

Classification Of Damaged Vegetation Areas Using Convolutional Neural Network Over Satellite Images

by

Samiha Haque
17101218
Nazibur Rahman
17101317

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

© 2021. Brac University
All rights reserved.

Declaration

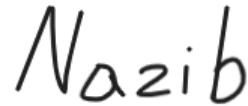
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Samiha Haque
17101218



Nazibur Rahman
17101317

Approval

The thesis/project titled “Classification Of Damaged Vegetation Areas Using Convolutional Neural Network Over Satellite Images” submitted by

1. Samiha Haque (17101218)
2. Nazibur Rahman (17101317)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 08, 2021.

Examining Committee:

Supervisor:
(Member)



Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Mahbub Alam Majumdar
Professor
Department of Computer Science and Engineering
Brac University

Abstract

Forests and wild vegetation have always been highly significant natural resources throughout history and play a crucial role in keeping the climate and ecosystems well balanced. Over the years, there has been a growing decline in the forest areas all over the world due to hurricanes, landslides and most notably forest fires which have been directly linked to global warming. Very recently, large fires in Brazilian Amazon, Australia, Thailand USA have had devastating effects, destroying millions of acres of vegetation. There has been a rise in frequency and severity of such forest disasters over the years and therefore, understanding the scale of damage on the forest areas is very important. For this purpose, multispectral high resolution remote sensing data from satellites such as Sentinel-2 can be used to extract spatial information before and after the disaster, identify the damaged areas and evaluate the amount damaged. In this research, we construct a dataset from the difference image of disaster struck vegetation areas and a pre-trained ResNet-50 is used to produce flattened feature maps from the difference image patches. These feature maps are then labelled using the K-Means clustering algorithm, thus creating a complete labelled dataset which is used to train two CNN models to classify the areas into undamaged and damaged classes for Model-1 with 89.77% test accuracy and undamaged, mildly damaged, and severely damaged classes for Model-2 with 85.69% test accuracy. Model-1 recorded 89.77% in micro F1-score, 76.99% for kappa score and 0.10 for overall error and Model-2 had 85.69%, 72.87% and 0.14 values in micro F1-score, kappa score and overall error respectively.

Keywords: Machine Learning; CNN; Convolutional Neural Network; K-Means; Change Detection; Multispectral Image; Transfer Learning; ResNet-50; Sentinel-2;

Acknowledgement

First and foremost, We thank Allah for keeping us in good health throughout the Corona pandemic and for blessing us with the ability and resources needed to carry out our research within the projected timeline.

We express our heartfelt gratitude to our supervisor Mr. Moin Mostakim, Lecturer of Computer Science and Engineering Department of Brac University, for his immense support and guidance throughout this research. We are also thankful to our parents for their undying love and support and finally, we wish to thank everyone who helped us directly or indirectly.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	x
Nomenclature	xii
1 Introduction	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Aims and Objectives	2
1.4 Thesis Outline	2
2 Background Information	4
2.1 Related Work	4
2.2 Machine Learning	9
2.2.1 Supervised vs Unsupervised Learning	9
2.3 Convolutional Neural Network	9
2.3.1 Convolutional Layer	10
2.3.2 Activation Function	11
2.3.3 Pooling layer	11
2.3.4 Fully Connected Layer	11
2.3.5 Optimizer	11
2.4 ResNet-50	11
2.5 Kmeans	12
2.6 Sentinel-2 Satellite	13
3 Data Collection and Preprocessing	15
3.1 Image collection	15
3.2 Band Selection	20
3.3 Band scaling	20

3.4	Making Image Crops	24
3.5	Transfer Learning Using ResNet-50	25
3.6	K-means	25
3.6.1	K-Means For Model-1	26
3.6.2	K-Means For Model-2	34
3.6.3	Algorithm for visualization	43
3.7	Train Test set	44
4	Model and Evaluation	45
4.1	Model-1	45
4.2	Model-2	46
4.3	Evaluation	47
4.3.1	Model-1 Results	47
4.3.2	Model-2 Results	53
4.3.3	Analysis	58
5	Conclusion and Future Work	59
	Bibliography	63

List of Figures

1.1	Workflow Diagram	3
3.1	Geographical location of Bolivia	16
3.2	Geographical location of Bombala	16
3.3	Geographical location of California	16
3.4	Geographical location of Canberra	17
3.5	Geographical location of Harrietville	17
3.6	Geographical location of Kangaroo Island left side	17
3.7	Geographical location of Kangaroo Island right side	18
3.8	Geographical location of New South Wales	18
3.9	Geographical location of Salem	18
3.10	Geographical location of Chiang Mai	19
3.11	Geographical location of Tucson	19
3.12	Geographical location of Victoria	19
3.13	Before image, After image and Difference image of Bolivia	21
3.14	Before image, After image and Difference image of Bombala	21
3.15	Before image, After image and Difference image of California	21
3.16	Before image, After image and Difference image of Canberra	22
3.17	Before image, After image and Difference image of Harrietville	22
3.18	Before image, After image and Difference image of Kangaroo Island left side	22
3.19	Before image, After image and Difference image of Kangaroo Island right side	22
3.20	Before image, After image and Difference image of New South Wales	23
3.21	Before image, After image and Difference image of Salem	23
3.22	Before image, After image and Difference image of Chiang Mai	23
3.23	Before image, After image and Difference image of Tucson	23
3.24	Before image, After image and Difference image of Victoria	24
3.25	Elbow diagram for NSW, Bolivia, Chiang Mai,Victoria, Canberra, Salem, Bombala and Harrietville	27
3.26	Elbow diagram for Kangaroo Island (both left and right sides)	27
3.27	Elbow diagram for California	28
3.28	Model-1 K-means output for Bombala	29
3.29	Model-1 K-means output for Bolivia	29
3.30	Model-1 K-means output for California	30
3.31	Model-1 K-means output for Canberra	30
3.32	Model-1 K-means output for Harrietville	31
3.33	Model-1 K-means output for Kangaroo Island left side	31

3.34	Model-1 K-means output for Kangaroo Island right side	32
3.35	Model-1 K-means output for New South Wales	32
3.36	Model-1 K-means output for Salem	33
3.37	Model-1 K-means output for Chiang Mai	33
3.38	Model-1 K-means output for Tucson	34
3.39	Model-1 K-means output for Victoria	34
3.40	Elbow diagram for Chiang Mai, Kangaroo Island right side , Kan- garoo Island left side, California, Victoria, Bolivia, Harrietville and Canberra	35
3.41	Elbow diagram for Salem and Tucson	36
3.42	Elbow diagram for NSW and Bombala	36
3.43	Model-2 K-means output for Bombala	37
3.44	Model-2 K-means output for Bolivia	38
3.45	Model-2 K-means output for California	38
3.46	Model-2 K-means output for Canberra	39
3.47	Model-2 K-means output for Harrietville	39
3.48	Model-2 K-means output for Kangaroo Island left side	40
3.49	Model-2 K-means output for Kangaroo Island right side	40
3.50	Model-2 K-means output for New South Wales	41
3.51	Model-2 K-means output for Salem	41
3.52	Model-2 K-means output for Chiang Mai	42
3.53	Model-2 K-means output for Tucson	42
3.54	Model-2 K-means output for Victoria	43
3.55	Pseudocode for visualizing k-means labels	43
4.1	Model-1 architectire	45
4.2	Model-2 architectire	46
4.3	Model-1 Training vs Validation curve	48
4.4	Model-1 ROC Curve	49
4.5	Model-1 Heatmap for Test Data	49
4.6	Model-1 Heatmap for NSW	50
4.7	Model-1 Heatmap for California	50
4.8	Model-1 Heatmap for Victoria	51
4.9	Model-1 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for NSW where red is dam- aged, green is undamaged and blue is the misclassification	51
4.10	Model-1 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for image California where red is damaged, green is undamaged and blue is the misclassification .	51
4.11	Model-1 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for Victoria where red is damaged, green is undamaged and blue is the misclassification	52
4.12	Model-2 Training vs Validation curve	54
4.13	Model-2 Heatmap for Test Data	54
4.14	Model-2 Heatmap for NSW	55
4.15	Model-2 Heatmap for California	55
4.16	Model-2 Heatmap for Victoria	56

4.17	Model-2 Comparison of "Test image" (far left), "Predicted image" (center) and "Unmatched labels image" (far right) for NSW where red is severely damaged, yellow is mildly damaged, green is undamaged and blue is the misclassification	56
4.18	Model-2 Comparison of "Test image" (far left), "Predicted image" (center) and "Unmatched labels image" (far right) for California where red is severely damaged, yellow is mildly damaged, green is undamaged and blue is the misclassification	57
4.19	Model-2 Comparison of "Test image" (far left), "Predicted image" (center) and "Unmatched labels image" (far right) for Victoria where red is severely damaged, yellow is mildly damaged, green is undamaged and blue is the misclassification	57

List of Tables

2.1	ResNet-50 Architecture [23]	12
2.2	Sentinel-2 Band Information[27]	14
3.1	Details of Image Crops	24
4.1	Model-1 metrics results	52
4.2	Predicted Damaged Areas for Model-1	53
4.3	Model-2 metrics results	57
4.4	Predicted Damaged Areas for Model-2	58

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AA average accuracy

AdaGrad Adaptive Gradient

ANN Artificial Neural Network

ARVI Atmospherically Resistant Vegetation Index

CIFAR Canadian Institute for Advanced Research

CNN Convolutional Neural Network

DBN Deep Belief Network

DFC2018 Data Fusion Contest 2018

DI Difference Image

DKTI Double-Thresholding KI algorithm

EVI Enhanced Vegetation Index

fc fully connected

FCM Fuzzy c-means

FLICM Fuzzy local information c-means

FN false negatives

FP false positives

JC Jaccard coefficient

KC kappa coefficient

KWFLICM kernel weighted FLICM

ML Machine Learning

MSI multi-spectral instrument

NAIP National Agriculture Imagery Program

NDVI Normalized Difference Vegetation Index

NIR Near Infra Red

NMI normalized mutual information

NSW New South Wales

OA overall accuracy

OE overall error

PA per-class accuracy

PCA Principal Component Analysis

PCC percentage correct classification

ResNet Residual Networks

RGB Red Green Blue

RMSprop Root Mean Squared propagation

RN Relation Networks

SAE Sparse Auto Encoder

SAR Synthetic-aperture radar

SDA Stacked Denoising Autoencoder

SSE sum of square error

SVM Support Vector Machine

SWIR short wave infrared spectral range

TN true negatives

TP true positives

VM Virtual Machine

VNIR visible near infrared

YC Yule coefficient

DN digital number

Chapter 1

Introduction

1.1 Introduction

Forests and jungles are home to various species of vegetation and wildlife. However, as the world population and economy is growing everyday, forests and natural resources are at stakes for providing for mankind. Illegal logging leading to deforestation together with overall global temperature rise play a major catalyst in forest fires and landslides. These have a long-term devastating impact on the wildlife, vegetation and the overall ecosystem of our planet. According to [1], forests cover 30% of the world's area and yet they are ceasing to exist due to natural and man induced reasons. The causes of deforestation include farming, grazing of livestock, illegal logging, mining, building roads through forests, wildfires and overgrazing. These heavily impact the ecosystem and push wildlife species towards extinction. Therefore, in this research, we will develop a method for classifying damaged vegetation and also evaluate the size of the damaged area by using temporal changes of multispectral images from the same region. Firstly, we will extract flattened feature maps from DI crops of the satellite images using a pre-trained ResNet-50. Later, we will use K-Means clustering algorithm to obtain labels for all the flattened feature maps for two models. This will create our dataset which will contain the flattened image crops and the corresponding labels. Hence, it will help us to carry out a supervised classification of the landscape. Finally, the prepared dataset will be fed into CNN to classify the damaged areas into two classes: "undamaged" and "damaged" for Model-1 and into three classes: "undamaged", "mildly damaged" and "severely damaged" for Model-2. This method can contribute towards obtaining an estimation and the severity of large damaged areas of the forest. Thus, reforestation plans can be made swifter and more accurate and restoring the damaging effects, in order to slow down the global warming rates, can be more fruitful in the future.

1.2 Problem Statement

The alarming rate of forest destruction indicates a great increase in the rate of global warming which in the long run can cause negative impacts on all forms of life on Earth. According to [2], Brazilian Amazon lost 17.5 million acres of forest land in 2019 and Australian forests lost 25.5 million acres of forest land in 2020 and 1 billion animals are feared dead due to massive forest fires. It also mentions that it is often difficult to estimate the size of these huge damaged areas. Even though

fire mitigation and reforestation techniques are known to people but most times the damaged area is incomprehensible. Therefore, a solution is needed to visualize and locate these damaged vegetation areas in order to quicken these processes. Another difficulty is that there is a scarcity of labelled satellite data. If any new destruction hits the forests, then there will not be any labelled satellite data available and manually labeling the satellite images will be inefficient and will ultimately fail the purpose of the whole scheme of rapid reforestation. Hence, this research will solve this problem by using an unsupervised approach. Satellite images from Sentinel-2 are readily available to the public and by classifying regions of the affected areas can provide us with an estimation and severity of the damaged vegetation after the destruction has occurred. These details along with necessary available resources can be used for reforestation and preserving our ecosystem.

1.3 Aims and Objectives

The goal of this research is to classify damage levels from bi-temporal multispectral satellite images. The objectives of this research are to :

- Prepare a labelled dataset from unlabeled bi-temporal multispectral satellite images of recently damaged vegetation areas.
- Construct two distinct CNN models to classify the dataset into damaged and undamaged classes and undamaged, mildly damaged and severely damaged classes.
- Estimate the amount of each class of damage.
- Help mitigate the effects of forest fires and make reforestation swifter.

1.4 Thesis Outline

This report aims to provide detailed steps that we have taken in constructing a labelled dataset of recent fire affected vegetation areas from unlabelled Sentinel-2 images. This dataset is then used to classify damaged vegetation using the two models. The paper is organized to highlight the steps we have taken for performing the research.

The Introduction(Chapter 1) section describes the motivation behind the research and addresses the issues in the problem statement. It also highlights the aims and objectives we had for doing the research and describes how the paper is organized with a summarized workflow diagram.

Background Information(Chapter 2) provides insight on previous works performed by researchers on related fields and some necessary background knowledge on machine learning, CNN, ResNet-50, K-means and Sentinel-2 satellite which are implemented in this research.

Data collection and Preprocessing(Chapter 3) outlines the detailed steps we have taken to collect, process and construct the dataset that we use to train and test our

models. It also provides visual representations of the data based on the locations from where they were collected.

Model and Evaluation(Chapter 4) describes the architecture of the two models implemented and their corresponding results and analysis are given to highlight and compare the performance of the models.

Finally, the report is concluded in the Conclusion and Future Work(Chapter 5) section which summarises the report and shows possible future implementations our research can have.

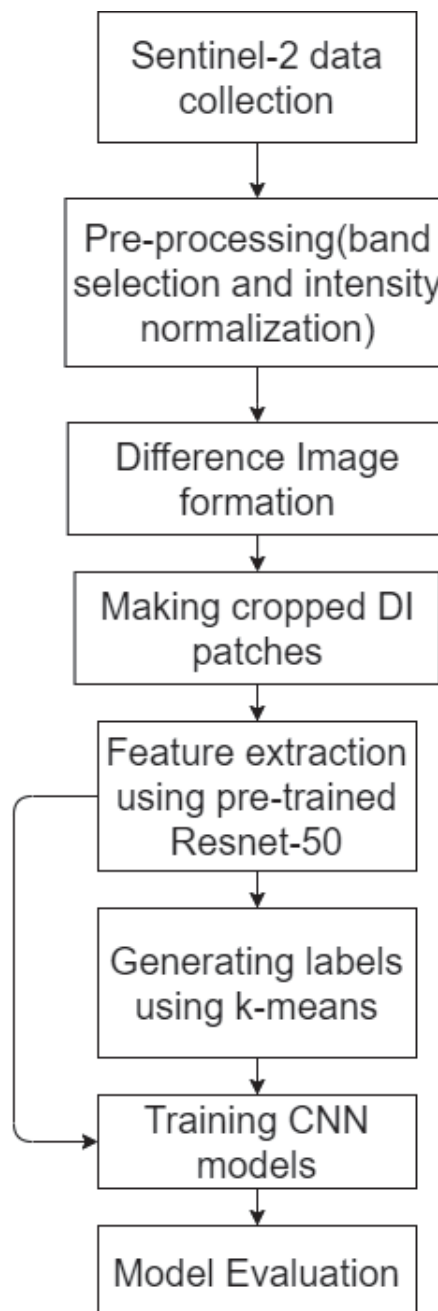


Figure 1.1: Workflow Diagram

Chapter 2

Background Information

2.1 Related Work

The main purpose of our research is to classify the temporal changes in vegetation caused by damage due to fire and therefore it can be considered a change detection problem where we used a neural network approach by using CNN for the classification. Change detection is a very large field in image classification problems and there are many research papers that have previously used a neural network approach for change detection in images.

The research paper [3], proposed a ternary change detection framework using sparse autoencoder(SAE), fuzzy c-means(FCM) clustering and CNN to classify each pixel in their log ratio difference image(DI) into either unchanged, positive change or negative change classes. The SAE was used to suppress noise by learning abstract representation of the data and extract change feature space from the DI which were then used to generate pseudo labels using FCM clustering. The accurately labelled pseudo labels were found by comparing the labels of the neighboring pixels and then used to train CNN to learn higher-level abstract features and to reclassify the pixels into the three categories. Five datasets of SAR images were used for the experiment. The authors stated that their sliding window for SAE training and the parameter controlling the sample selection were tested to find optimum values for the best performance of their proposed method. They later compared the accuracy with Double-Thresholding KI algorithm(DKTI), FCM, Fuzzy local information c-means (FLICM) and kernel weighted FLICM(KWFLICM) using four criteria, overall error(OE), percentage correct classification(PCC), kappa coefficient(KC) and normalized mutual information(NMI).The best results were obtained for Xi'an dataset where the proposed model performed better than the rival algorithms with OE, PCC, KC and NMI values of 1050, 0.9769, 0.8756 and 0.7462 respectively.

In this paper [4], the authors explore various change detection techniques in remote sensing over multi-temporal images. Post Classification (Delta Classification) method produces thematic maps after rectification of classified images . After that it compares corresponding labels to detect changed regions. Since the data from the two times are independently classified, this method allows minimum normalization of atmospheric and sensor differences between the two dates. Image Ratioing calculates the ratio of the DN values of two images obtained at two different times and

then compares the data in a pixel wise manner. If a change exists in the region, then the ratio would be greater or less than 1 depending on the change occurred. If the ratio is exactly 1, then no change has occurred. This method faces difficulty in selecting the threshold value between change and no change. In Image Differencing a pixel and band wise subtraction of the DN values of two images of the same place obtained at different times is performed. If no change occurs, the output is 0. If change occurs, the output is either positive or negative. This method also faces difficulty in selecting the threshold value between change and no change. Standard deviation is used in selecting the threshold value. In Principal Component Analysis (PCA) method, linear transformation is used to reduce the spectral components of multispectral images to a lower number of principal components. This is because these components cause the occurrence of most of the variance in the original image. These components are compared for different dates. Direct Multidate Classification method combines multi-data images into one single data set. Then either a supervised or unsupervised classification is carried out on the data. In both cases, change classes have different statistics compared to no change classes. Multiple classes are generated which display the change in spectra within the single image and also display temporal changes. Change Vector Analysis (CVA) uses a vector called a spectral change vector to represent the change in position of a pixel in spectral space if change occurs over time. This vector represents both the direction (determines nature of change) and magnitude of the change (determines level of change). This method also faces challenges in determining the threshold of magnitude. The authors further describe the types of study areas which are suitable for each technique and also briefly lists down the preprocessing methods that are carried out before any of these techniques.

