

SATELLITE IMAGE DETECTION USING DEEP LEARNING ALGORITHM, CONVOLUTIONAL NEURAL NETWORKS.

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology degree in Information Technology

by

IMMACULATE MARIA FERNANDO

(Reg. No: 38120031)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600

119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Ms.Immaculate Maria Fernando (Reg. Np. 38120031) who carried out the project entitled " **Satellite Image Detection using Deep Learning Algorithm, Convolutional Neural Networks**" under my supervision from November 2021 to April 2022.

Internal Guide

Dr.R.Subhashini, M.E, Ph.D

Head of the Department

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.R.Subhashini M.E., Ph.D. , Head** of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.R.Subhashini M.E.,Ph.D.**, for her valuable guidance, suggestions and constantencouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

DECLARATION

I **Immaculate Maria Fernando (Reg. No. 38120031)** hereby declare that the Project Report entitled **SATELLITE IMAGE DETECTION USING DEEP LEARNING ALGORITHM, CONVOLUTIONAL NEURAL NETWORKS.** done by me under the guidance of **Dr. R. Subhashini, M.E., (Ph.D)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in INFORMATION TECHNOLOGY.

DATE:

PLACE: CHENNAI.

SIGNATURE OF THE CANDIDATE

ABSTRACT

The applications of AI are known far and wide. Every application, and every service uses AI in some or the other way. AI in the field of remote sensing, has applications that includes land cover analysis, image segmentation, hyperspectral image analysis, vegetation and land change detection and so on. Due to the rich available libraries that are able to extract even the smallest pixel, AI is vastly used in the field of image recognition and processing. AI is a broad term that envelops certain terms or processes that frequently appear while considering AI, i.e., Machine Learning and Deep Learning. These two methodologies contain separate groups of algorithms, each having their own sets of accuracy scores and metrics.

There are certain types of applications that require a ML approach. These applications would not function the best in case of using Deep Learning Algorithm. Simply put, Machine Learning Algorithms are useful when considering small scale application, whereas, Deep Learning is used for much larger and complex applications. Change Management is a technique wherein, changes are observed between two objects that are taken over a period of time. The end result, the difference image, would portray the changes that have been detected between the two objects.

Change Management, in case of remote sensing, is used to study regions or places of high interest over a year or a few and note the changes observed between the two periods of time. The difference images obtained can be used to study the land degradation, deforestation rates, water scarcity or water surplus levels, urbanization, etc. There are many ways that change management can be performed using AI, the method proposed here, is using the Deep Learning Algorithm, CNN, to detect changes between two satellite images taken over a period of time.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	I
	TABLE OF CONTENTS	II
	LIST OF ABBREVIATION	IV
	LIST OF FIGURES	V
1	INTRODUCTION	1
	1.1 Introduction	1
	1.2 Background	2
	1.3 Problem statement	3
	1.4 Motivation	3
2	LITERATURE REVIEW	4
	2.1 Literature survey	4
	2.2 Objectives	5
	2.3 Problem definition	5
3	EXPERIMENTAL METHODS AND ANALYSIS	6
	3.1 Proposed plan of work	6
	3.1.1 Existing work	6
	3.1.2 Proposed work	6
	3.1.3 Advantages of proposed system	6
	3.2 Methodology	7
	3.2.1 Modules	7

4	SYSTEM REQUIREMENTS AND DESIGN	
	4.1 Requirements	10
	4.1.1 Hardware Requirements	10
	4.1.2 Software Requirements	10
	4.2 System Architecture	11
	4.2.1 Use case diagram	12
5	RESULTS	
	5.1 Software setup	13
	5.2 Experimental Results	13
6	CONCLUSION	
	6.1 Conclusion	16
	6.2 Future Work	16
	REFERENCE	17
	APPENDIX	18
	SOURCE CODE	18

