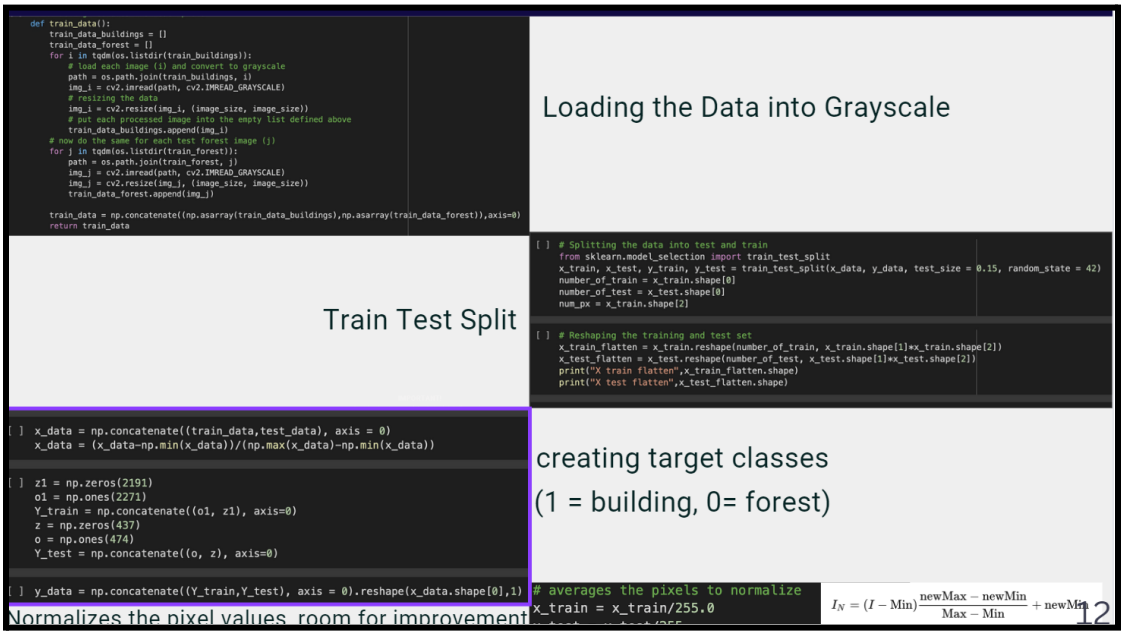

Fig no: 05 Data Preprocessing

As seen in Fig no: 03, we loaded two classes of the available six, making them all grayscale, and then train test split with a test size of 0.15. We kept resizing throughout the process, and also assigned the chosen classes Buildings = 1, and Forests = 0, making sure to match them to the correctly scaled images. FInally, in order to make sure the light and dark regions weren't skewing the results in either direction, we normalized the pixel values by dividing by 255. Histogram equalization must be added to this data pipeline to better our normalization technique.

2.2.2. Models for processing

i. Logistic Regression:

Logistic Regression is one of the supervised machine learning algorithms that can work on both binary and multiclass classification very well. It operates basically through a sigmoidal function for values ranging between 0 and 1. In the context of image processing, this could mean identifying whether a given image belongs to a particular class (y=1) or not (y=0), e.g. "cat" or "not cat".

Formulas used in logistic regression:

$$z = w^T x + b$$

Here z is the output variable, and x is the input variable. w and b will be initialized as zeros to start with and they will be modified by the number of iterations while training the model.

Hyperparameters which gave the maximum accuracy:
LogisticRegression(C=10,penalty="l2",random_state = 42)

- The accuracy we got from LR Model on the test dataset was **75%**
- Confusion Matrix:

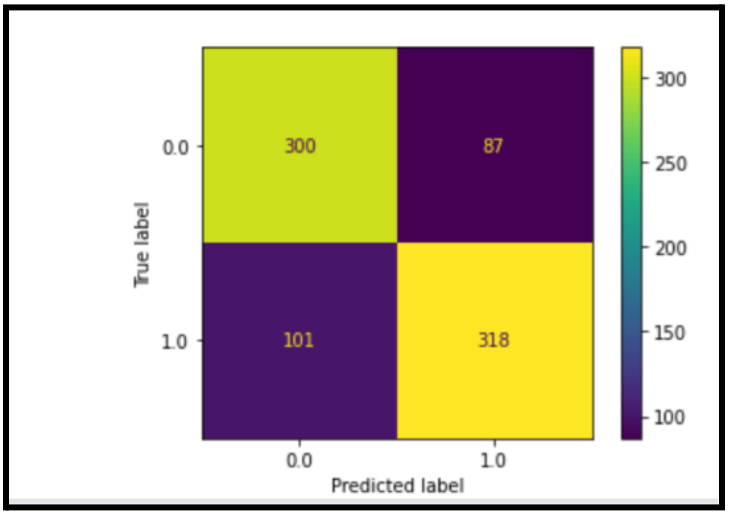

Fig no: 06 Confusion Matrix Logistic Regression

ii. Random Forest:

A Random Forest is a classifier consisting of a collection of tree structured classifiers (…) independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x .