In [5] and [6], a systemic survey of modern change detection algorithms is carried out which primarily focuses on providing detailed information on preprocessing methods and the common processing steps needed for different change detection algorithms on raw images. In change detection there are both important changes and unimportant changes and the preprocessing steps need to be able to effectively filter out the unimportant changes. These steps include geometric adjustments and radiometric adjustments. Geometric adjustments make sure the images are aligned in the same coordinate frame and changes due to camera motion are negated. These adjustments can be achieved by applying similarity, affine or projective transformations, which are all low-dimensional spatial transformations. The appropriate spatial transformation needed to be used depends on where the change detection algorithm is being applied and therefore can vary. Radiometric adjustments are necessary to deal with changes in illumination of the images either due to changes in the light source or calibration in the instrument. These adjustments include intensity normalization, homomorphic filtering, illumination modeling, linear transformations of intensity, sudden changes in illumination and speckle noise. Intensity normalization makes sure both images have the same mean and variance. Homomorphic filtering models the image intensity at a pixel as the product of illumination from the light sources and the reflectance of the object surface and then logarithms are used to filter out the two components of the intensity signal. In Illumination modelling a piecewise polynomial function can be made for illumination components between images and later PCA can be applied to extract sets of bias for views under all

possible lighting conditions. Linear transformations of intensity transform the multispectral image to a different intensity space before applying change detection by a decision threshold over multispectral bands to discriminate change using neural networks. Sudden changes in illumination can be removed using the known position of the sun to remove strong shadows of objects in the image prior to change detection. Finally speckle noise deals with noise in coherent imagery such as ultrasound and radar images like SAR. These noises are removed in various ways which include frame-averaging, thresholding and local spatial averaging techniques. The researchers also discussed the change mask consistency and the performance evaluation comparison of the change detection models. Change mask consistency is usually obtained by using standard binary image processing operations to remove small label groups that differ from neighboring ones or morphological operations to smooth boundaries. The performance evaluation can be done using either a visual or a quantitative approach. Flicker animation is used to play image pairs in rapid succession in visual approaches which causes the changed regions to flicker. Quantitative evaluation needs the availability or establishment of ground truth which can be tedious and time consuming. After ground truths are obtained percentage correct classification(PCC), Jaccard coefficient(JC) and Yule coefficient(YC) are calculated using the true positives(TP), true negatives(TN), false positives(FP) and false negatives(FN) obtained from ground truth.

We have used ResNet-50 following the previous performances of this architecture over the other models. In [7] they used pretrained ResNet-50 to analyze urban patterns in different ways and found best results when ResNet-50 was tuned to the new dataset before predictions. In another paper [8], a comparative analysis is carried out between ResNet50, VGG19, GoogLeNet and Alexnet in terms of accuracy in classification by feature extraction. The researchers used three datasets SAT4, SAT6 and UCMD. SAT4 is made of 500,000 patches of normalized 28 x 28 pixels of which 100,000 are used for testing. SAT6 consists of 405,000 patches of the same size and 81,000 of these are used for training. UCMD has 100 images of 256 x 256 pixels and a total of 21 classes. The datasets are converted to grayscale images and fed into the pretrained feature extraction models for training and afterwards the remaining part of the datasets are used for the testing phase. Results were calculated from the outputs extracted from the fc8 layer of Alexnet, VGG-19 and googlenet and for ResNet50 it was extracted from the fc1000 layer. The results show that ResNet50 performs better compared to the other models with accuracy values of 98%, 95.8% and 94.1% for the UCMD, SAT4 and SAT6 respectively.

In paper [9] uses Caffe framework and AlexNet Network for CNN to detect areas which are damaged by flood or landslides by feature extraction and obtained accuracy of 0.95 for 10,000 iterations and 0.97 for 250,000 iterations. They used Google Earth Aerial Imagery to collect pre-disaster and post disaster(flood or landslide) aerial imagery of Japan and Thailand. Their training set consisted of 18 sets with 2200 patches/set of landslide images and 9 sets with 2200 patches/set of flood images. They stacked the RGB pre disaster and post disaster on top of one another to create the six channel input training and test image section instead of only focusing on post disaster image for learning. After that, they compared their merged training set with ground truth patches and labelled them for training CNN to learn

damaged features. Testing set consisted of 17 sets with 2200 patches/set of landslide images and 13 sets with 2200 patches/set of flood images. They used the ground truth images to cross-check the accuracy and also used precision, recall and f-score as parameters to calculate accuracy.

The research paper[10] classifies fire prone forest areas by using Convolutional Neural Network (CNN) . They processed multispectral satellite imagery of “Camp Fire” struck forest of California, USA from Landsat 8. They analyzed their results based on the spectral indices: NDVI, NDWI and PSRI which were determined by the parameters : green vegetation of the area, humidity and dry carbon. Using the coordinates of the “Camp Fire” area, the input images were selected. From each image, three indices were calculated using index equations which use each pixel’s index value depending on the combination of brightness values of specific bands or channels. These three indices as RGB channels(red responsible for PSRI, green responsible for NDVI and blue responsible for NDWI) were merged after calculation. This merged image was the input image to their convolutional neural network. They used 40,000 32x32 pixel input images for the research, from which 20,000 were images of fire prone areas and the remaining 20,000 were of not fire prone areas. The training set used 28,000 images, validation set used 6,000 images and test set used 6,000 images. The Tensorflow 1.6 and Keras 2.2.4 frameworks were used. The convolutional neural network was trained during 50 epochs and the training time was one hour. The accuracy of this research on the test set was 94.27%.

The main objective of research paper [11], is to compare the performances of different deep learning architectures on three different hyperspectral image datasets, Pavia, Indian Pines, and Data Fusion Contest 2018(DFC2018). Before using the hyperspectral data, they discarded outlier bands, normalized the data and reduced dimensionality using several well known methods such as NDVI and NDWI. The architectures applied over the three datasets were SVM, 1D NN, 1D CNN, RNN, 2D+1D CNN and finally 3D CNN. The results obtained show that 1D NN provides the best results of 83.13 ± 0.84 accuracy and a kappa index of 0.807 ± 0.009 over the Indian Pines dataset. For the Pavia dataset and DFC2018, 3D CNN provides the best results of 84.32 ± 0.72 and 49.26 accuracy respectively and a kappa index of 0.799 ± 0.009 and 0.46 respectively.

The paper[12] clasified aerial imagery into seven land cover classes which include building, grassland, dense vegetation, waterbody, barren land, road and shadow using Convolutional Neural Network (CNN). The initial aerial photo was first corrected using geometric calibration and normalized to avoid abnormal gradients since CNN uses initial weights between 0-1. The metrics that were used for evaluating the system were overall accuracy (OA), average accuracy (AA), per-class accuracy (PA) and Kappa index (k). This architecture using dropout and batch normalization had OA=0.973, AA=0.965 and k=0.967. The minimum accuracy was 0.894 for shadow class. It was discovered during confusion metric evaluation that several samples were misclassified which affected the PA of those classes. They also carried out the experiment without drop out and batch normalization and concluded that the system works more accurately with dropout and batch normalization and improves classification by 4%. Furthermore, they discussed the impacts of parameters such

as number of convolutional filters, activation function, drop probability, optimizer, batch size and patch size on the performance of the CNN and the time taken for the training.

In [13], the researchers built the unsupervised deep learning classification framework which uses Deep Belief Network(DBN) for classification of satellite images. Their research is based on the two labeled National Agriculture Imagery Program(NAIP) datasets, SAT-4 and SAT-6. The images are made of red, green, blue and NIR bands. The best performance for SAT-4 was an accuracy of 81.78% and 76.47% for SAT-6. They also performed Convolutional Neural Network(CNN) and Stacked Denoising Autoencoder(SDAE) on both datasets and obtained accuracy of 86.827% for CNN and 78.43% for SDAE on SAT-4 and 79.063% for CNN and 78.43% for SDAE on SAT-6. They then used their own framework which extracts features from satellite images and then feeds the data into DBN after normalizing the data into feature vectors. The features used were Enhanced Vegetation Index(EVI), Normalized Difference Vegetation Index(NDVI), and Atmospherically Resistant Vegetation Index(ARVI). The results obtained from the classification were claimed to have accuracy of 97.946% on SAT-4 and 93.916% on SAT-6. They also stated that their classifier performed 11% and 15% better on SAT-4, compared to CNN and SDAE, and on SAT-6 respectively. They also carried out Random Forest classifier with an accuracy of 69% on SAT-4 and 54% on SAT-6 to show that unsupervised learning is better for satellite image classification. The researchers also justified their feature extraction process by calculating the statistical analysis of their approach using Distribution Separability Criterion.

The main goal of research paper[14] is to classify land cover and crop types into eleven classes using remote sensing images from Landsat-8 and Sentinel-1A satellites over Ukraine. They used four Landsat-8 images and fifteen Sentinel-1 images from 2015. Landsat 8 images are masked with cloud and shadow masks but they had to pre-process the Sentinel-1 images. Six time series spectral bands from every Landsat 8 and two bands with combinations of vertical transmit and vertical receive and combinations of vertical transmit and horizontal receive polarizations from each Sentinel-1 scene was used as an input to the classification system. 547 polygons of various classes were randomly divided into 50% training samples and 50% validation samples. They used a four level architecture, preprocessing which restores missing values in the images. The second level was CNN compared with RF and ENN methods. The supervised CNN used the six bands from four Landsat 8 and two bands from fifteen Sentinel-1 images as input feature vectors. They compared two CNN models, 1-D CNN and 2-D CNN. The accuracy of RN was 88.7%, ENN was 92.7%, 1-D CNN was 92.5% and 2-D CNN was 94.6% .

Another research paper [15] detects damaged forest areas by wind throw using U-Net CNN architecture over multispectral aerial remote sensing data of Bavaria, Germany with 92% accuracy over test dataset. A digital mapping camera was used to capture the aerial photographs which consisted of blue, green, red and NIR and 20-cm spatial resolution. Of these, 12 labeled orthophotos containing damaged and non forest areas along with other objects such as buildings, roads or meadows were selected for this research. They divided each of their 10,000x10,000 pixels images into 1525

tiles of 256x256 pixels of which 80% was training set and 20% was validation set. They used 2 full 10,000x10,000 pixels photo for testing. The encoder part of U-Net reduces the spatial dimension of the feature maps and learns to extract important features whereas the decoder expands the spatial dimension of the feature maps and learns to recover the full information. After optimization of hyperparameters, the performance of this system was calculated using the intersection over union metric and the accuracy metric. A custom metric algorithm was used for evaluating a threshold which was used to compute the performance metrics. However, There were a few labeling errors in reference data which made the algorithm's accuracy slightly skewed since it correctly detected damaged areas better than humans and the parts it missed were due to shadows in aerial imagery

2.2 Machine Learning

Machine learning(ML) is a subclass of Artificial Intelligence (AI) whose main objective is to make machines learn the surroundings automatically without the intervention of humans or any definite written programs. Learning requires the machine to alter it's actions based on the patterns in the data that we provide. By doing so, it achieves a greater accuracy i.e. it managed to pick the correct actions in more number of instances [16] . Machine learning plays a role in various fields of computing and statistics. ML model simulation is used in Computational Statistics to predict outcomes of dataset instances using computers. Real world problems such as Image Processing, Classification, Regression, Remote Sensing, Clustering etc. involve different sectors of ML.

2.2.1 Supervised vs Unsupervised Learning

Supervised and Unsupervised Learning are subcategories of Machine Learning. According to [17], in supervised learning, “experience” can be achieved by training the machine over a training dataset which consist of data instances and corresponding labels and “expertise” can be achieved by correctly predicting maximum number of labels from an unseen and unlabeled test dataset. Hence, significant information i.e. labels are provided to the machine to learn a classifier which can later predict labels from unseen data and evaluate the accuracy of its predictions by calculating the loss using the labelled input data. Unlike supervised learning, no labels are provided in unsupervised learning which implies that the dataset is not divided into train and test data and therefore the whole input data is fed to the machine whose aim is to perceive existing patterns from the data and use it to summarize, compress or cluster the data. In this research we used an unsupervised clustering algorithm called K-Means algorithm to cluster our flattened dataset of cropped difference images with the goal of obtaining labels for our unlabeled dataset.

2.3 Convolutional Neural Network

An Artificial Neural Network(ANN) is a subclass of ML. Its theory is analogous to the biological neural network. An ANN is a network of neurons(learning units) that processes a signal upon receiving it and based on a threshold, it either passes

the information to other connected neurons or remains inhibitory. It has 3 main layers: the input layer, the hidden layer and the output layer. According to [18], Deep Learning models consist of several processing layers to learn data which contain multiple levels of abstraction by adjusting weights during back propagation. Given a complex dataset, the multiple layers allow the models to learn higher levels of features in consecutive layers using non-linear modules at each layer. A Convolutional Neural Network(CNN or ConvNet) is a category of Deep Neural Network. According to [18], it is a supervised neural network whose speciality is to process multi-channelled multidimensional data, for example a three channelled (RGB) 2D image. The RGB are the red, green and blue channels of the image which carry the pixel values. CNNs are typically used for audio recognition, image classification, object recognition, detection and segmentation and volumetric images. The key properties that make CNNs popular in the field of Deep Neural Networks are : it reduces the number of connections (introduces local connections) , it shares weights on edges, it introduces pooling to reduce complexity and it uses several layers [18]. A CNN architecture for image classification is split into two stages : a series of alternating convolutional layers and pooling layers followed by multiple fully connected or dense layers and an activation function is used on the output layer [10].

2.3.1 Convolutional Layer

Convolutional layers are used to extract features and produce feature maps by convolving the input array with filters or kernels. The authors in [19] state that the units in each convolutional layer are laid out as feature maps. A filter or a kernel is an arrangement of weights which connects every unit in a feature map to a block of neurons in the previous feature map or input image(receptive field) through convolutional operation followed by non-linear activations. In order to retrieve different types of features from different areas of the image, multiple feature maps with different weights are used.

$$Y^k = f(W^k * x)$$

Here, the (*) implies the convolutional operation between the input image x and the k^{th} filter W^k which is simply the dot product between the filter W^k and every equal sized block on the input image. The f represents the non-linear activation function and Y^k represents the k^{th} feature map. Convolutional layers are defined by the hyperparameters : depth, stride and zero padding where depth is the number of kernels which control the depth of the output feature maps, the stride controls the overlapping of the receptive fields and zero padding fills up the borders with 0 values to control the dimension of the output feature maps as stated in [20]. It further states the formula to calculate the dimensions of the output feature maps is:

$$\frac{(V - R) + 2Z}{S + 1}$$

where the input image dimensions are denoted by V (height x width x depth), the kernel dimensions are denoted by R , zero padding is denoted by Z and the stride is denoted by S .

2.3.2 Activation Function

According to [21], an activation function is mainly added to a CNN layer to perform nonlinear operations which help to minimize the vanishing gradient problem. Some notable non-linear activation functions are : ReLU, Leaky ReLU, Tanh and Sigmoid. In this research, we used rectified linear unit (ReLU) activation function in our CNN model. According to [22], in comparison to Sigmoid and Tanh, ReLU performs better in terms of generalizing and this activation function is fast learning too which leads to fast computations. ReLU is a piecewise linear function which maintains the linearity in models making them easily optimizable during gradient descent. Given an input value, ReLU ignores negative values and outputs 0 , otherwise outputs the input value itself and hence introduces sparsity in hidden units.

$$f(x) = \max(0, x) = \begin{cases} 0 & x_i < 0 \\ x_i & x_i > 0 \end{cases}$$

2.3.3 Pooling layer

The role of pooling layers is to minimize the spatial resolution of the feature maps which in turn decreases the number of parameters and hence computational complexity is minimized even further [20]. In this research max pooling layers are used in the CNN model. In max pooling layers, the maximum value in a receptive field of a feature map is selected and passed on to the next layer [19].

2.3.4 Fully Connected Layer

The output from the convolutional layers are flattened and inputted to fully connected layers or dense layers which contain a layer of neurons where each neuron is connected to all the other neurons in both the previous and the next layer the same way they are organized in a classical ANN [20].

2.3.5 Optimizer

In order to attain optimum training result, optimization algorithms(optimizers) are used to reduce the loss function. The hyperparameters of the model determine proper and fruitful training and help to produce results with greater accuracy. Similar to [21] our research also uses Adam optimizer which is a first order optimization algorithm. This optimizer regulates the learning rate of every parameter on the basis of both first-order and second order moment values of every parameter's gradient. This is useful since it maintains the benefits of AdaGrad (adaptive gradient) algorithm and RMSprop (root mean square propagation) algorithm. Adam converges faster and has improved learning compared to other different types of self-adaptive learning rate optimizers.

2.4 ResNet-50

ResNet was built to make the training process of very deep neural networks easier. It has been stated that deeper networks tend to show degraded performance at convergence due to the increasing depth which leads to larger training error and

vanishing gradient problem [23]. The problem is not related to overfitting but to the saturated accuracy at greater depths making optimization difficult. In order to solve the degradation problem, “shortcut connection” or skip connections are added which performs identity mapping without increasing the number of parameters or complexity of the problem. The main goal of the network is to learn the residual function which can approximate the differences between the output and the input. ResNets achieved excellent results on both the ImageNet dataset and CIFAR-10 datasets which proves its robust nature. This network originally takes a 224x224 image as input and classifies it in one of the 1000 classes. ResNet-50 goes 50 layers deep with five convolutional blocks. The second block contains three residual blocks, the third block contains four residual blocks, the fourth block contains six residual blocks and the fifth block contains three residual blocks. Each residual block has three convolutional layers: 1x1 followed by 3x3 and lastly 1x1 convolutional layers. The 3x3 layer in the middle is called the bottleneck layer and the 1x1 layers are used to first lower the dimensions and later to increase the dimensions. ResNet-50 gives 20.74% top-1 error and 5.25% top-5 error on ImageNet dataset.