LIST OF ABBREVIATION

1. AI – Artificial Intelligence
2. ML – Machine Learning
3. DL – Deep Learning
4. RS – Remote Sensing
5. CM – Change Management

LIST OF FIGURES

Figure no	Figure name	Page no
4.1	System Architecture	11
4.2	Input Image 1	12
5.1	Input Image 2	13
5.2	Original Image	14
5.3	Grey-scaled Image	15
5.4	Output Image	15

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The need to develop accurate algorithms as a means to identify the changes that are detected between two satellite images, has arisen due to the fact that traditional image recognition methods such as MATLAB, Octave and other image processing applications tend to display contents with missing values or inaccurate metrics. This can be considered as a float value in the case of small applications. But, in large scale, the costs of such inaccuracies are threefold. There are certain things that need to be considered in the case of satellite images. There are factors such as image resolution, the blur factor, and the clouds are some of them.

Training an image that is blurry or has dense patches of clouds over the region, may result in highly inaccurate results with the statistical values crossing the threshold. The introduction of the Machine Learning Algorithms for change detection, has brought forth considerable amounts of improvements in the field of image analysis. Machine Learning Algorithms such as Support Vector Machines and K-Means Clustering Algorithms provide a wide range of libraries that are used for image detection and analysis.

Each algorithm has its own set of advantages and disadvantages. The growth of AI lies in the fact that, when a certain procedure does not perform as expected, improvements can be done to enhance the efficiency of the algorithms, thereby, introducing new and improved versions of the previous algorithms. Using Machine Learning Algorithms, change detection can be performed, but only to an extent. Considering K-Means Clustering Algorithm, the change detection can be performed over a set of images, passed as parameters to the K-Means function, which returns the changes as a grey scale image.

In worst cases, the K-Means Algorithm does not display the areas of land that appear to be darker than the areas surrounding them. This can be explained as the growth of vegetation surrounding the area. Manually, the changes can be detected and identified. Considering the value 1 to represent grasslands and 0 as waterbodies, the algorithm displays the grasslands value as 0. In certain small-scale application that does not involve in any predictions, this can be manually verified. But, the purpose of developing algorithms is for the purpose

of making future predictions.

1.2 BACKGROUND

With the disadvantages of using Machine Learning methods, the need for algorithms with accuracy values scoring higher than the traditional methods, paved the way for Deep Learning algorithms. The Deep Learning algorithms are powerful, with the capabilities of processing millions and billions of images in a dataset. The popular Convolutional Neural Networks (CNN) algorithm is used for image analysis and detection. The CNN algorithm takes an image as an input, which is then processed and classified according to the labels it is trained on. The computer, which views the images as an array of pixels, determines the size of the array of the dimensions of the array depending on the resolution of the image. The image is processed through a series of convolution layers that contain pooling and fully connected layers. The probabilistic value, which lies between 0 and 1, is output as a function of the Soft-max function. The traditional methods in Machine Learning involves a lot of preprocessing to be performed. The use of Convolutional Neural Networks, greatly reduces the pre-processing required. The CNN are capable of learning the methods that were manually done for processing images.

An image is made up of pixels which correspond to the height, width and dimensions of the image. There are certain areas or weights in an image that involves usability and certain weights which are redundant. The CNN algorithm filters these redundant weights that affects the performance of the algorithm. The main reason for using CNN algorithm is to process the images in a reduced way without losing any important features which is important to get accurate predictions. The first part of a convolutional layer is the kernel/filter which moves through the image matrix to collect certain features such as the edges and colour gradients.

The next layer of the CNN is the pooling layer, which is used to decrease computational power that takes to process the image features. There are two types of Pooling which are Max Pooling and Average Pooling. The Max Pooling returns the maximum value of the portion of the image that is covered by the Kernel. The Average Pooling returns the average of all the values that the kernel covers. Both the convolutional layer and the pooling layer form the i^{th} layer of the CNN. As the complexity of the image's increases, so does the number of layers in the CNN. The fully connected layer is the final stage of the CNN algorithm.