This formula uses the class and probability to determine the Gini of each branch on a node, determining which of the branches is more likely to occur. Here, pi represents the relative frequency of the class you are observing in the dataset and c represents the number of classes.

- The accuracy of the Random Forest Model was **82%** on the test dataset.
- Confusion Matrix:

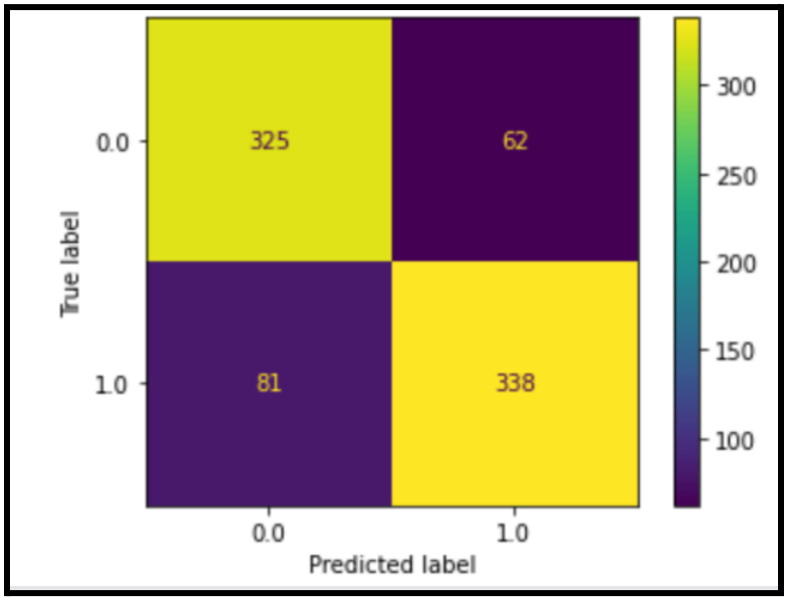

Fig no: 07 Confusion Matrix using Random Forest

2.2.3. Challenges and Comprehension while drafting solution

We needed more computational power to run a proper pipeline for our models as the process is very computationally intensive, this severely limited us in the hyperparameters tuning.

3.2. RESULTS

	Model	Accuracy Score
0	Logistic Regression	0.767
1	Random Forest	0.815
2	SVM	0.815
3	KNN	0.692
4	Decision Tree	0.711

FIg No: 08 Accuracy Score Tab

4. CONCLUSION

The real world application of the development can be used for drones system where drone can adapt according to the classification of our categories

In the future enhancements of our project by the application of deep learning, specifically deep convolutional neural networks (DCNNs), to the classification of remotely-sensed imagery of natural landscapes has the potential to greatly assist in the analysis and interpretation of geomorphic processes.

We can apply the method to several sets of images of natural landscapes, acquired from satellites, aircraft, unmanned aerial vehicles, and fixed camera installations to examine the general effectiveness of transfer learning to landscape-scale image classification.

5. REFERENCES

[1] Nisarg Vora, Arush Patel, Kathan Shah, Pallabi Saikia, "'Land Cover Classification from Satellite Data using Machine Learning Techniques", 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV) ,Jan 2022.

[2] Clopas Kwenda , Mandlenkosi Gwetu, and Jean Vincent Fonou Dombeu, "Machine Learning Methods for Forest Image Analysis and Classification", Apr 2022

[3] Daniel Buscombe, Andrew C. Ritchie, "Landscape Classification with Deep Neural Networks",Geosciences 2018, Jul 2018

[4] Akhtar Jamil, Aftab Ahmed Khan, Alaa Ali Hameed, Sibghat Ullah Baza, "Application of Machine Learning Approaches for Land Cover Classification", Journal of Applied and Emerging Sciences Vol (11), Jun 2021

[5] Brad Conlin, Umar Ruhi,"Current Research Landscape of Machine Learning Algorithms in Online Identity Fraud Prediction and Detection", 2021 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), Mar 22

GitHub repository with all the relevant files:

https://github.com/alexvecchiettini/image-processing.git