Layer name	Layer description		
Block-1	Conv(7,64),Maxpool(3,2)		
Block-2	$1 \times 1, 64$ $3 \times 3, 64$ $1 \times 1, 256$	$\times 3$	
Block-3	$1 \times 1, 128$ $3 \times 3, 128$ $1 \times 1, 512$	$\times 4$	
Block-4	$1 \times 1, 256$ $3 \times 3, 256$ $1 \times 1, 1024$	$\times 6$	
Block-5	$1 \times 1, 512$ $3 \times 3, 512$ $1 \times 1, 2048$	$\times 3$	

Table 2.1: ResNet-50 Architecture [23]

2.5 Kmeans

K-Means is an unsupervised clustering algorithm that groups similar data points into one cluster. Each cluster is assigned a centroid (this process can be random) which is the mean of the cluster and each datapoint is assigned to a cluster depending on the minimum distance or difference from it i.e. each datapoint will be put into the closest cluster [24]. If the distance is assumed to be Euclidean distance , then according to [25]

$$C_i = \frac{1}{M} \sum_{j=1}^M x_j$$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

After assigning the data points to closest clusters, the means are recalculated. [25] further adds that, the objective function that K-Means uses is :

$$a_{ij} \begin{cases} 1 & d = \min \{D(x_i, c_i)\} \\ 0 & \text{otherwise} \end{cases}$$

where D is the minimum distance between a datapoint and the cluster it belongs to. This process continues until the centroid values no longer change. Elbow Method is used to discover the best cluster number K from a dataset. The process is simply to calculate the sum of square error (SSE) which is the distance of each data value from it's own cluster centroid and by analyzing the result to determine the sharp decrease in the SSE which indicates the "elbow" and thus best compact clusters are expected from the corresponding K value [25] [24].

$$J = \sum_{i=1}^n \sum_{l=1}^k a_{il} D(x_l^{(i)}, c_i)^2$$

In this research, we used the K-Means algorithm to produce clusters from our flattened DI image crops based on the elbow method. The cluster labels predicted by the algorithm turned our unlabeled dataset into a labelled dataset ready for supervised learning.

2.6 Sentinel-2 Satellite

Sentinel-2 Satellites are Earth observation twin satellites (Sentinel-2A and Sentinel-2B) that were launched as a mission of the European Commission's Copernicus program. According to [26], each of the satellites are equipped with a multi-spectral instrument (MSI) that consists of 13 spectral channels in the visible near infrared (VNIR) and short wave infrared spectral range (SWIR) zones. The sensors capture information at 10m, 20m and 60m spatial resolution which enable the satellites to produce imagery useful for agriculture, land cover change, forest and vegetation monitoring, inland water, glaciers and flood monitoring and coastal regions mapping. The two satellites move on the same orbit at an 180° phase difference and each of them revisits the same location in 10 days which ensures data collection from the same location in 5 days at the equator and 2-3 days at middle latitudes by the pair. The band information for Sentinel-2 Satellite is given in the table 2.2.

Sentinel-2 Bands	Central Wavelength (μm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water vapour	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

Table 2.2: Sentinel-2 Band Information[27]

Chapter 3

Data Collection and Preprocessing

3.1 Image collection

Our objective was to collect high quality multispectral satellite images of very recent fire affected vegetation areas from freely available data resources. Multispectral satellite images are obtained from Earth observing satellites that use sensors to obtain information from the reflectance of landscapes. Different objects on the surfaces of the Earth reflect different ranges of wavelengths called bands from the electromagnetic spectrum. Each pixel of an image consists of information from multiple bands that are measured by satellite sensors. For collecting the images, we chose Copernicus Open Access Hub [28] which provides user products from Sentinel-1, Sentinel-2, Sentinel-3 and Sentinel-5P which are readily available to the public through an user account on the website. All data used for this research is from Sentinel-2. User products from Sentinel-2 Satellites are available from August of 2015 to the current date. The Sentinel-2 products contain the Multispectral Satellite Images and also all the necessary sensing information related to the images. We chose 12 forest areas which are damaged due to large forest fires and two images of the same geometrical location, before and after the destruction, are downloaded from Copernicus Open Access Hub. The Sentinel-2 images consist of 3 spatial resolutions : 10m, 20m and 60m. For the purpose of this research, we chose the maximum resolution i.e., 10m in order to capture maximum spatial details and also because it contains the Red(R), Green(G) , Blue (B) and Near Infrared (NIR) bands. Each image is 10980x10980 pixels representing an area of 100x100 km² and does not contain any labels. Images with least cloud coverage, water bodies and random clippings due to the rotation of the satellites are chosen in order to obtain maximum comprehensible information from the images. The images are also pre-processed by scene classification and atmospheric correction [29]. The images have unique footprints which are related to the geographical location of the regions. Geographic corrections are not required on the images since we obtained both the before and after images having the same footprint, i.e. both the images have the same spatial coordinates.