1.3 PROBLEM STATEMENT

To detect the changes observed between two satellite images taken over a period and perform various inferences from the difference image using Deep Learning Algorithm. This study aims to provide the methodologies that are involved in change management using CNN algorithm in Deep Learning. Each process is discussed with the steps and the supporting diagrams that are labelled.

1.4 MOTIVATION

Satellite imaging is gaining importance in many applications like remote surveillance, environmental monitoring, aerial survey etc. All these applications involve searching objects, event of interest, facilities etc., from the satellite images. In most applications, manual detection and classification of objects becomes very difficult especially with large volumes of data and the number of satellite images to process collectively. In the image processing domain, detection in satellite (aerial) images is more challenging as the objects are small and their visual features are extremely hard to track and capture making it more difficult. Many deep learning models based on convolutional neural network (CNN) are proposed for detection and classification of objects in satellite images. These models involve two steps. In the first step, the regions of presence of object in the image are detected. In the second step, the objects are classified using convolutional neural network.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

The CNN algorithm is implemented in the Data Science tool, Jupyter Notebook of Anaconda. The Jupyter Notebook kernel is used to run the. ipynb notebooks. There are various other online tools that can be used to run the. ipynb notebooks such as Kaggle and Google Collabs. Based the complexity of the implementation, the suitable tool can be used. For instance, for large datasets that involves the usage of a GPU, Kaggle can be a good option to opt for. Kaggle has an inbuilt GPU and cloud storage with more than 20 GB processing data per single notebook.

Since the current study involves images with varying resolution and size, the Jupyter Notebook is much suitable. The images are downloaded from the respective satellite that covers the region and a portion of the image is selected for the change detection process. The image, in case of a grey-scale, can be directly input to the algorithm. In case of coloured images, using Matplotlib library, the image is converted into gray-scale and then input into the algorithm.

The input images are placed in the same directory as the notebook. The images after filtering, output a complete black and white image. This makes the process of differencing the images much easier. The images are input to the CNN which then performs the change detection and the output of the change detection is the image that is obtained as a result of the process. This image can be used for further analysis to determine the various elements and land regions that are subject to change.

The end result, the difference image, would portray the changes that have been detected between the two objects. Change Management, in case of remote sensing, is used to study regions or places of high interest over a year or a few and note the changes observed between the two periods of time. The difference images obtained can be used to study the land degradation, deforestation rates, water scarcity or water surplus levels, urbanization, etc. There are many ways that change management can be performed using AI, the method proposed here, is using the Deep Learning Algorithm, CNN, to detect changes between two satellite images taken over a period of time.

2.2 OBJECTIVES

- To effectively detect the satellite images and convert them into their respective grayscale versions.
- To use Deep Learning Algorithm, CNN, to flatten the image and detect the changes between both the images using TensorFlow.
- To display the changes detected between both the images.

2.3 PROBLEM DEFINITION

Detecting the changes observed between two images can be daunting if it is done manually. Due to the development of recent technologies, we have found the implementation of AI in the field of Remote Sensing helps to detect the changes observed between two satellite images taken over a period effectively. This not only helps us know better about the environment around us, but also helps us understand the impact of AI in almost any field. This study aims to provide the methodologies that are involved in change management using CNN algorithm in Deep Learning. The processes that are discussed in this project study is labelled along with the images and the output results respectively.

CHAPTER 3

EXPERIMENTAL METHOD AND ANALYSIS

3.1 PROPOSED PLAN OF WORK

3.1.1 EXISTING WORK

Using Machine Learning Algorithms, change detection can be performed, but only to an extent. Considering K-Means Clustering Algorithm, the change detection can be performed over a set of images, passed as parameters to the K-Means function, which returns the changes as a grey scale image. In worst cases, the K-Means Algorithm does not display the areas of land that appear to be darker than the areas surrounding them. This can be explained as the growth of vegetation surrounding the area. Manually, the changes can be detected and identified. Considering the value 1 to represent grasslands and 0 as waterbodies, the algorithm displays the grasslands value as 0. In certain small-scale application that does not involve in any predictions, this can be manually verified. But, the purpose of developing algorithms is for the purpose of making future predictions.