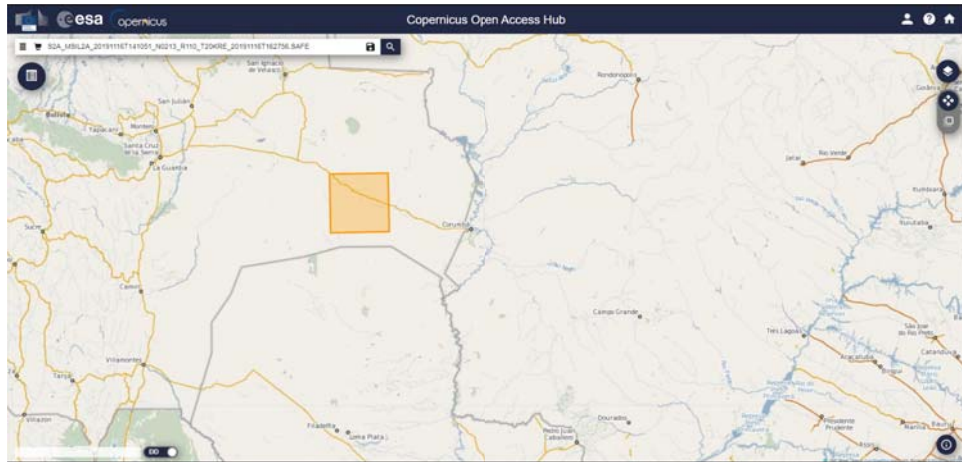


Figure 3.1: Geographical location of Bolivia

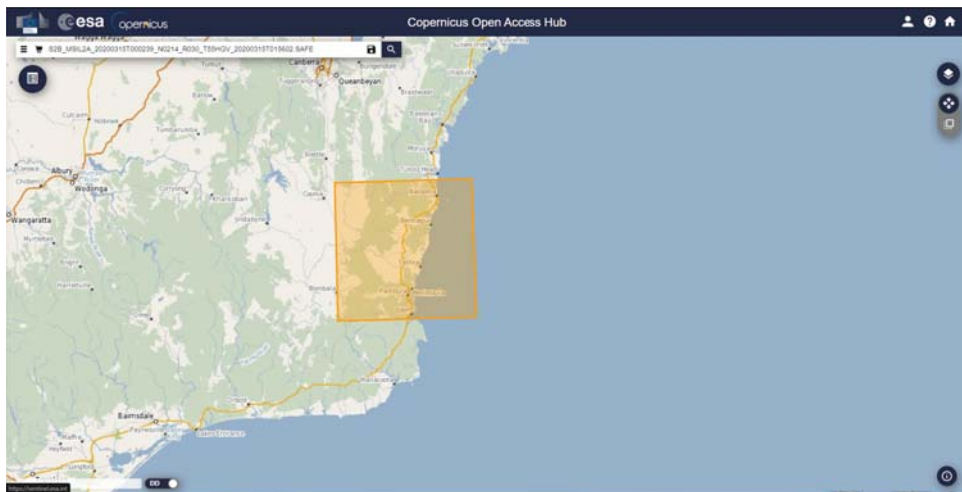


Figure 3.2: Geographical location of Bombala

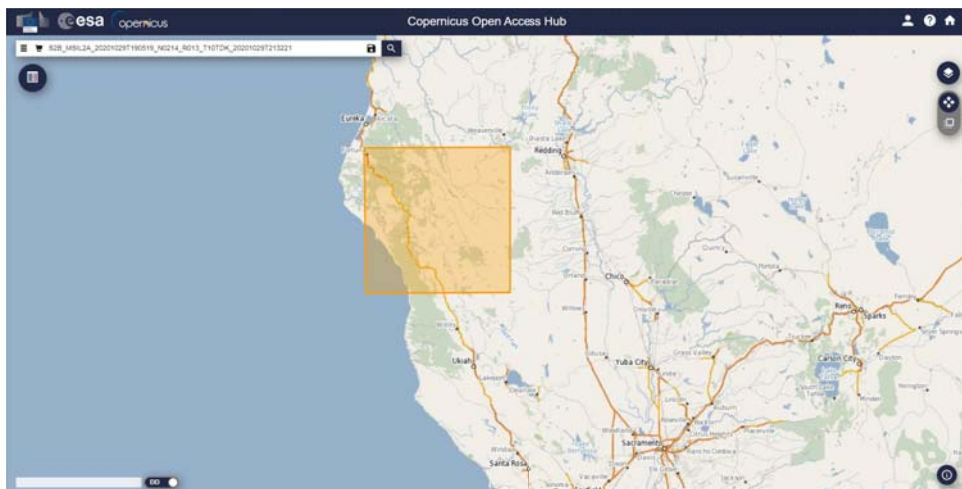


Figure 3.3: Geographical location of California

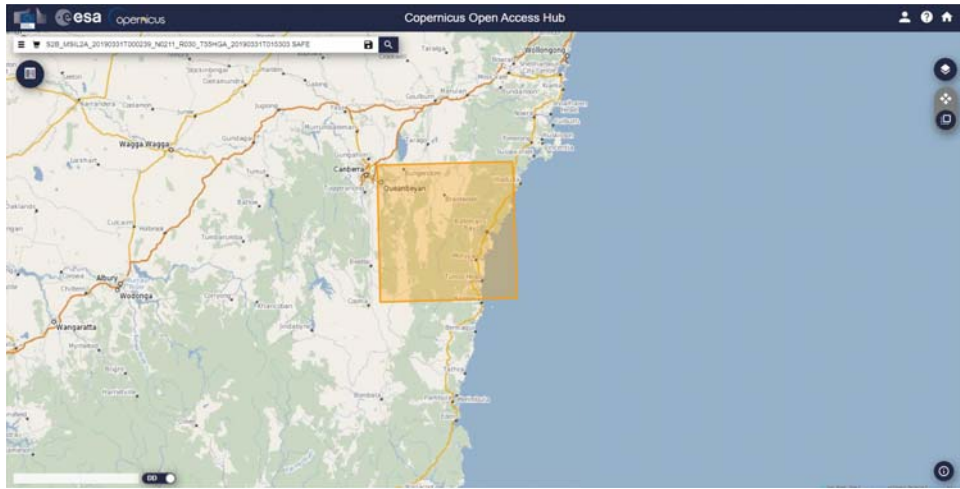


Figure 3.4: Geographical location of Canberra

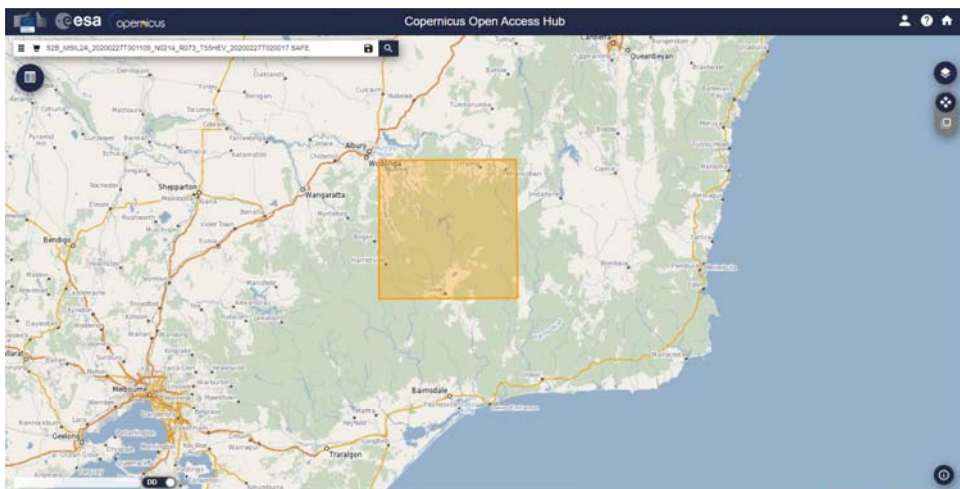


Figure 3.5: Geographical location of Harrietville

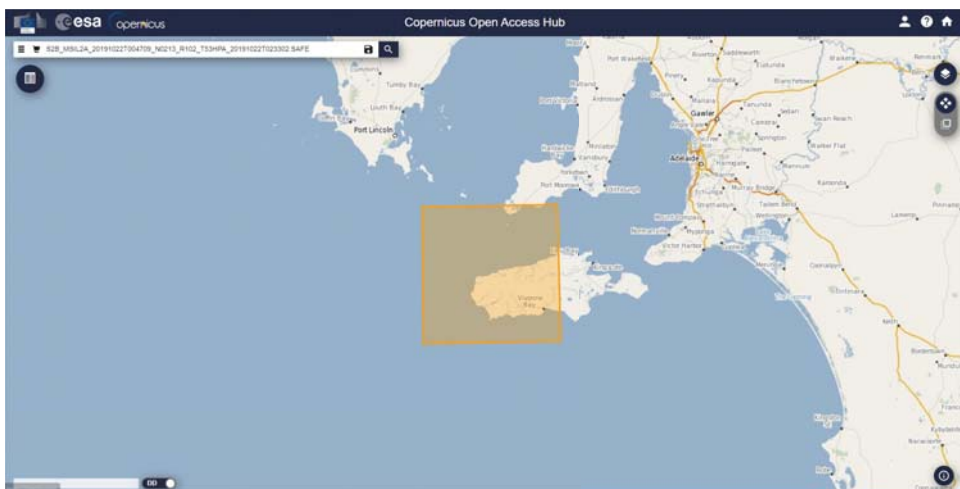


Figure 3.6: Geographical location of Kangaroo Island left side

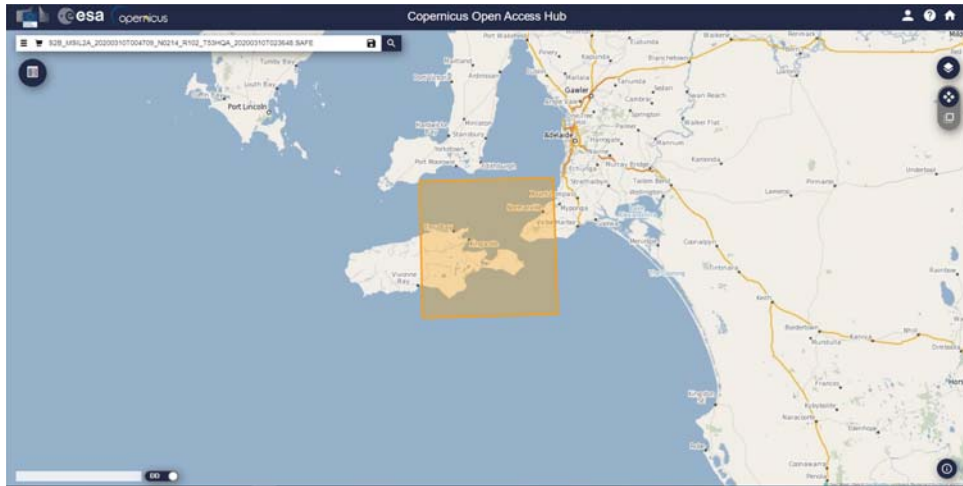


Figure 3.7: Geographical location of Kangaroo Island right side

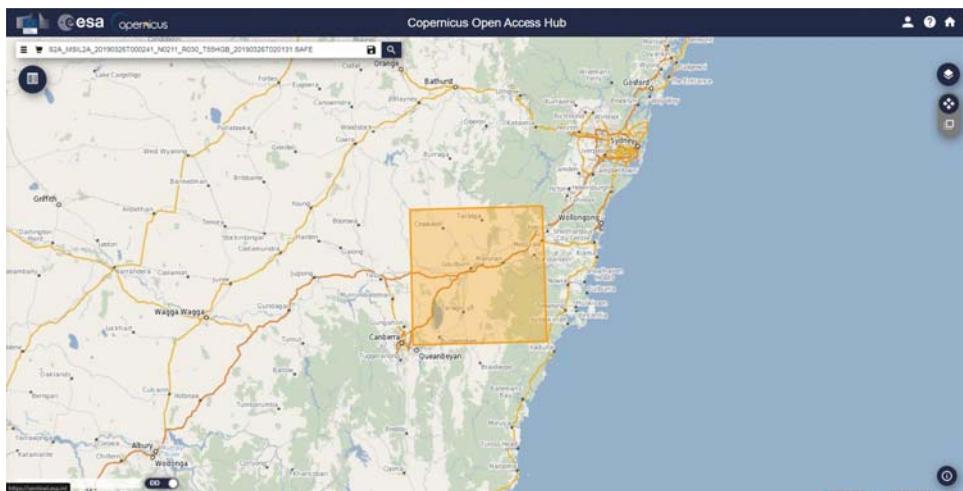


Figure 3.8: Geographical location of New South Wales

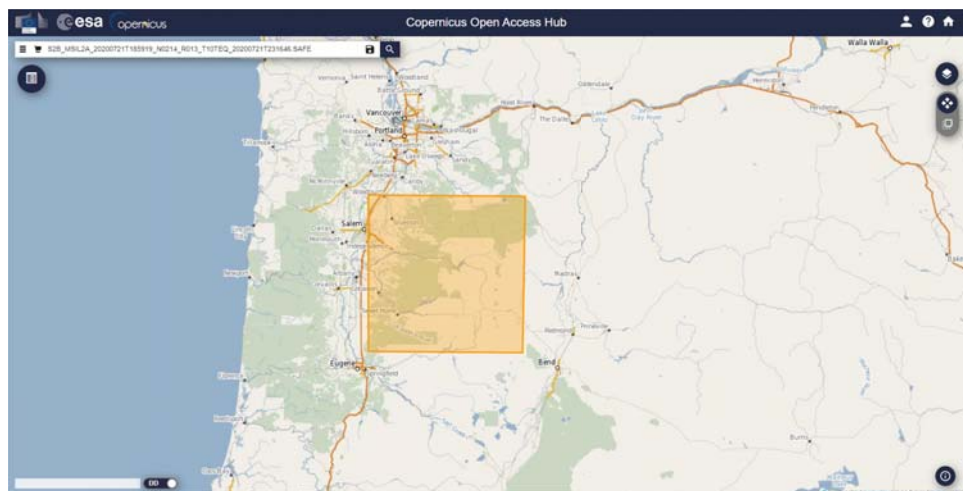


Figure 3.9: Geographical location of Salem

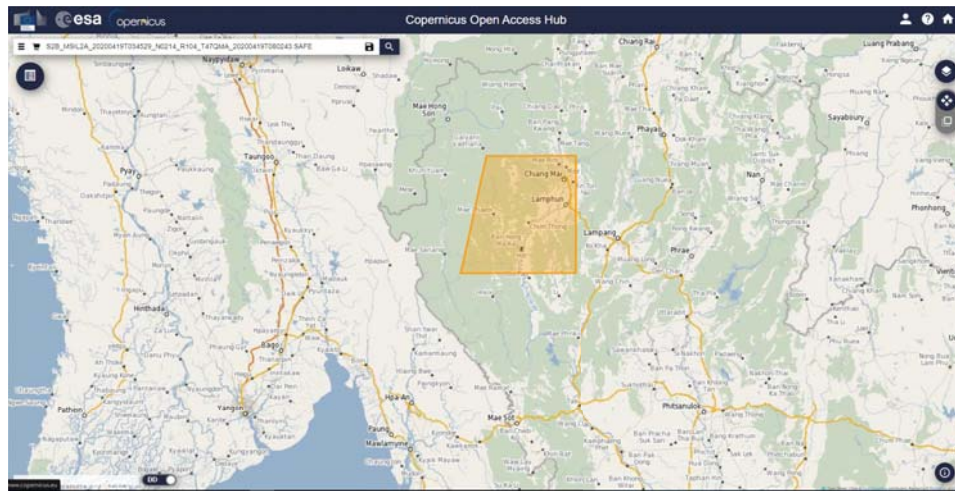


Figure 3.10: Geographical location of Chiang Mai

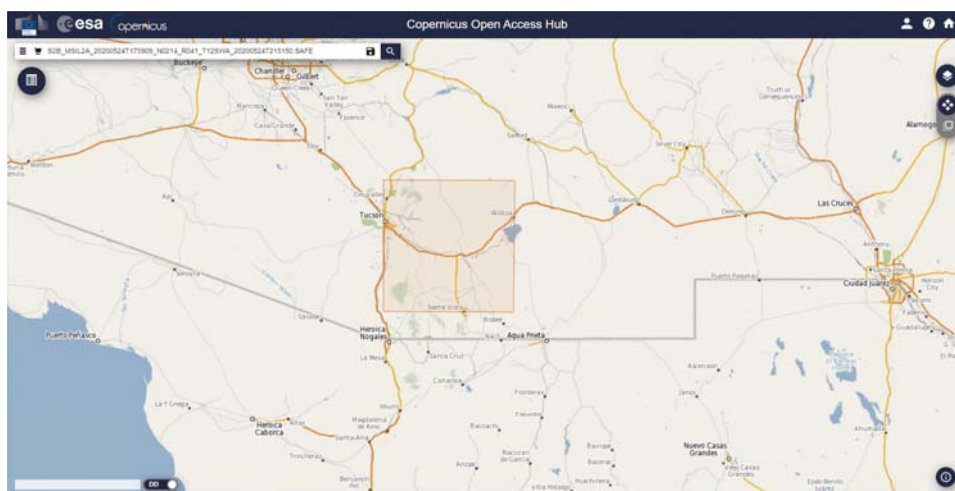


Figure 3.11: Geographical location of Tucson

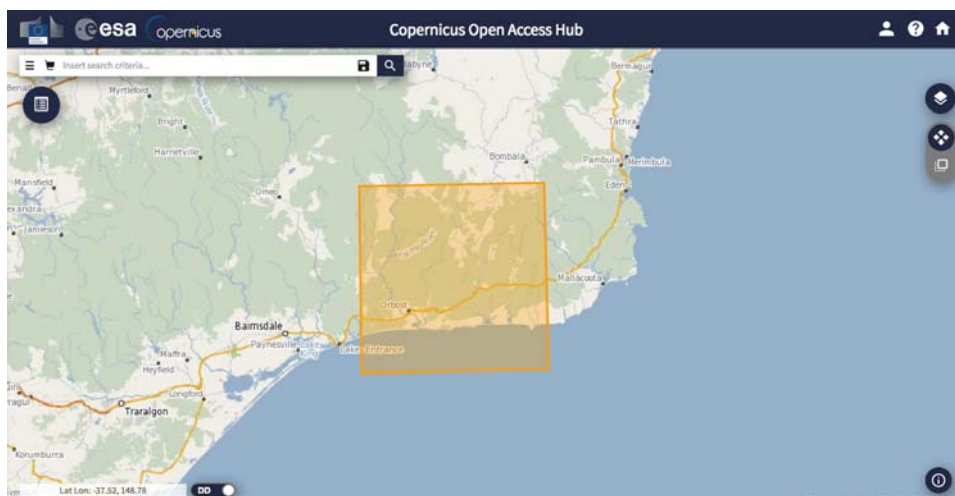


Figure 3.12: Geographical location of Victoria

3.2 Band Selection

In order to obtain optimal and accurate information of damages made to vegetation and trees in a forest by fires, bands which emphasize the abundance or lack of vegetation in a particular region distinctly are selected. Red and green bands play a major role in indicating vegetation. Near-Infrared bands reflect chlorophyll well and thus, damaged vegetation will have comparatively low reflectance than healthy vegetation. Blue bands are seldom used for detecting greenery from healthy plants since they do not contribute to the chlorophyll reflectance. Therefore, a three channeled (three dimensional) image is created, consisting of Green (G), Red (R), and Near-Infrared (NIR) bands in the Tiff format. This combination of three bands display a red or pink color to indicate greater chlorophyll reflectance i.e. healthy vegetation [27]

3.3 Band scaling

Each of the before and after images, from the 12 pairs of images, are scaled using intensity normalization method [5]. Each pixel value of all the bands of both the before and after images are normalized such that the mean of each band is approximately zero and variance is one. This is done by subtracting the mean of each band from the pixels of that band and by dividing the result by the standard deviation of that band. For B_{ij}^k , ij represents the coordinate of a pixel and k is the band number of any band B . The μ^k and σ^k represent the mean and standard deviation of k^{th} band.

$$B_{ij}^k = (B_{ij} - \mu^k) / \sigma^k$$

This ensures better comparison between the bands of the before and after images since all the bands will have approximately equal scales and units. As stated by [4], a difference image, DI, is created by Image Differencing method which is the pixel wise subtraction of the scaled bands of the after image, Xa_{ij}^k , from the corresponding scaled bands of the before image, Xb_{ij}^k . The scaled G, R and NIR bands are stacked on top of one another with G written as band-1, R written as band-2 and NIR written as band-3.

$$DI = Xb_{ij}^k - Xa_{ij}^k$$

This DI captures the pixel wise change values which represent either unchanged (zero change) or a positive or negative change in vegetation. This will help in feature extraction based on changed pixel values by the convolutional neural network. The reason for this is that the DI will solely contain values which represent only temporal changes that occurred and will omit values which remained constant over the same geographical location. Hence, the land will have the following types of patches: change due to fire damage, change due to growth of new vegetation, seasonal loss of vegetation, land surface etc, construction of new buildings nearby or changes on nearby roads or housing plots and unchanged land. Each DI is a three dimensional “change image” with a size of 10980 x 10980 pixels. Using this method, we obtained 12 DI from the 12 pairs of before and after images. After visualizing NIR, R, G bands of each DI and comparing each DI with the corresponding original before and after images, the variations on the surfaces were visually detectable. In almost all cases, three distinct colors were noticeable : red/pink, teal, gray/ashy green.

Careful observation of the comparison of each DI with the before and after image pairs reveal that the red/pink patches of land represent loss of vegetation, the teal patches represent either new vegetation or road/housing construction and ashy green patches represent unchanged land. The red/pink color representation of damage is obvious since NIR, R, G bands have greater chlorophyll reflectance and hence, if after images contain loss of vegetation, then subtracting after image pixels from before image pixels is supposed to result in values consistent with abundant vegetation resulting in red/pink color.

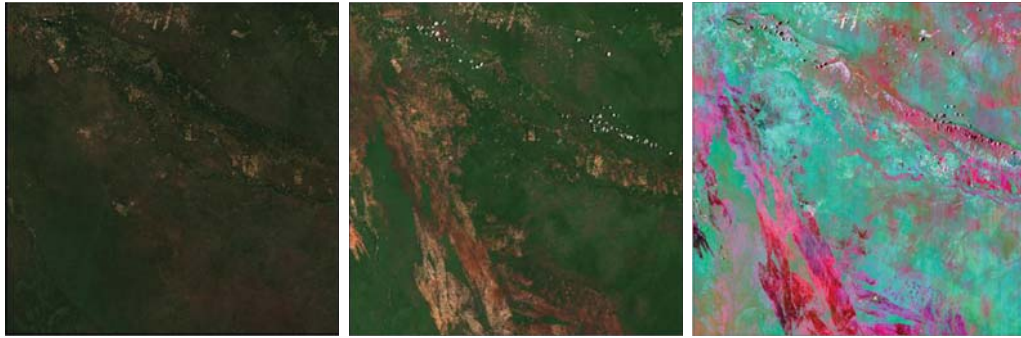


Figure 3.13: Before image, After image and Difference image of Bolivia



Figure 3.14: Before image, After image and Difference image of Bombala

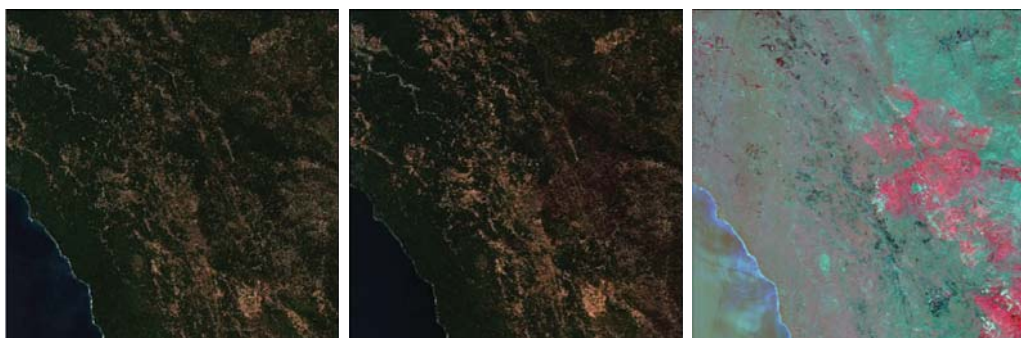


Figure 3.15: Before image, After image and Difference image of California



Figure 3.16: Before image, After image and Difference image of Canberra



Figure 3.17: Before image, After image and Difference image of Harrietville



Figure 3.18: Before image, After image and Difference image of Kangaroo Island left side



Figure 3.19: Before image, After image and Difference image of Kangaroo Island right side

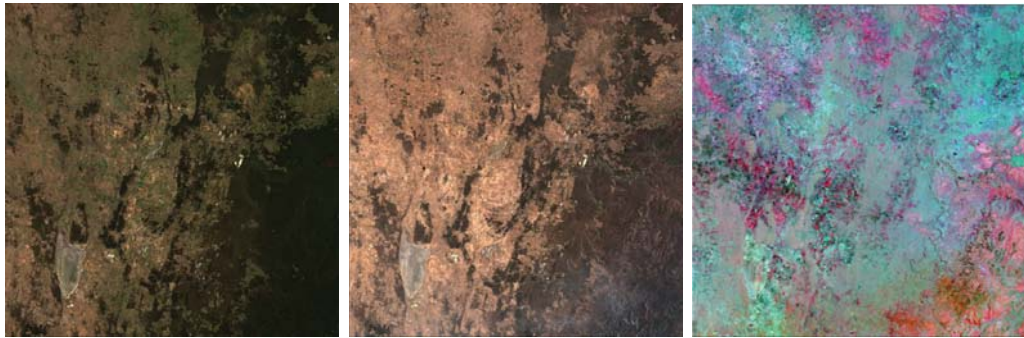


Figure 3.20: Before image, After image and Difference image of New South Wales



Figure 3.21: Before image, After image and Difference image of Salem



Figure 3.22: Before image, After image and Difference image of Chiang Mai



Figure 3.23: Before image, After image and Difference image of Tucson

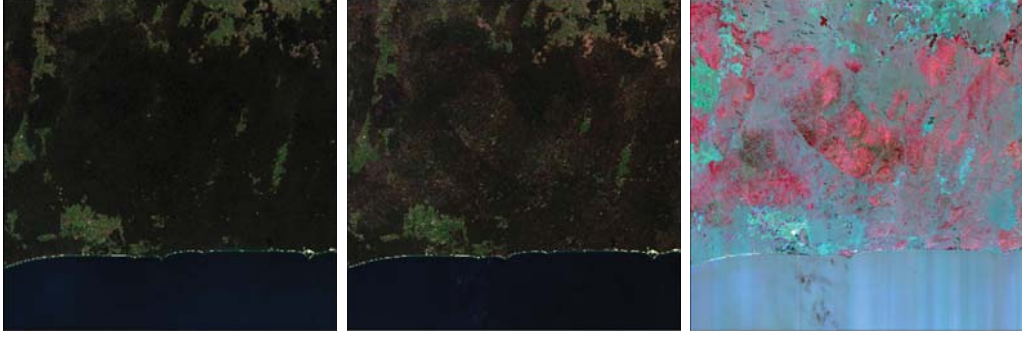


Figure 3.24: Before image, After image and Difference image of Victoria

3.4 Making Image Crops

The 12 DI obtained are unlabelled and large in size. The large size will increase the training time and will put huge computational load in the first few layers of the Convolutional Neural Network. Also, 12 DI is insufficient to train a Convolutional Neural Network since it might not learn important features and thus might produce poor results. To overcome these problems, each DI is cropped into approximately 8100 smaller three dimensional image crops of 122x122 pixels which means each image crop represents approximately 1.235 km². A few images contained water bodies and random clippings due to the rotation of the satellites which not only did not carry any valuable information for this research but also had the potential to interfere with the learning process of the Convolutional Neural Network by introducing irrelevant features. Thus, regular rows or columns of image crops which contained problematic sections were cropped out of the difference images which contained them and not included in the dataset. A total of 85,860 image crops of 122x122x3 pixels were obtained. These image crops are essentially the DI clipped into smaller 122 x

Name	Before date	After date	Number of image crops
Bolivia	09/07/2019	16/11/2019	8100
Bombala	30/04/2019	15/03/2020	6030
California	25/09/2019	29/10/2020	5850
Canberra	31/03/2019	14/04/2020	8100
Chiang Mai	10/01/2020	19/04/2020	8100
Harrietville	13/01/2019	27/02/2020	8100
Kangaroo Island left	20/10/2019	09/02/2020	5130
kangaroo Island right	23/08/2019	10/03/2020	8100
Salem	21/07/2020	29/09/2020	5940
Tucson	24/05/2020	11/10/2020	8100
Victoria	30/04/2019	11/09/2020	6210
New South Wales	26/03/2019	04/02/2020	8100

Table 3.1: Details of Image Crops

122 sized crops. Therefore, these are still unlabelled but the only difference is they are in larger quantities. Labelling these 85,860 image crops manually is inefficient

and will lead to a large number of inaccurate labellings. To avoid this, an unsupervised clustering algorithm called K-Means is used to cluster the image crops with similar pixel values. In order to improve the clustering by K-Means an important step of extracting features is included before it. Using transfer learning a pretrained ResNet-50 is used to extract feature maps from each 122x122x3 image crop of each DI. The one dimensional flattened outputs from ResNet-50 are later inputted into the K-Means to cluster similar flattened outputs based on similar features.

3.5 Transfer Learning Using ResNet-50

Transfer Learning is using a model already pretrained on another dataset. The pretrained model has the correct weights stored which the model learned when it was trained on the previous dataset. We have used transfer learning because it not only saves time needed for training a new model from scratch but also yields better performance. The reason for using ResNet-50 is the ability of the network to learn complex features robustly and also the flexibility of the code structure and it also provided consistent and better results when compared to other architectures in previous researches [7] and [8]. It can achieve satisfactory flattened feature maps without training the model which is time efficient and also reduces the use of hardware resources. Therefore, using ResNet-50 ensures adequate flattened feature map extraction which is necessary for accurately producing labels from K-means algorithm.

In this research, we modified the code in order to input our 122x122x3 size images and also to obtain flattened feature maps as outputs instead of object classes. The size of our images is also the reason for not being able to use outputs from the fully connected layers because the image size of original ResNet-50 is different from ours. However, [[30]] used different classifiers on feature maps taken from different layers of different neural networks and the features extracted from last layer of ResNet-50 provided satisfactory results when inputted into other classifiers. The flatten layer is added to the second last layer which is the addition of the output of residual block-2 of 5th convolutional block and the batch-normalized output of the third convolutional layer of residual block-3 in the fifth convolutional block. The second last layer has output shape of 4x4x2048 and therefore, the flattened output shape for 8100 crops (a single DI) is (8100,32768).

3.6 K-means

We used the flattened DI crops from ResNet-50 as input to the K-Means clustering algorithm in order to obtain labels for our upcoming phase of supervised classification. K-Means algorithm is simple to implement and it did an overall satisfactory job in clustering the flattened feature maps. The flattened outputs of each DI have the shape $(N^k, 32768)$, where N represents the total number of crops of the k^{th} DI. The aim of K-Means algorithm is to cluster similar flattened crops produced by ResNet-50. The flattened crop images obtained from the last convolutional layer of ResNet-50 contain characteristics of the change in forest areas in the form of feature maps and thus, flattened crops which are homogenous will be grouped in

the same cluster by K-Means. The crops grouped in the same cluster will have the same labels i.e. the labels predicted by K-Means will become the labels for the corresponding flattened crop images to complete our dataset for the next phase of supervised learning. Our goal is to build two models: one for detecting only undamaged and damaged land from satellite images and another for classifying levels of damage within the damage areas and hence outputting undamaged, mildly damaged and severely damaged land surfaces. For this reason, this research requires two datasets and since the flattened image crops are the same for both models, it specifically indicates that two sets of labels are required.

3.6.1 K-Means For Model-1

For classifying image crops into two categories: undamaged and damaged areas, the corresponding label set must also contain two values, one for undamaged areas and another for damaged areas. In order to ensure the correct number of clusters K , we applied the elbow method on K-Means clustering. As stated by [31], elbow method is an iterative algorithm which computes the sum of squared errors (SSE) for every value of K starting from $K=1$ till a finite number. A curve plotted will display the “elbow” where SSE decreased steeply. The K for which the elbow occurs is the cluster number for which the clusters produced by K-Means will be most compact and suitable for the dataset. The SSE(or inertia) plotted against the number of clusters shows two distinct elbows (one at 2 and another at 3) for NSW, Bolivia, Chiang Mai, Victoria, Canberra, Salem, Bombala and Harrierville , only one distinct elbow at 2 for Kangaroo Island left side, Kangaroo Island Right side and Tucson and 2 elbows (one at 2 and another at 4) for California. Our hypothesis was to choose three clusters since the DI showed three distinct colors for three types of change. Therefore, the suitable option was to follow the K-Means algorithm elbow method and also to cross check the result by comparing the result with the DI visually. Since the labels are numeric values between 0 and m , the labels are used to visualize the image patches in order to interpret which numeric value corresponds to which group, each time after the algorithm is run. For example, a $90 \times 90 \times 3$ image is constructed using the labels outputted by K-Means for each $(N^k, 32768)$ shaped flattened output where N^k is the number of flattened crops of k^{th} DI and 32768 is the depth of the flattened array. An algorithm of the visual representation process is given in Figure 3.55.