3.1.2 PROPOSED WORK

The Deep Learning algorithms are powerful, with the capabilities of processing millions and billions of images in a dataset. The popular Convolutional Neural Networks (CNN) algorithm is used for image analysis and detection. The CNN algorithm takes an image as an input, which is then processed and classified according to the labels it is trained on. The computer, which views the images as an array of pixels, determines the size of the array of the dimensions of the array depending on the resolution of the image. The image is processed through a series of convolution layers that contain pooling and fully connected layers.

3.1.3 ADVANTAGES OF PROPOSED WORK

- Effective detection of satellite images and conversion of the same from coloured to grey-scaled images.
- Flattened image produced without any loss of properties due to conversion from pixels into a NumPy array.
- Display of a change map which shows the changes that are observed between both the images.

3.2 METHODOLOGY

3.2.1 MODULES

Module 1.1:

Input: Raw Satellite Image (coloured).

Process: Read the Satellite Image using the Python OpenCV library for image processing.

Output: The image is displayed as a graph.

Module 1.2:

Input: Raw Satellite Image (coloured).

Process: Display the image as a grayscale version. This is useful for change detection since it is easier to predict changes in a black and white image.

Output: The grayscale version of the image is displayed.

Module 2:

Input: Grayscale Image.

Process: The image is flattened into a NumPy array, and the image edges are determined for further processing.

Output: The flattened array.

Module 3:

Input: The array containing the pixels and the image dimensions (Flattened array).

Process: The array is input into the PyTorch library which clears noise in the array. The number of filters needed depends upon the image clarity. Higher resolution images need a minimum of 10 to 15 filters.

Output: The model dimensions are displayed.

Module 4:

Input: The filtered array.

Process: The array is now converted back to an image and displayed in grayscale.

Output: The processed grayscale image.

Module 5.1:

Input: Raw Satellite Image (coloured) – The second image for change detection.

Process: Read the Satellite Image using the Python OpenCV library for image processing.

Output: The image is displayed as a graph.

Module 5.2:

Input: Raw Satellite Image (coloured) – Image 2.

Process: Display the image as a grayscale version. This is useful for change detection since it is easier to predict changes in a black and white image.

Output: The grayscale version of the image is displayed. (Modules 2 and 3 are repeated for the second image as well)

Module 6:

Input: Filtered Array (Image 1) and Filtered Array (Image 2)

Process: The difference between these two arrays are calculate and any negative values are made positive through the abs() function.

Output: The difference between Image 1 and Image 2

Module 7

Input: The array containing the difference between the input images.

Process: The array is converted back to an image and displayed as the changes detected between both the images.

Output: The image containing the changes noticed between both the input images.

CHAPTER 4

SYSTEM REQUIREMENT AND DESIGN

4.1 REQUIREMENT

4.1.1 Software Requirements

SYSTEM:

- Front-end / GUI Tools: Kaggle, A Collaboratory software for Artificial Intelligence Applications.
- Operating System: Windows 10 Pro N
- Language: Python
- Anaconda, Jupyter Notebook
- Google Collab

4.1.2 Hardware Requirements

OS: WINDOWS 7 OR ABOVE

- Processor: GPU (Kaggle in-built)
- RAM: 16 GB
- Hard Disk Drive: None

4.2 SYSTEM ARCHITECTURE

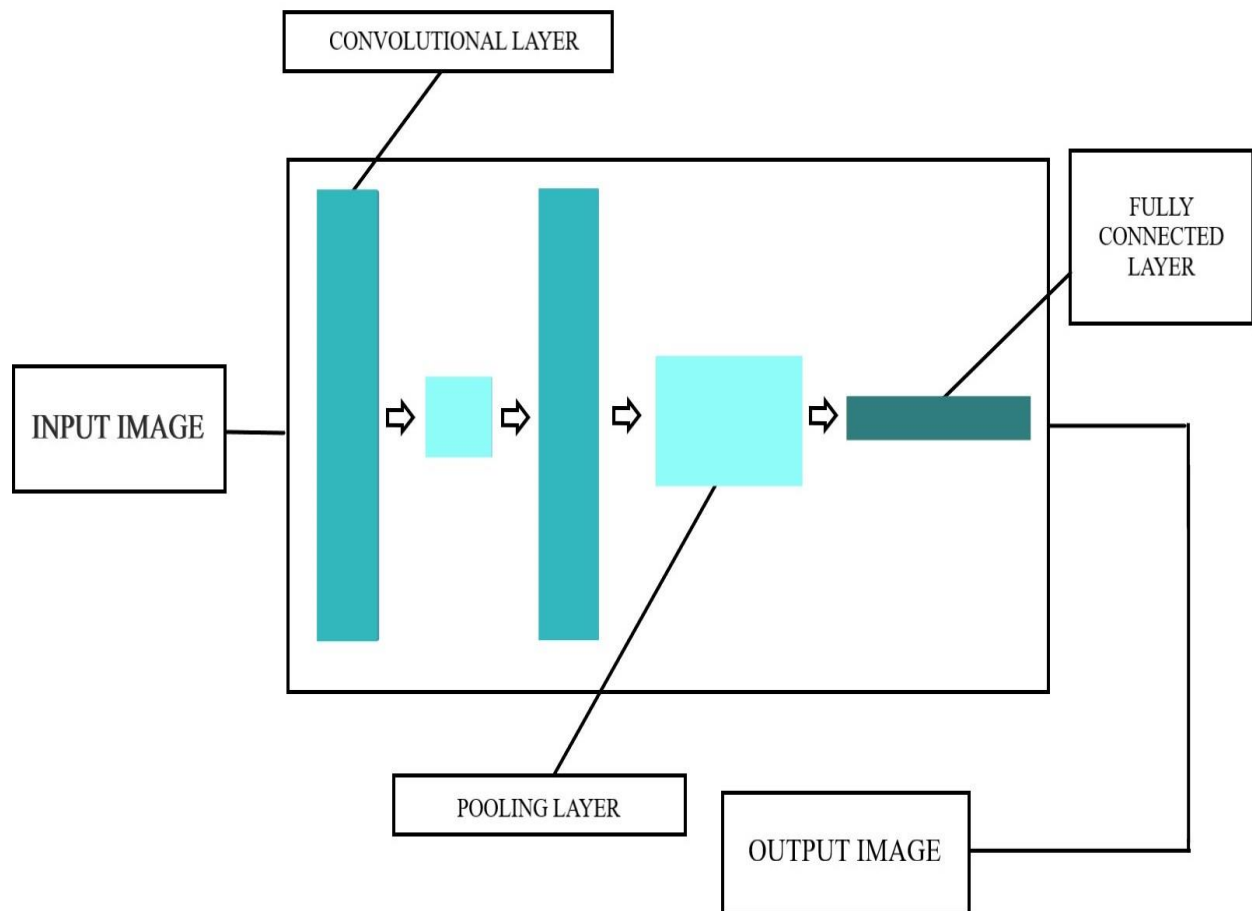


Fig. 4.1 System Architecture

CNN takes an image as input, which is classified and process under a certain category such as dog, cat, lion, tiger, etc. The computer sees an image as an array of pixels and depends on the resolution of the image. Based on image resolution, it will see as $h * w * d$, where h = height w = width and d = dimension. For example, An RGB image is $6 * 6 * 3$ array of the matrix, and the grayscale image is $4 * 4 * 1$ array of the matrix.

In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.