Each visual representation of the labels are compared with each original DI and we had to correctly determine the corresponding numeric label for each of the three classes by means of our visual approximation.

- For the 2 elbows at 2 and 3 : 3 clusters were chosen since the visual representation of the labels produced by K-Means with $K=3$ for all the 8 (NSW, Bolivia, Chiang Mai, Victoria, Canberra, Salem, Bombala and Harrierville) flattened outputs closely matched with the corresponding DI. When $K=2$ was used over the flattened outputs, the visual representation was not fully representative of the DI and displayed minimal similarities with the DI.

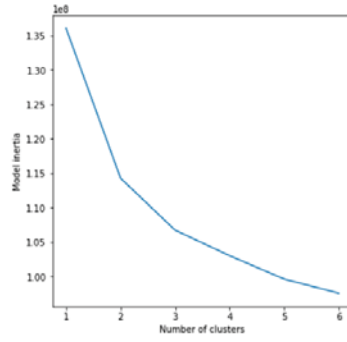


Figure 3.25: Elbow diagram for NSW, Bolivia, Chiang Mai, Victoria, Canberra, Salem, Bombala and Harrietville

- For the single distinct elbow at 2 : For Kangaroo Island (both left and right sides) , 2 clusters were chosen since the visual representation of the labels produced by K-Means with K=2 for both sides were consistent with the DI in terms of similarity. However, this was not the case for Tucson as K=2 on these flattened outputs were not representative of the DI and did not seem consistent with the DI. But with K=3, the visual representation of the flattened crops of Tucson seemed more similar and coherent to the original DI. This is the first anomalous case since it did not follow the elbow method.

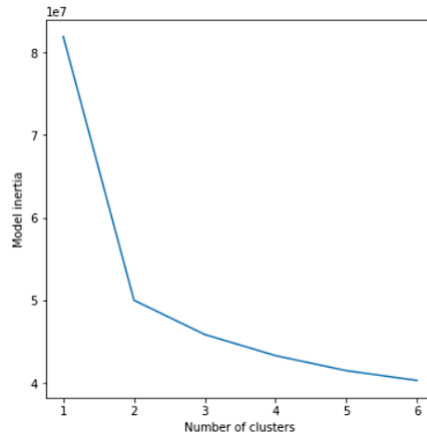


Figure 3.26: Elbow diagram for Kangaroo Island (both left and right sides)

- For the 2 elbows at 2 and 4 : K-Means was performed on the flattened output of California with K=2 however the visual representation from the labels did not satisfy with the original DI since it was not representative of the DI and showed minimal similarities. By performing K-Means with K=4 and K=3, the visual representation matched with the DI in both cases but K=4 displayed a very small portion of patches which were not relevant to damage areas. Hence to keep the clustering labels consistent K=3 was chosen in the end.

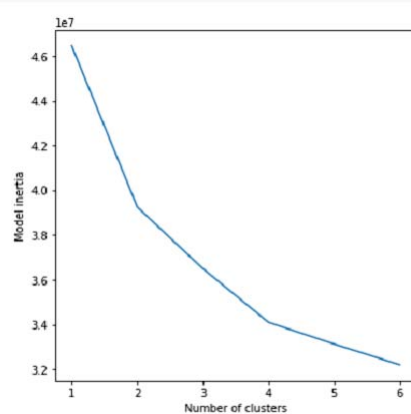


Figure 3.27: Elbow diagram for California

Since for this model, only 2 labels were required, the DI images along with the before and after pairs were compared with the corresponding visual representations to determine which color represented the damaged patches in the test image, in other words, which numeric value of the label corresponded to damage. It was observed that the red/pink patches on the DI which represented damage were consistent with one of the three labels. The other two labels represented unchanged regions and new growth of vegetation or changes in roads/housing plots. Therefore, from the visual representation of the flattened crops, the label corresponding to damaged vegetation was selected. Keeping this “damaged label” constant, the other two labels were given the same numeric value. This means that the labels which are responsible for solely “damaged vegetation” are given one label and the “unchanged vegetation” and “regrowth of vegetation or change due to roads/housing plots” were combined under one label called “undamaged vegetation”. It is to be noted that, K-Means clustering algorithm outputs random numeric values as labels for each cluster every time it runs. This meant that the labels of one flattened output were not consistent with another flattened output. Therefore, in order to bring about consistency in labels, labels which represented “undamaged areas” in each dataset were changed to value 0 and labels which represented “damaged areas” were changed to 1. Thus, all 12 sets of flattened crops contain uniform labels. This finally created the complete dataset for Model-1 with 2 classes : “damaged” and “undamaged”. In the visual representation of each K-Means output, the undamaged areas (label=0) represent two sections: unchanged lands which are given blue color and new growth of vegetation or change in roads/housing plots which are given green color and the damaged areas are given red color. This color scheme is selected only for ease of visualization and representing the original three clusters of K-Means.

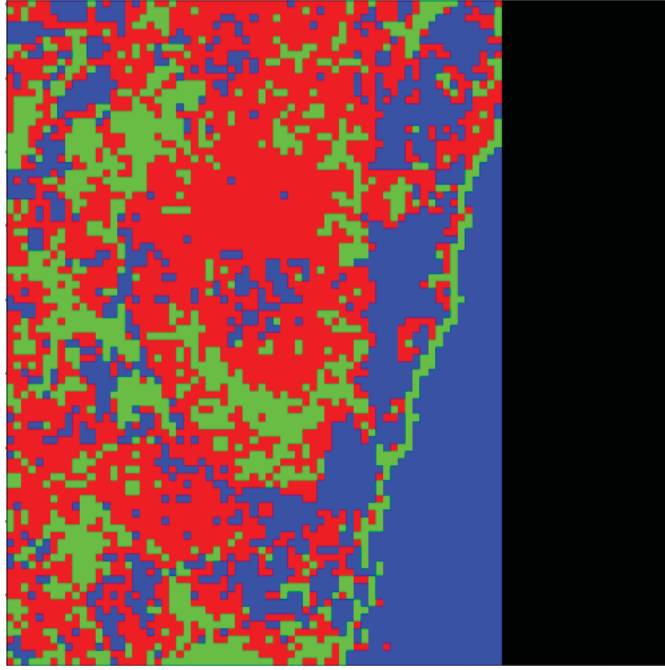


Figure 3.28: Model-1 K-means output for Bombala

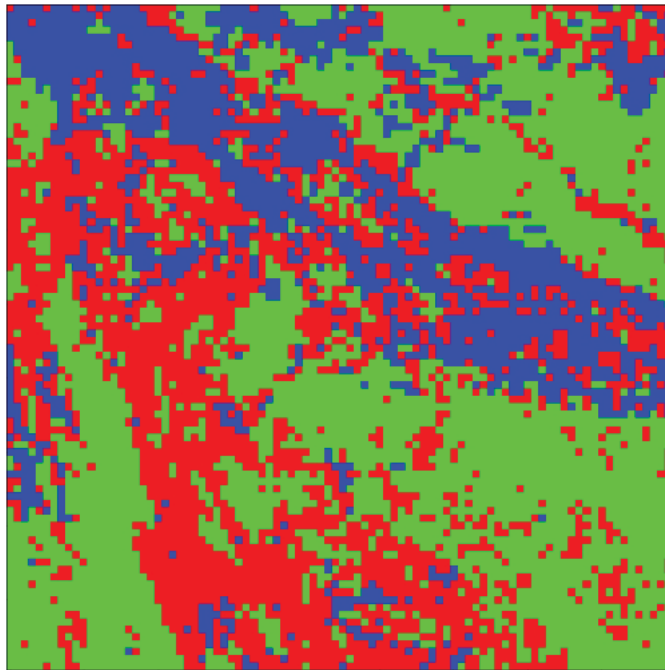


Figure 3.29: Model-1 K-means output for Bolivia

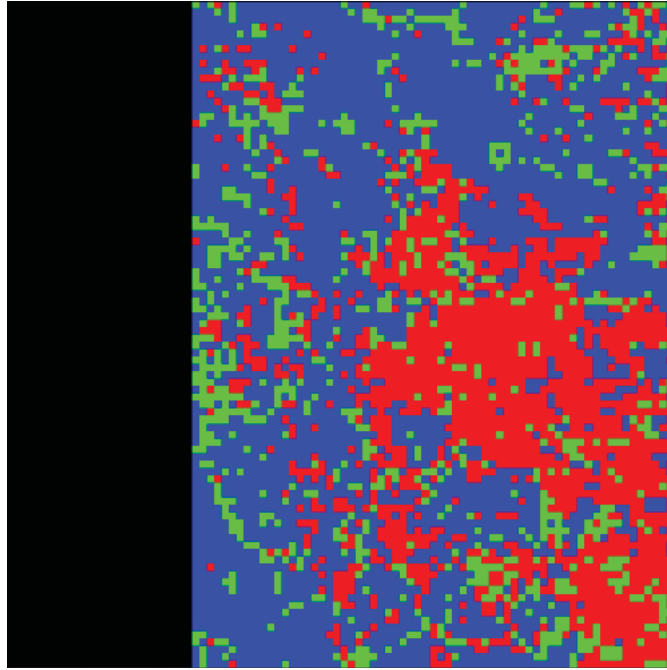


Figure 3.30: Model-1 K-means output for California

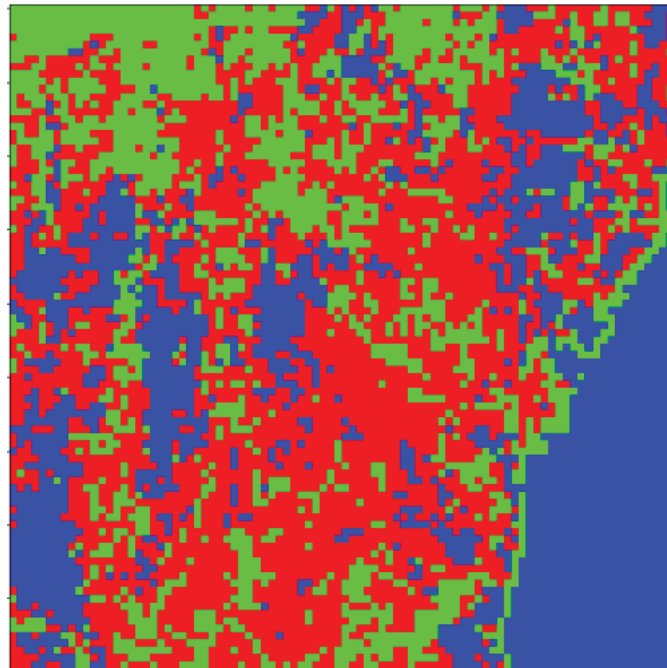


Figure 3.31: Model-1 K-means output for Canberra

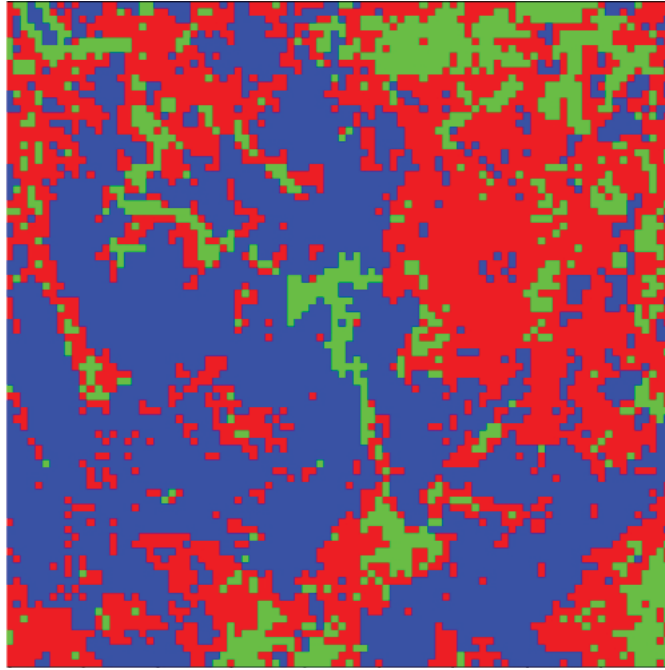


Figure 3.32: Model-1 K-means output for Harrietville



Figure 3.33: Model-1 K-means output for Kangaroo Island left side



Figure 3.34: Model-1 K-means output for Kangaroo Island right side

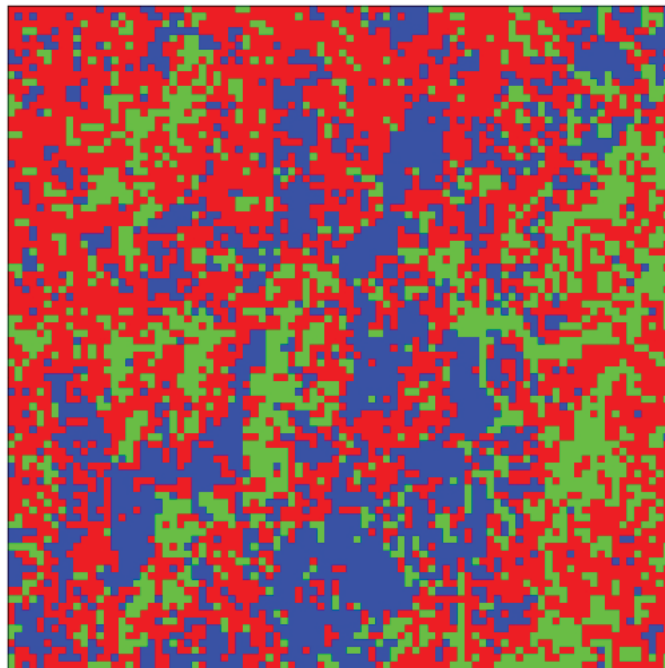


Figure 3.35: Model-1 K-means output for New South Wales

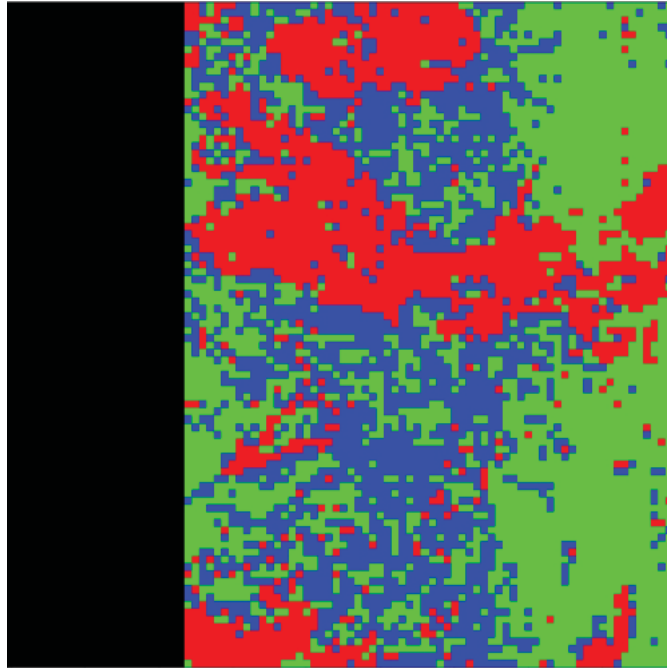


Figure 3.36: Model-1 K-means output for Salem

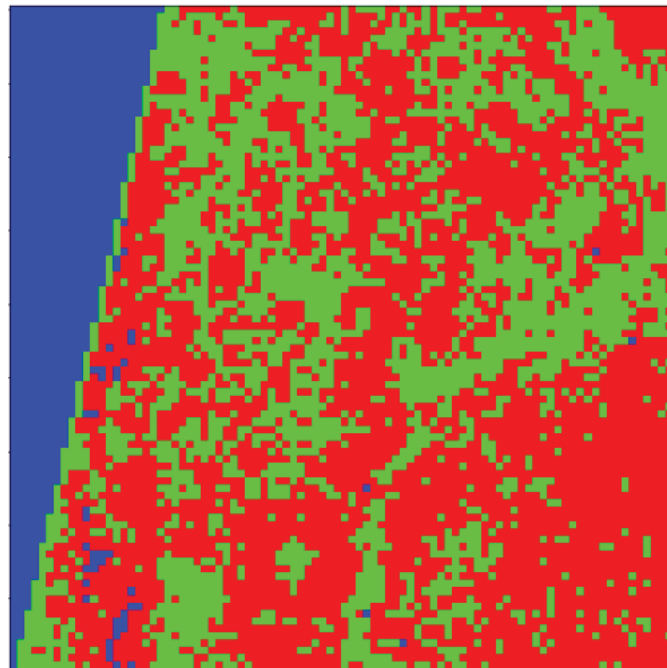


Figure 3.37: Model-1 K-means output for Chiang Mai

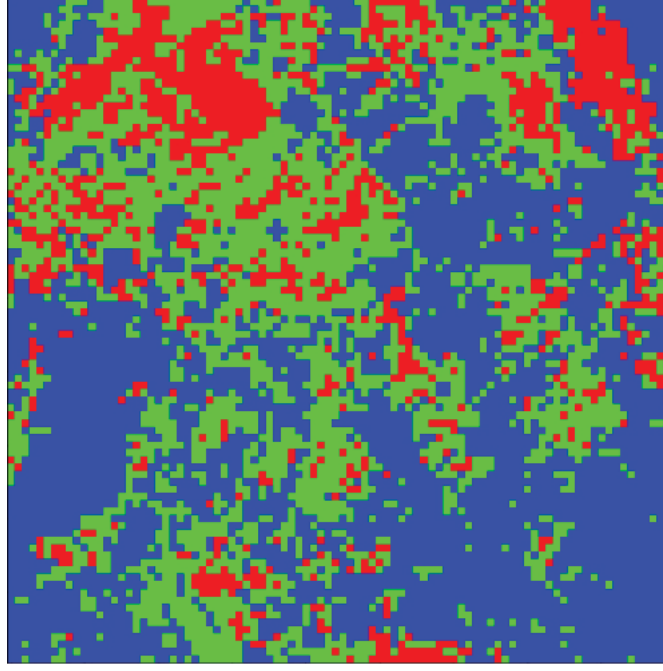


Figure 3.38: Model-1 K-means output for Tucson

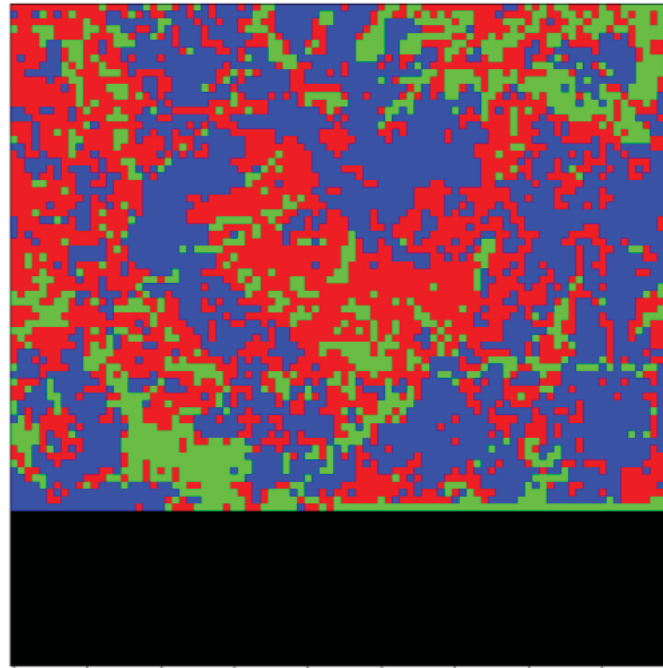


Figure 3.39: Model-1 K-means output for Victoria

3.6.2 K-Means For Model-2

For classifying image crops into three categories: undamaged, mildly damaged areas and severely damaged areas, the corresponding label set must also contain three values, one for undamaged areas and another for classification of two types of damage : mild and severe. For this reason, a second K-Means is carried out only on the damaged flattened outputs. Since the “damaged labels” i.e. label value = 1

from Model-1 are already known, these labels are used to separate out the corresponding “damaged flattened outputs” for K-Means to perform clustering. Similar to Model-1, elbow method is used to obtain the ideal cluster number K. According to our hypothesis, since the aim is to obtain 2 levels of damage within the damaged regions, running K-Means with $K=2$ should give desired outputs. Once again the K-Means algorithm elbow method was performed and the result was cross checked by comparing the output with the DI visually using the algorithm mentioned in Figure 3.55. By plotting the curve of SSE against the number of clusters displayed single distinct elbow at 2 for Chiang Mai, Kangaroo Island right side (a milder elbow than the rest), Kangaroo Island left side, California, Victoria, Bolivia (a milder elbow than the rest), Harrietville and Canberra, two distinct elbows (one at 2 and another at 3) for Salem and Tucson and extremely mild (close to none) elbow for NSW and Bombala. Once again, the suitable option was to follow the K-Means algorithm elbow method and also to cross check the result by comparing the result with the DI visually. Like Model-1, each visual representation of the labels are compared with each original DI and we had to correctly determine the corresponding numeric label for each of the two classes: mild damage and severe damage by means of our visual approximation.