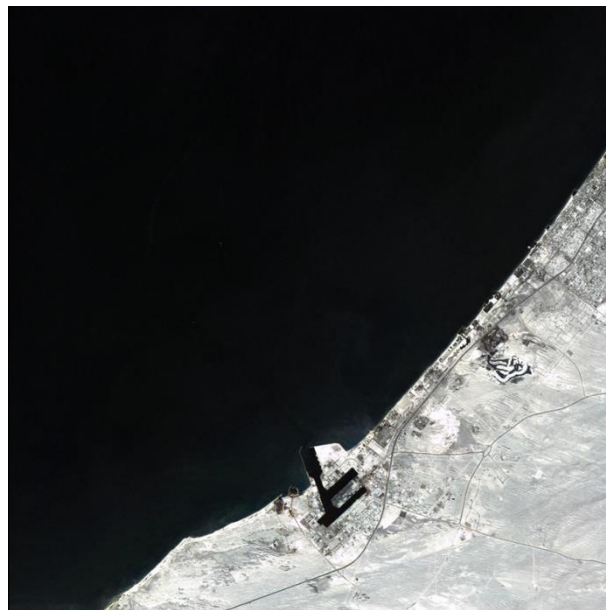
CHAPTER 5

RESULTS AND DISCUSSION

5.1 RESULTS

INPUT IMAGES

The satellite images are checked for clouds and other factors such as resolution and blurriness. The images should be of the same dimensions and the same size. The images are processed and filtered before feeding it into the algorithm for change detection.



Fig, 4.2 Input Image 1



Fig 4.3 Input Image 2

COLORED TO GREY-SCALED IMAGES

The images are first subject to gray scaling where the images are checked for colorization. If the images are colored, they are converted into gray scale. The image is flattened into a NumPy array and the edges are determined for further processing. The same procedure is done for the second image. The output of the first module is the flattened array.

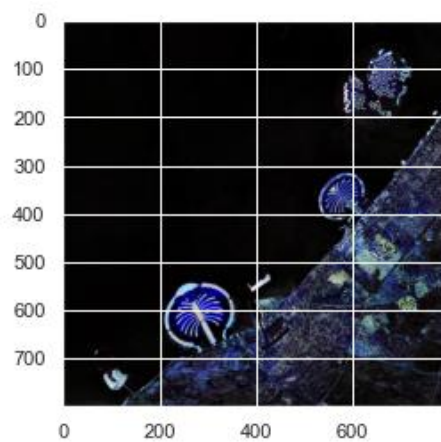


Fig 4.4 Original Image

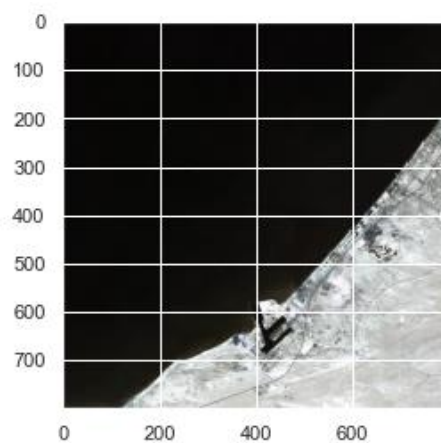


Fig. 4.5 Grey-scaled Image

DIMENSIONALITY REDUCTION

The array that contains the pixels and the image dimensions, i.e., the flattened array is input into the Pytorch library which clears the noise in the array. The number of filters that are required depends on the clarity of the image. High resolution images require a minimum of 10 to 15 filters. The model dimensions are displayed as an output. The filtered array is converted back into a gray scale image to be displayed as a process of displaying the change after change detection is performed.