- For the single distinct elbow at 2: 2 clusters were chosen since maximum similarity between the visual representation and the DI along with the before and after image pairs was found for all 8 (Chiang Mai, Kangaroo Island right side, Kangaroo Island left side, California, Victoria, Bolivia, Harrietville and Canberra) sets of flattened outputs. These flattened outputs closely matched with the damaged patches in DI and by close observation the comparison between the each visual representation and each before-after image pair, the high levels of damage were determined from the severely scorched regions in the after images. Hence $K=2$ was chosen.

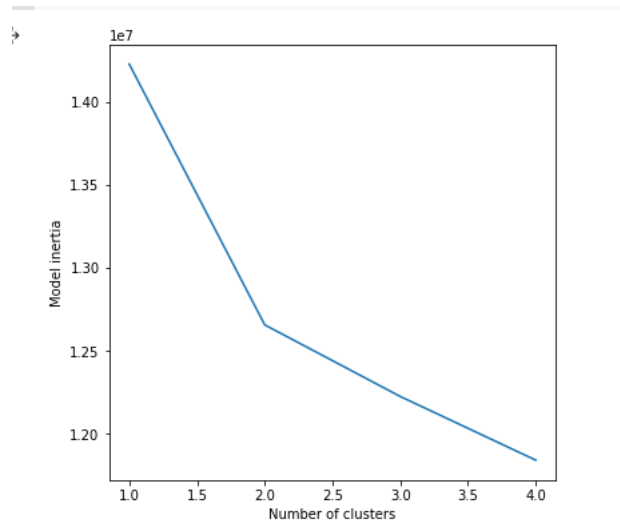


Figure 3.40: Elbow diagram for Chiang Mai, Kangaroo Island right side, Kangaroo Island left side, California, Victoria, Bolivia, Harrietville and Canberra

- For the 2 elbows at 2 and 3: Again for Salem and Tucson, we had to rely upon the comparison of the visual representation with the DI and the before-after

image pairs. It was observed that, $K=3$ formed clusters within bigger clusters made by $K=2$ which meant that those areas potentially had 3 levels of damage. Since the goal of this model is to classify 2 levels of damage only, $K=2$ was chosen. The visual representation produced was representative of the damage levels of before-after image pair and also the DI.

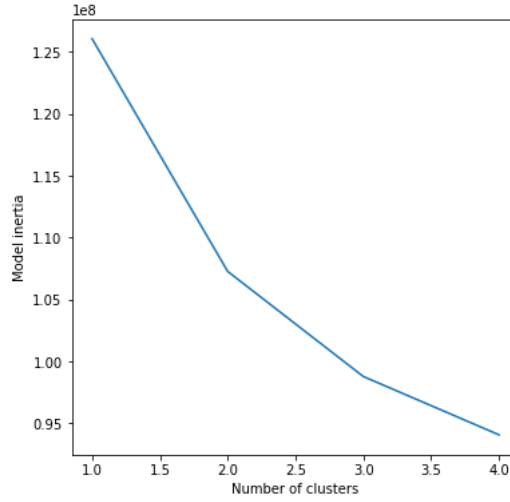


Figure 3.41: Elbow diagram for Salem and Tucson

- For the no elbow : For both NSW and Bombala, K-means with $K=2$ was performed. For both, K-Means divided the damaged patches into two clusters which looked very similar to the before-after image pair damages where very scorched and burnt ground on after images fell into one cluster and mildly damaged regions were grouped in the other cluster. Hence for both $K=2$ were chosen. These presented the second anomalous case where the elbow method could not output any compact cluster number.

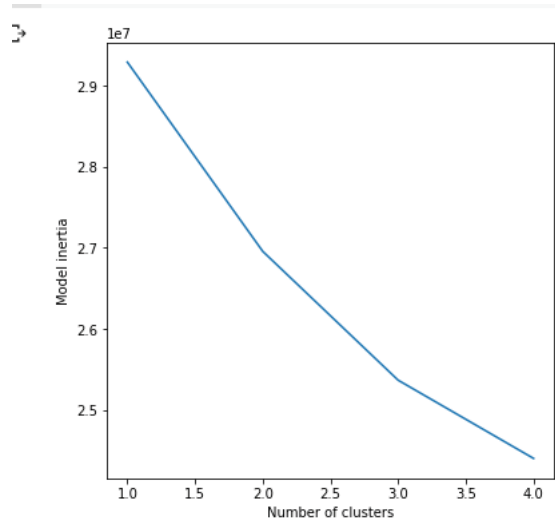


Figure 3.42: Elbow diagram for NSW and Bombala

For this model, three labels are required. The “undamaged” labels already present from Model-1 along with the labels outputted by the second K-Means on the damaged flattened outputs produced labels corresponding to three classes. In this case, by close comparison of the test image of each flattened output with the corresponding before-after image pair along with the DI, it can be observed that heavily brown burnt ground in after images produced labels of one cluster and mild damages produced labels of another cluster. Again, in order to attain consistency in dataset, “undamaged” area is kept at label=0 but labels which represent “mild” damaged areas/crops are converted to label=1 and labels which represent “severe” damaged areas/crops are converted to label=2 across all flattened outputs. Thus, this completes our labelled dataset for Model-2. In the visual representation of each K-Means output, the undamaged areas (label=0) are given black color, the mildly damaged areas (label=1) are given yellow color and the severely damaged areas are given red color for ease of visualization and representing the levels of damage.

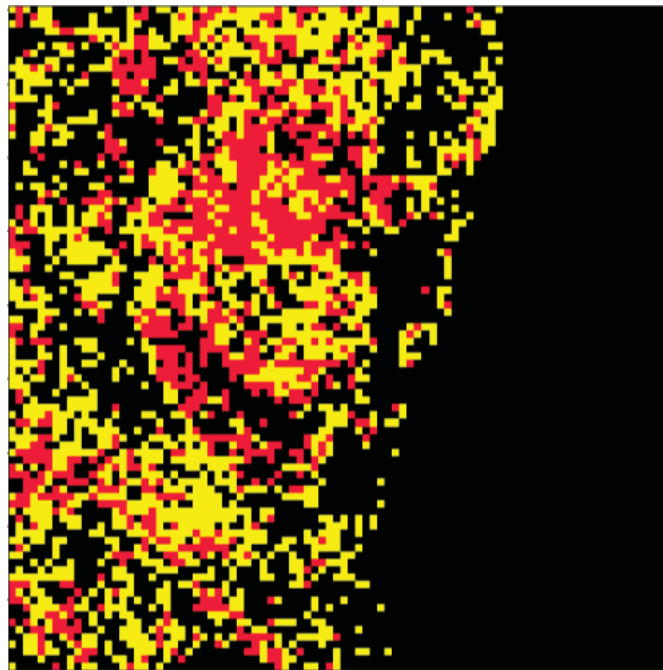


Figure 3.43: Model-2 K-means output for Bombala

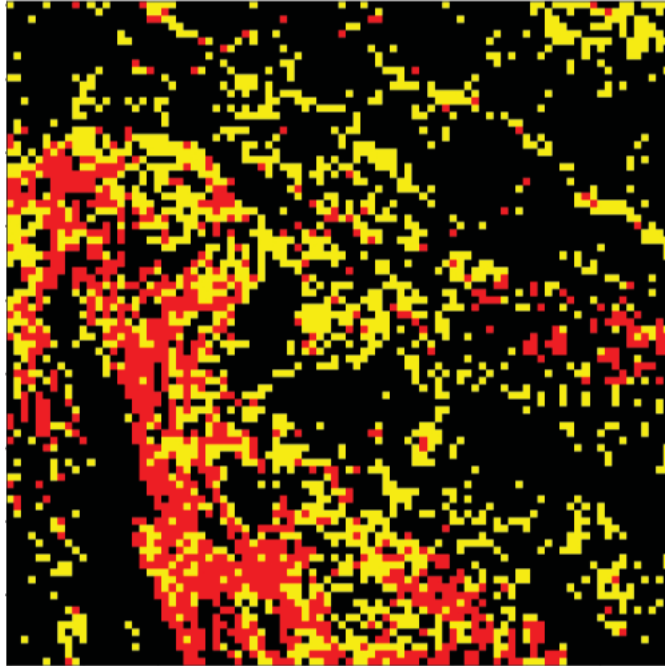


Figure 3.44: Model-2 K-means output for Bolivia

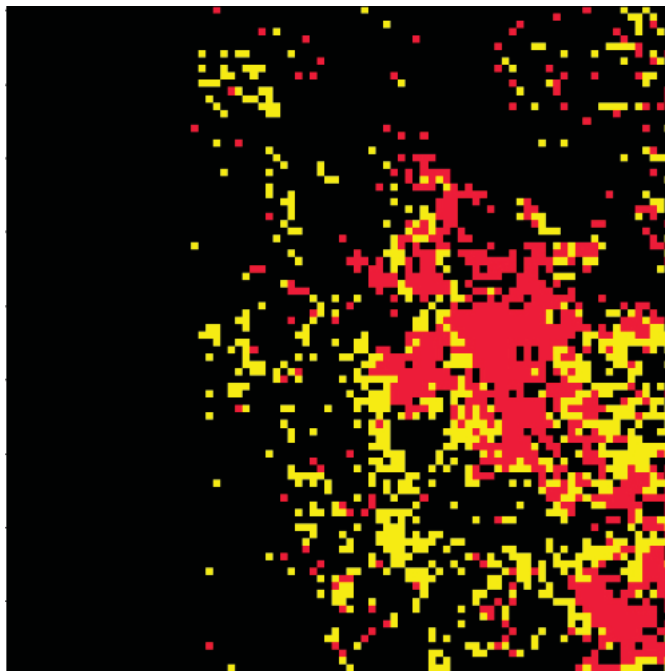


Figure 3.45: Model-2 K-means output for California

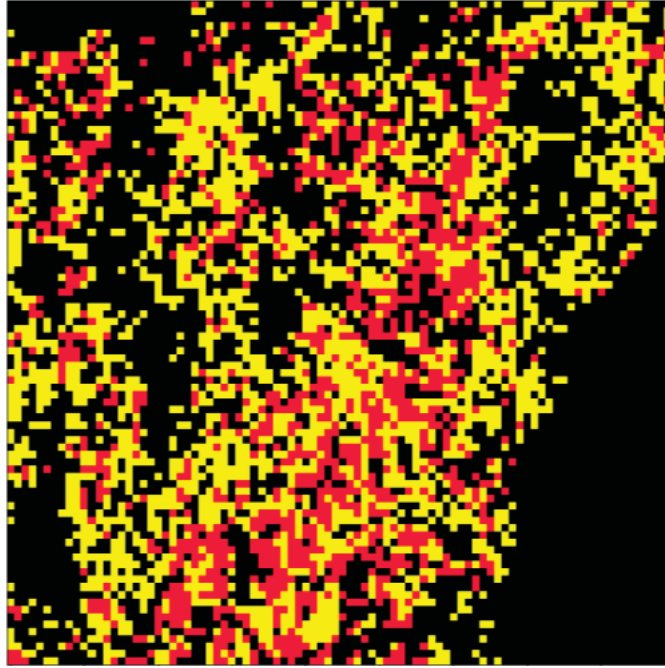


Figure 3.46: Model-2 K-means output for Canberra

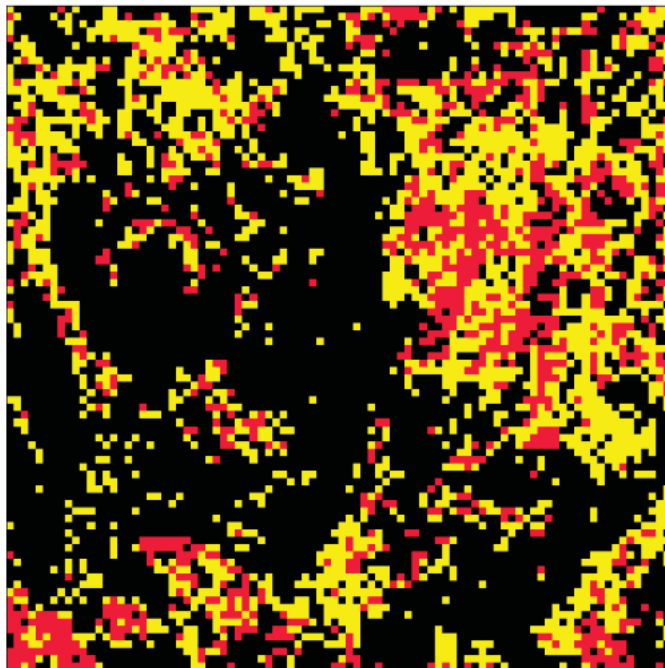


Figure 3.47: Model-2 K-means output for Harrietville

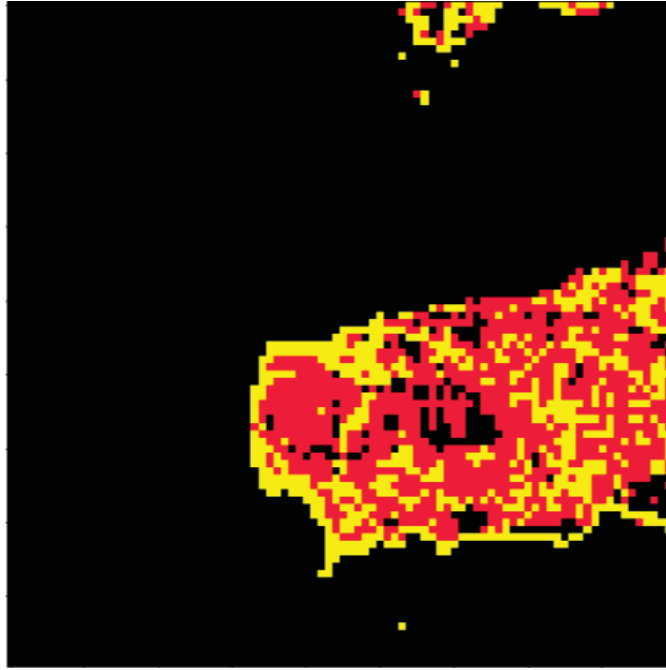


Figure 3.48: Model-2 K-means output for Kangaroo Island left side

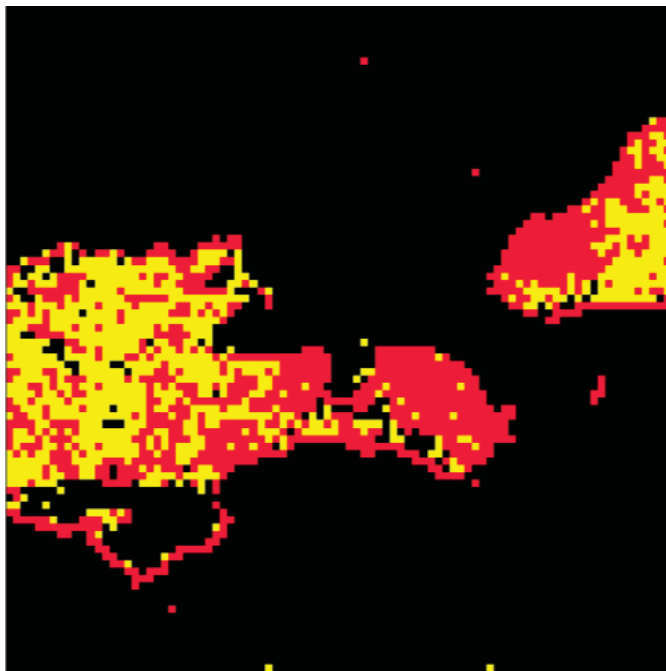


Figure 3.49: Model-2 K-means output for Kangaroo Island right side

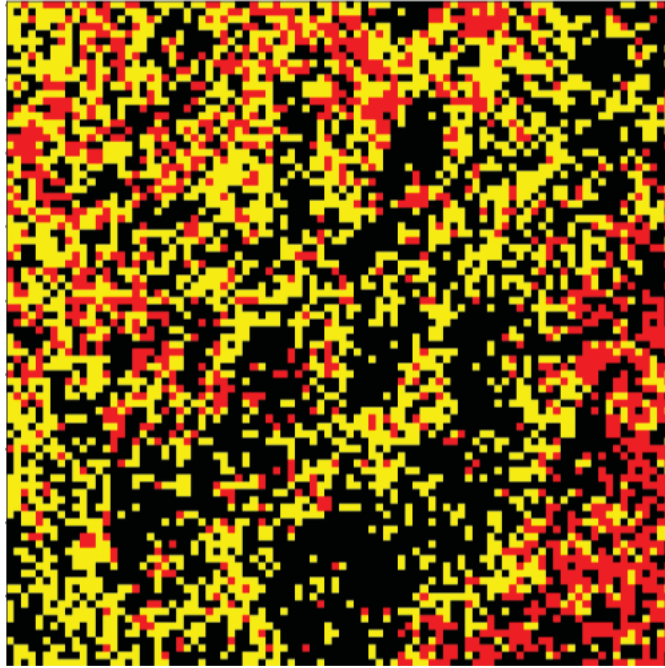


Figure 3.50: Model-2 K-means output for New South Wales

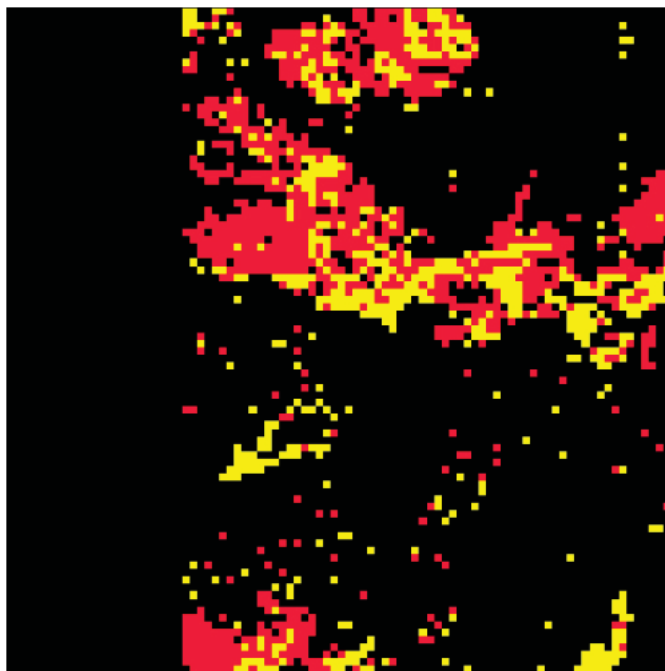


Figure 3.51: Model-2 K-means output for Salem

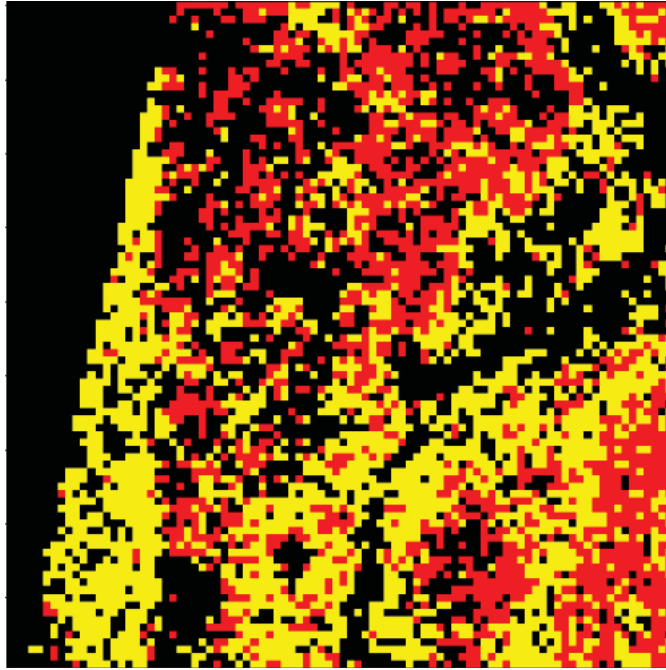


Figure 3.52: Model-2 K-means output for Chiang Mai

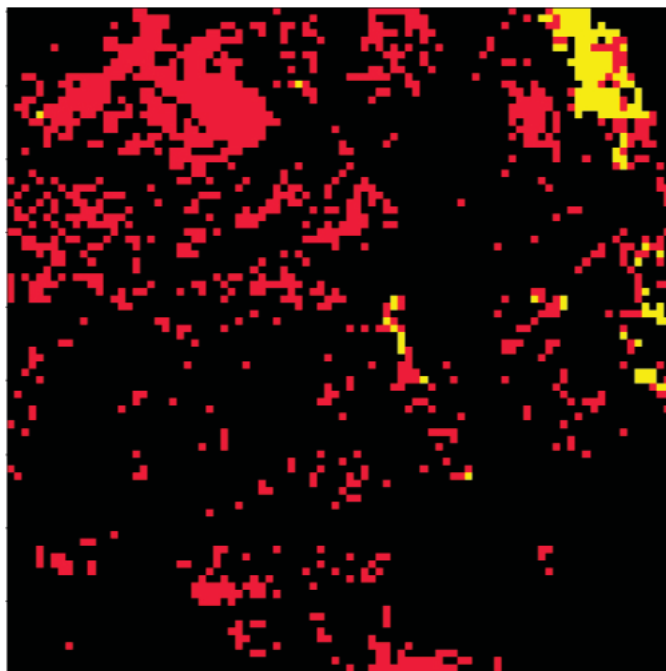


Figure 3.53: Model-2 K-means output for Tucson

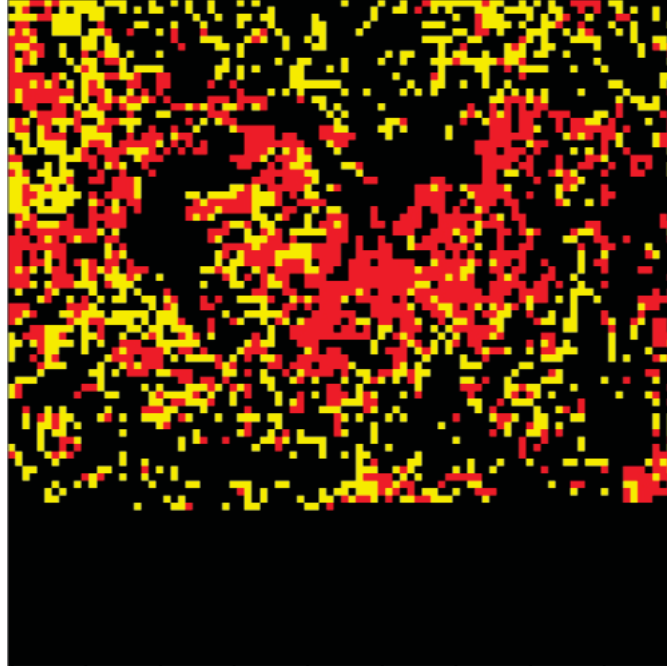


Figure 3.54: Model-2 K-means output for Victoria

3.6.3 Algorithm for visualization

```

Create a 90x90x3 test image with 0 pixels
for each of the 90 rows
  for each of the 90 columns
    if label value is m
      change pixel color of the test image to c_m
    if label value is m-1
      change pixel color of the test image to c_m-1
    .
    .
    .
    if label value is 0
      change pixel color of the test image to c0
  Visualize the test image

```

Figure 3.55: Pseudocode for visualizing k-means labels

Using the above code, each cluster is given a distinct color, i.e. crops which are in the same cluster will have the same label and therefore will be represented using the same color. In doing so, when the test image is plotted and displayed, m distinct colors form patches in the test image which is a representation of the DI in terms of the labels predicted by K-Means.

3.7 Train Test set

With the complete dataset for Model-1 and Model-2, the next phase is supervised classification for both the models. The dataset contains 85,860 flattened feature maps/crops with shape $(N^k, 32768)$. The flattened arrays are reshaped to $(N^k, 4, 4, 2048)$ where 4×4 are the width and height of each crop and 2048 is the number of channels or depth of each crop. The crops are reshaped since they will be inputted into a CNN for classification. Since the ultimate goal of both the models is to be able to predict any given image once they have been trained, tested and evaluated, 3 images : NSW, California and Victoria are kept for predictions only and not included in the train or test set. These 3 images contain 8100 crops of NSW, 5850 crops of California and 6210 crops of Victoria. Therefore, the dataset contains 65700 flattened crops to be split into train and test sets. One thing is to be noted that, these crops were scaled using intensity normalization method before inputting into ResNet-50 and because the CNN model is treated like a continuation from where the ResNet-50 ended, a second scaling is not required on the flattened crops. The train and test data is split into 70% training set and 30% testing set using random state. Therefore, the train set contains 45,990 flattened feature maps and the test set contains 19,710 flattened feature maps. The same flattened image crops are used to train and test both models. The train and test label sets for both the models are coded using one hot encoding method. In this method, the labels of p number of classes are transformed into p binary values where 1 in the binary string represents the presence of the class and 0 represents the absence of the class and there will be no overlapping of 1s [32].

Chapter 4

Model and Evaluation

The aim of this research is to build a Convolutional Neural Network which will act as the continuation from the ResNet-50 model by taking the reshaped flattened feature maps and the labels outputted by K-Means as inputs. Due to lack of hardware resources such as RAM and Secondary Storage, we could not perform grid search algorithm for fine tuning our parameters and for the same reason, we could not apply K-Folds Cross Validation method to obtain an averaged train and test results. Therefore, we resorted to manual fine tuning which is the tweaking of a single hyperparameter by keeping the rest of the hyperparameters fixed followed by training the model, analysing the results obtained and repeating the process with different hyperparameter values till the model reaches satisfactory optimal results. The hyperparameters used in this model are number of filters, learning rate, number of epochs, batch size, number of convolutional layers, number of maxpool layers, number of neurons in dense layers and number of dense layers.

4.1 Model-1

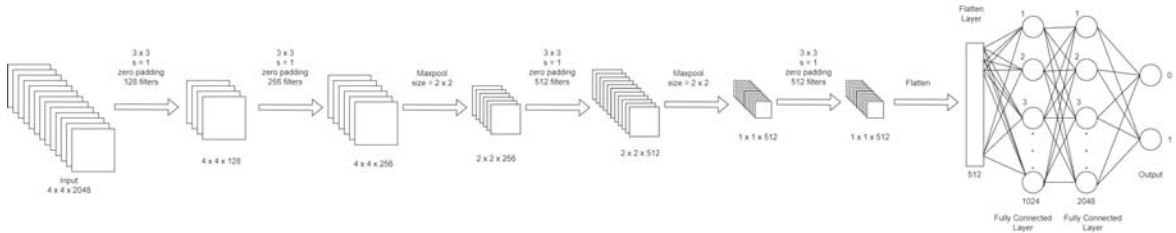


Figure 4.1: Model-1 architecture

Model-1 takes inputs in the shape 4x4x2048. The first convolutional layer contains 128 filters followed by ReLU activation function. The second layer is another convolutional layer with 256 filters followed by a ReLU activation function. The third layer is a maxpool layer and this layer reduced the feature maps' width and height to 2x2. The fourth layer is another convolutional layer with 512 filters followed by a ReLU activation function and a fifth layer of another maxpooling. This pooling reduced the feature maps' width and height to 1x1. The final convolution layer also contains 512 filters followed by a ReLU activation function. After this, the feature maps are flattened followed by two fully connected layers with 1024 and 2048 neurons. ReLU activation function is used after each fully connected layer. Finally,

a softmax classifier is used to output 2 labels where 0 labels indicated damaged land and 1 labels indicated undamaged lands. Binary crossentropy is used as the loss function.

4.2 Model-2

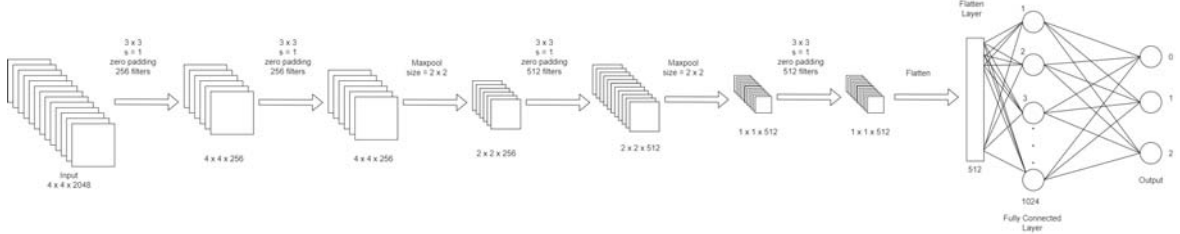


Figure 4.2: Model-2 architecture

This model also takes inputs in the shape 4x4x2048. The first two convolutional layers has 256 filters and each followed by ReLU activation function. Maxpooling is added which reduces the width and height of the feature maps to 2x2. A convolutional layer is added with 512 filters followed by ReLU activation function and another maxpool layer. The height and width of the feature maps decreases to 1x1 after this layer. The last convolutional layer also included 512 filters followed by ReLU activation function. The features maps are flattened and one fully connected layer with 1024 neurons followed by a ReLU activation function is added. Lastly, a softmax classifier is used to classify the feature maps into 3 categories where labels 0 correspond to damaged areas, labels 1 correspond to mildly damaged areas and labels 2 corresponds to severely damaged areas. Here, categorical crossentropy is used as the loss function.

Throughout all the convolutional layers of both models, a kernel size of 3x3 is used and zero padding is used in order to retain the shape of the feature maps due to the already small width and height. For both the models, all the maxpool layers use a window size of 2x2 and Adam optimizer is used to perform optimization with a learning rate of 0.0001. We also included Dropout and Batch Normalization for regulation but the results declined and hence we omitted them. Early stopping has been used as a regularization method to prevent the models from overfitting. The training data is split further into two sections, one for training the model and the other for validating the models' performance during training. Early stopping is a process of terminating the training process from the point when the validation accuracy quickly declines but the training accuracy keeps on increasing. This is an indication of overfitting[33]. Our training data for both the models was split into a 70% training set and a 30% validation set. We used early stopping module of keras with a patience of 4 which means that the training was terminated after 4 epochs of non improving validation accuracy. This gave us an approximation of the number of epochs that the models needed before they start overfitting. After obtaining the approximation of the optimum number of epochs through early stopping, we trained Model-1 over 25 epochs with a batch size of 300 and we trained Model-2 over 24 epochs with a batch size of 300 as well. We saved the best weights i.e the weights

of the models for which the maximum validation accuracy was achieved and used these weights for testing the models and for further predictions.

4.3 Evaluation

Both the models were trained and tested on a custom built Google cloud VM instance with 6vCPUs using Intel Skylake CPU platform and 30 GB memory. In order to evaluate the performance of both the models, evaluation metrics are used. There are several metrics which can be used to assess the models and their predicting capabilities. As described by [34], metrics can be divided into three categories: for making qualitative evaluation of the error which estimates how much error the model was able to reduce, for making probabilistic evaluation of error which measures the difference between the predicted and actual probabilities and for making an evaluation of the ranking of samples made by the model. In this research, the goal of both the models is to make predictions on crops of a whole image after training and thus, both error evaluation and probabilistic accuracy is crucial. In order to evaluate the classifiers qualitatively the error or accuracy that both models achieved were measured. We used the following :

- **Accuracy** : It measures the percentage of correct labels made by the model
- **Kappa statistic** : It measures how much the predicted labels were in agreement with the actual labels
- **Micro f1-score** : It measures the weighted mean of recall and precision of a model
- **Overall Error, OE** : It measures the total errors a model made in prediction [3].

For evaluating the classifiers as good rankers we used AUC which is the area under the Receiver Operator Characteristic (ROC) curve. This is a metric for binary classifiers and therefore, we used it for Model-1 only. It measures the ability of the classifier to differentiate between two classes. The higher the value, the better the model can separate positive and negative classes.

For evaluating the classifiers as good probabilistic models , we used binary cross-entropy during training and testing for Model-1 since it is a binary classifier and categorical cross entropy during training and testing for Model-2 since it is a multiclass classifier. These calculate how much loss occurs when the predicted probabilities diverge from the actual probabilities. Along with these, a heat map based on confusion matrix is also generated to visualize true positive, true negative, false positive and false negative values for both the models.

4.3.1 Model-1 Results

Model-1 trained for 938s and the best weight was taken at training accuracy of 92.84% with a training loss of 0.1682 and validation accuracy of 90.20% with a validation loss of 0.2450. Model-1 tested over 19,710 crops each with a shape of 4x4x2048 and produced a test accuracy of 89.77% with a test binary crossentropy

loss of 0.2507. In terms of other metrics, Model-1 achieved a micro F1 score of 89.77%, kappa statistic of 76.99% and OE of 0.10. The metrics display that Model-1 was able to accurately predict approximately 90% of the time and made overall errors in 10% of the time. The precision and recall are equal and this suggests the model is as much precise/accurate as it is robust, i.e. approximately 90% of the time it makes correct predictions and also does not misclassify any portion of significant data. Approximately 77% of the time the labels predicted by the model agreed with the true labels. Therefore, the quality of reducing errors by Model-1 is satisfactory. The AUC for Model-1 was 0.964 which is very close to 1 which suggests that Model-1 is a good separator of damaged and undamaged classes.

The heatmap showed that the model could predict 12,136 undamaged crops out of total 12,882 undamaged crops and 5,557 damaged crops out of 6,828 damaged crops (Figure 4.5). For NSW (Figure 4.6) it predicted 3,000 undamaged crops out of total 4070 undamaged crops and 3,330 damaged crops out of 4030 damaged crops. For California (Figure 4.7) it predicted 3,638 undamaged crops out of total 4,159 undamaged crops and 1,008 damaged crops out of 1691 damaged crops. Finally for Victoria (Figure 4.6) it predicted 3,310 undamaged crops out of total 3,722 undamaged crops and 1,731 damaged crops out of 2,488 damaged crops. Using the number of true positives (damaged areas) from corresponding heatmaps, the predicted damaged areas are calculated in Table 4.2 by multiplying the number of crops predicted with 1.235 km^2 , i.e. the area represented by each crop.

For the three individual predictions, Model-1 performed quite well on California and Victoria datasets with California with Victoria. Victoria has the highest accuracy (81.27%), micro f1-score (81.27%) and kappa statistic (60.10%) out of the three. However on NSW it performed less well than the other two. One thing is to be noted that, in between California and NSW, NSW has a greater kappa score than California which indicates that on NSW the model had approximately 8% more agreement with the true labels than California.