CHANGE DETECTION

Change Detection is performed using the Convolutional Neural Network Algorithm that outputs the image that is obtained as a result of the difference between the original image and the image that is taken over a period of time of the same region. The input and the output images obtained are displayed in the figures below. The figure 1.1 and 1.2 displays the input image after the dimensionality reduction is performed. The figure 2 shows the final output that is obtained after the change detection process. The regions that are taken here are display the urbanization



Fig. 4.6 Output Image

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

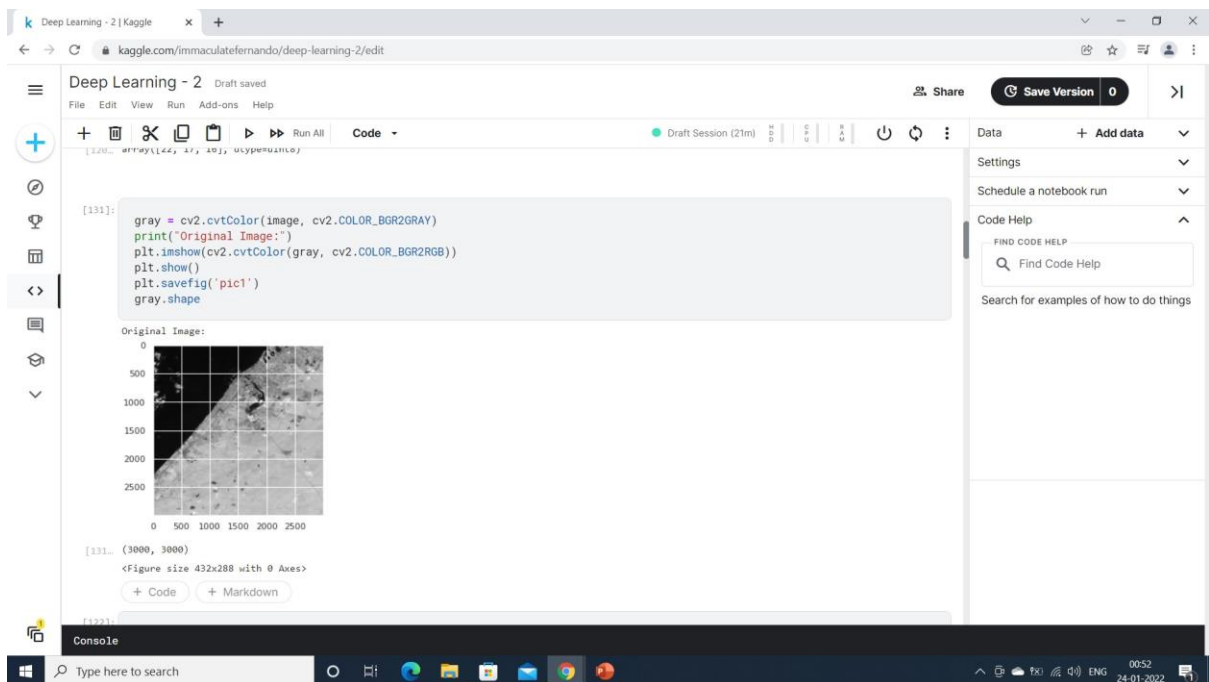
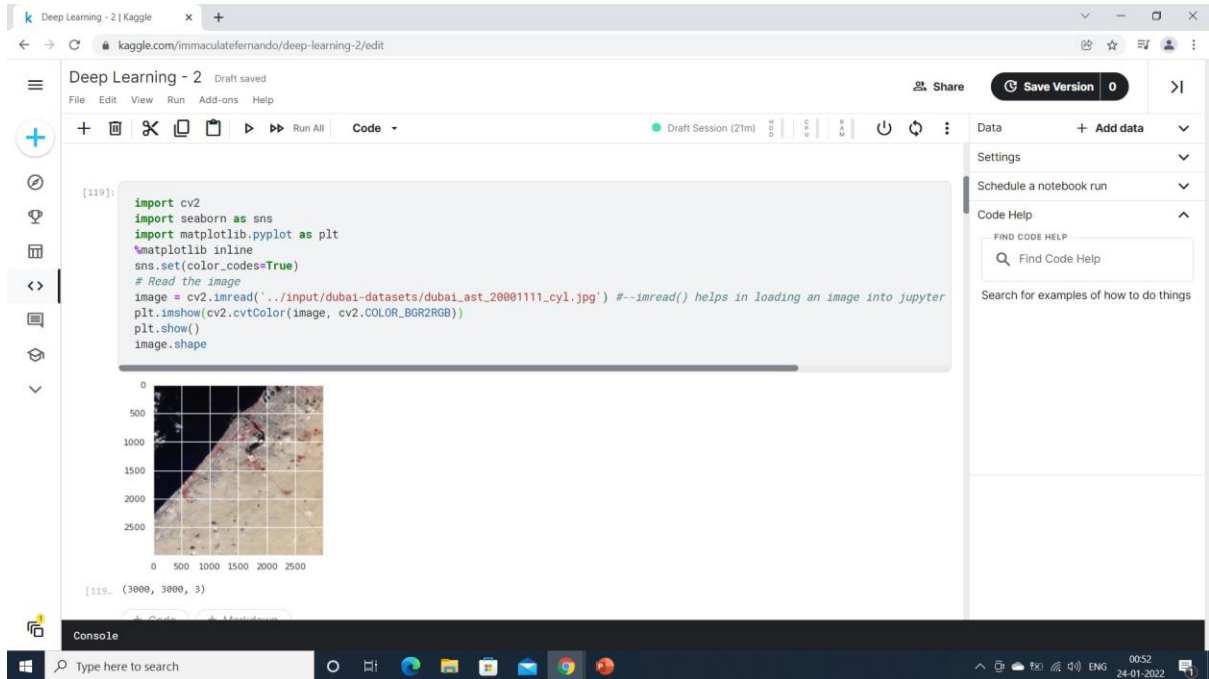
The convolutional neural network algorithm provides a rich set of libraries that can be used to perform such operations such as filtering, dimensionality reduction, and so on. Converting the image to grey scaled, does not result in any data loss, but rather makes the process of change conversion much simpler. It is rather difficult to calculate the changes with the RGB colours involved as the size of the matrices will not be suitable to perform any sort of operations on them. The below images display the original image as a graph, followed by the grayscale image. The final output of the change detection is displayed.

6.2 FUTURE WORK

The results obtained from the reference image is used to perform further analysis and derive conclusions from it. This method is more advantageous over the traditional K-Means clustering method since all the pre-processing methods are reduced and the filtering method involves just passing the input array into a function. With further research, the CNN algorithm can take in more input images rather than just two and simultaneous processes such as filtering, converting into gray scale images and other operations can be performed. AI evolves every day and each day, it brings forth new changes and developments in the field of remote sensing. In the coming days, the predictions done over satellite images can be more reliable and accurate.

APPENDIX

SOURCE CODE



Deep Learning - 2 | Kaggle

kaggle.com/immaculatefernando/deep-learning-2/edit

Deep Learning - 2

Draft saved

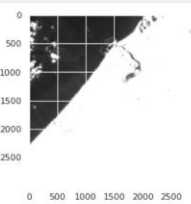
File Edit View Run Add-ons Help

Share Save Version 0

+ Draft Session (22m)

```
mat_y = np.array([[ -1, -2, -1],
                  [ 0, 0, 0],
                  [ 1, 2, 1]])
mat_x = np.array([[ -1, 0, 1],
                  [ 0, 0, 0],
                  [ 1, 2, 1]])

filtered_image = cv2.filter2D(gray, -1, mat_y)
plt.imshow(filtered_image, cmap='gray')
filtered_image = cv2.filter2D(gray, -1, mat_x)
plt.imshow(filtered_image, cmap='gray') # 3x3 array for edge detection
plt.savefig('pic2')
```



+ Code + Markdown

Console

Type here to search

00:53 24-01-2022