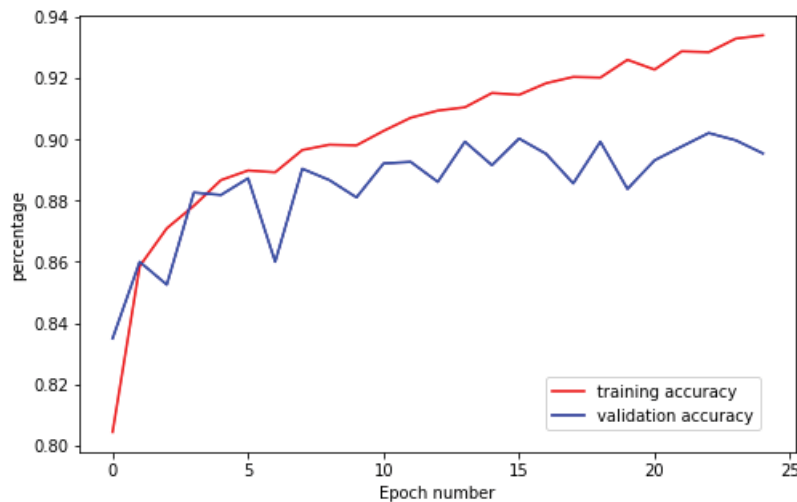


Figure 4.3: Model-1 Training vs Validation curve

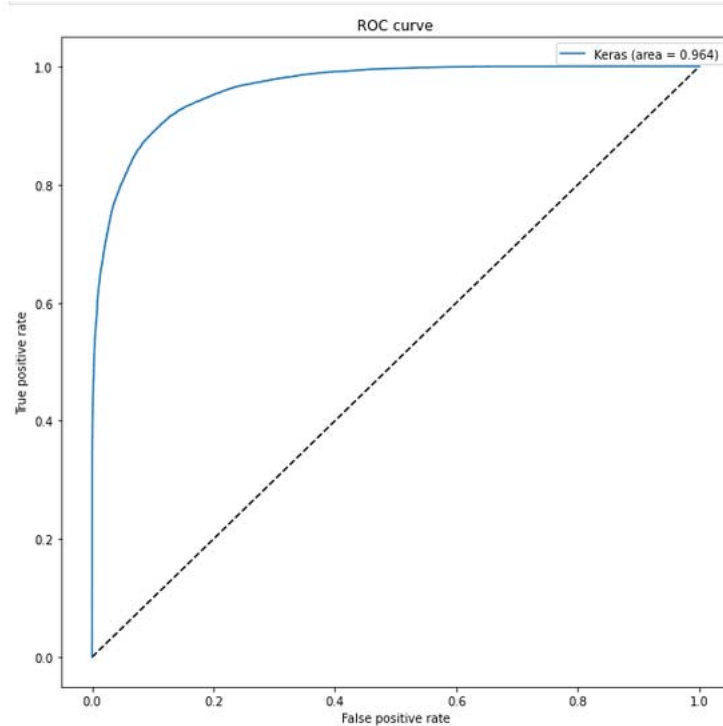


Figure 4.4: Model-1 ROC Curve

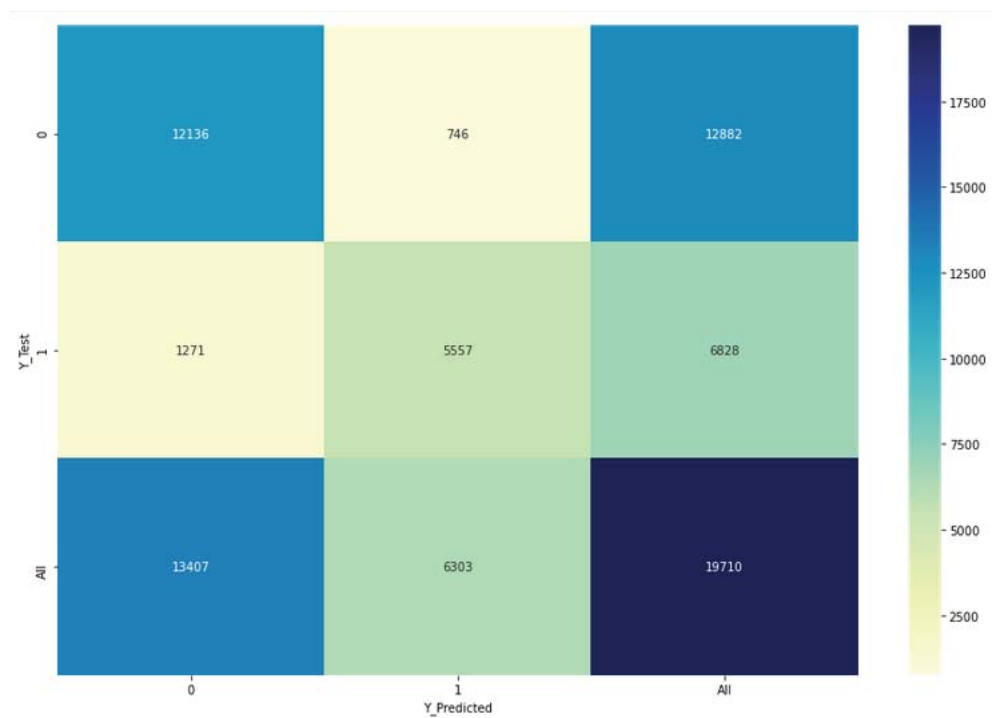


Figure 4.5: Model-1 Heatmap for Test Data

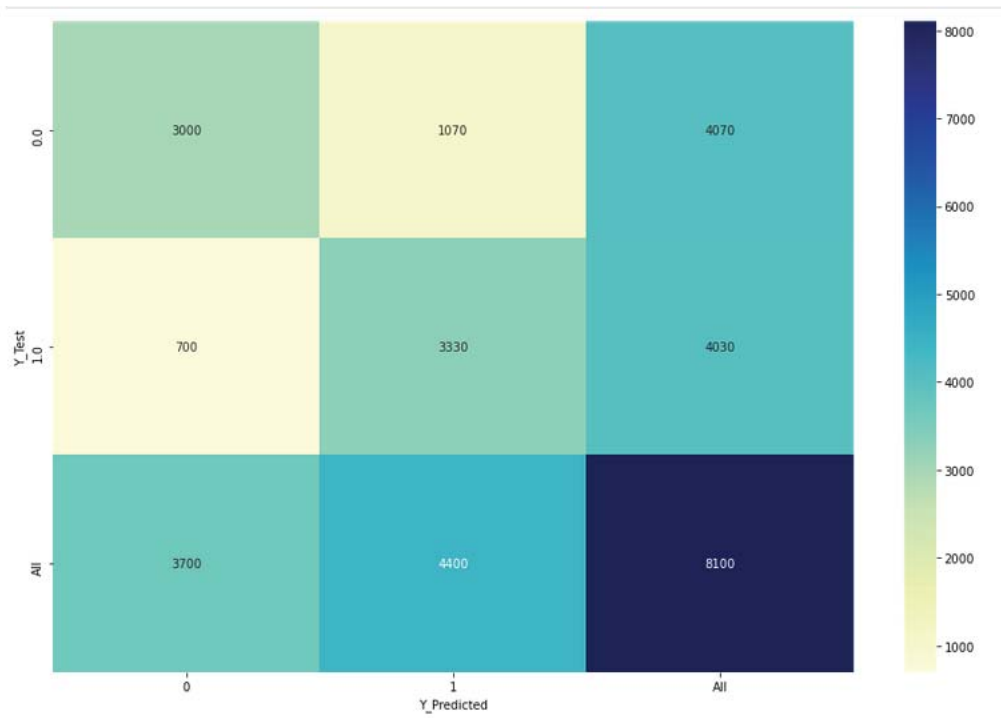


Figure 4.6: Model-1 Heatmap for NSW

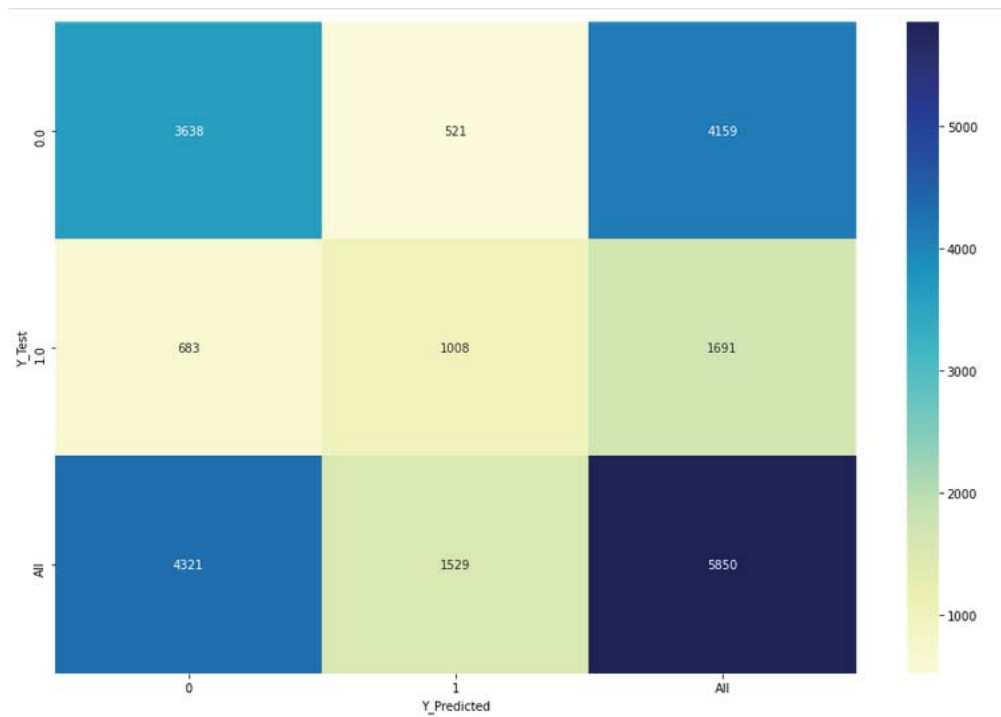


Figure 4.7: Model-1 Heatmap for California

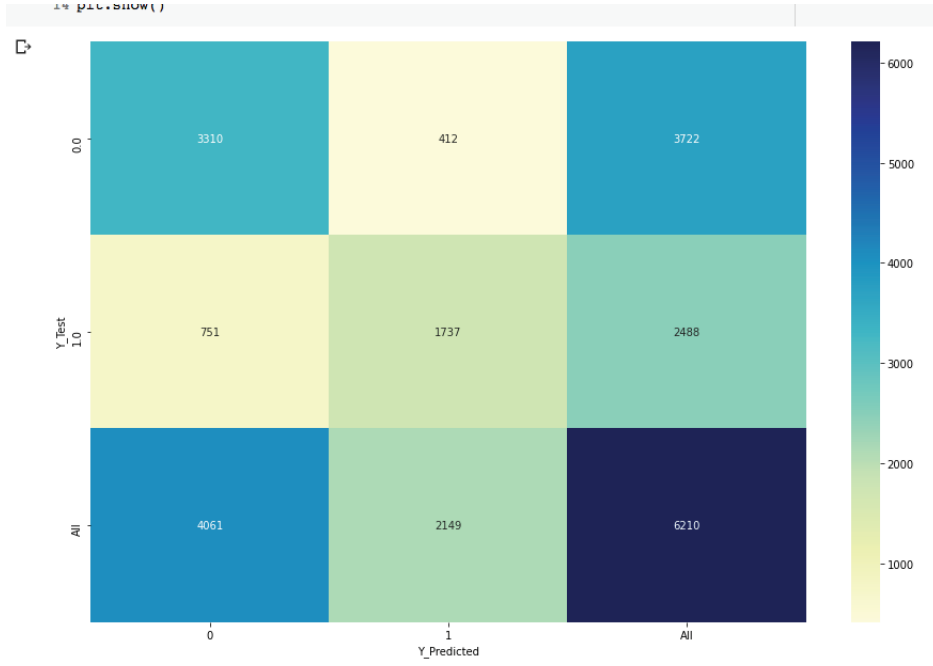


Figure 4.8: Model-1 Heatmap for Victoria

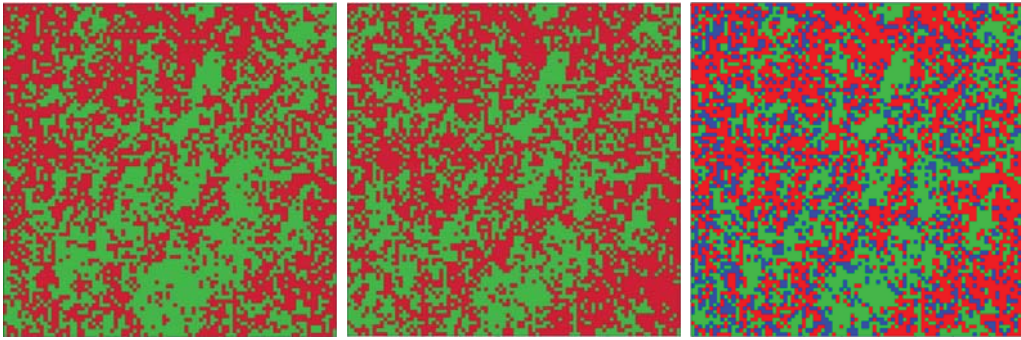


Figure 4.9: Model-1 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for NSW where red is damaged, green is undamaged and blue is the misclassification

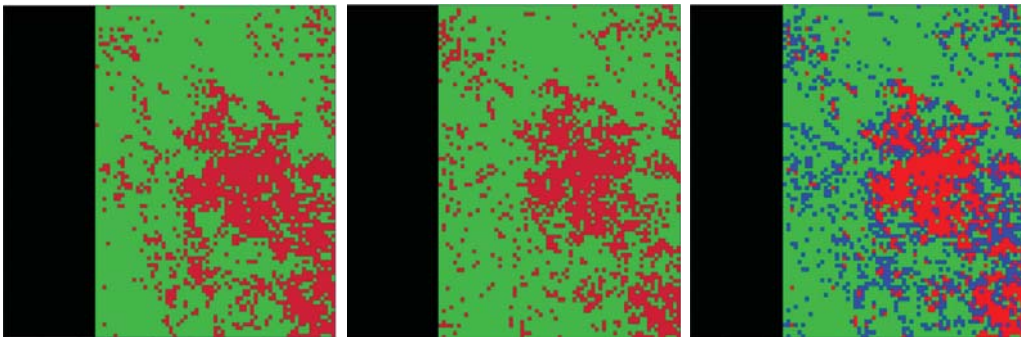


Figure 4.10: Model-1 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for image California where red is damaged, green is undamaged and blue is the misclassification

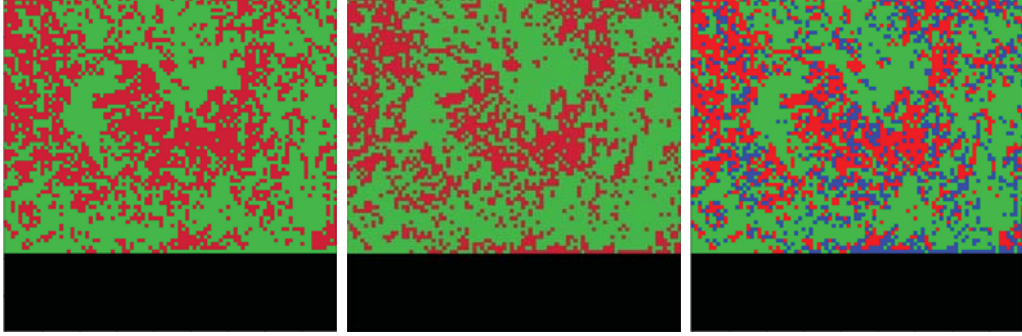


Figure 4.11: Model-1 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for Victoria where red is damaged, green is undamaged and blue is the misclassification

Data	Accuracy	Precision	Recall	Micro f1-score	Kappa	OE
Test data	89.77	89.77	89.77	89.77	76.99	0.10
NSW	78.15	78.15	78.15	78.15	56.31	0.22
California	79.42	79.42	79.42	79.42	48.46	0.21
Victoria	81.27	81.27	81.27	81.27	60.10	0.19

Table 4.1: Model-1 metrics results

Data	Damaged Area(km ²)
NSW	4112.55
California	1244.88
Victoria	2145.195

Table 4.2: Predicted Damaged Areas for Model-1

4.3.2 Model-2 Results

Model-2 trained for 1741s, and the best weight was taken at training accuracy of 89.26% with a training loss of 0.2651 and validation accuracy of 85.93% with a validation loss of 0.3602. Model-2 tested over 19,710 crops each with a shape of 4x4x2048 and produced a test accuracy of 85.69% with a test categorical cross-entropy loss of 0.3741. In terms of other metrics, Model-2 achieved a micro F1-score of 85.69%, kappa statistic of 72.87% and OE of 0.14. These statistics show that Model-2 could correctly predict approximately 86% of the time and made overall errors in other 14% of the time. The same values in both precision and recall of this model suggests that approximately 86% of the time it acts both precisely and robustly by correctly labelling samples and also not mislabelling large significant portions of samples. Model-2 is in agreement with the true labels approximately 73% of the time. These statistics make Model-2 a quite satisfactory model in terms of reducing errors.

The heatmap for test data (Figure 4.13) displayed that this model could predict 11,998 undamaged crops out of 12,882 undamaged crops, 2,680 mildly damaged crops out of 3,700 mildly damaged crops and 2,211 severely damaged crops out of 3,128 severely damaged crops. For NSW (Figure 4.14) it predicted 2,770 undamaged crops out of 4070 undamaged crops, 1547 mildly damaged crops out of 2117 mildly damaged crops and 1020 severely damaged crops out of 1913 severely damaged crops. For California (Figure 4.15) it predicted 3566 undamaged crops out of 4159 undamaged crops, 359 mildly damaged crops out of 814 mildly damaged crops and 565 severely damaged crops out of 877 severely damaged crops. Finally for Victoria (Figure 4.16) it predicted 3272 undamaged crops out of 3722 undamaged crops, 837 mildly damaged crops out of 1185 mildly damaged crops and 248 severely damaged crops out of 1303 severely damaged crops. Using The number of predicted labels from corresponding heatmaps, the predicted damaged areas are calculated in Table 4.4 by multiplying the number of crops predicted with 1.235 km², i.e.the area represented by each crop.

For the three individual predictions, Model-2 did significantly well in classifying California compared to the other two, with accuracy of 76.75%, micro f1-score of 76.75%, kappa statistic of 48.16 and OE of 0.23. However on NSW, Model-2 performed very poorly with an accuracy and micro f1-score of 65.89% and OE of 0.34. From the heatmaps, it can be observed that Model-2 faces difficulty in classifying mildly damaged areas (label=1) as it either classifies a notable amount of mildly damaged crops into either undamaged or severely damaged.

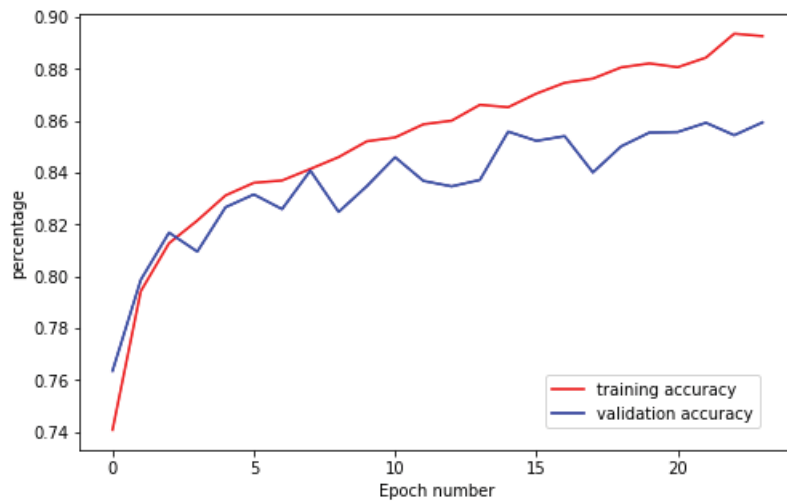


Figure 4.12: Model-2 Training vs Validation curve

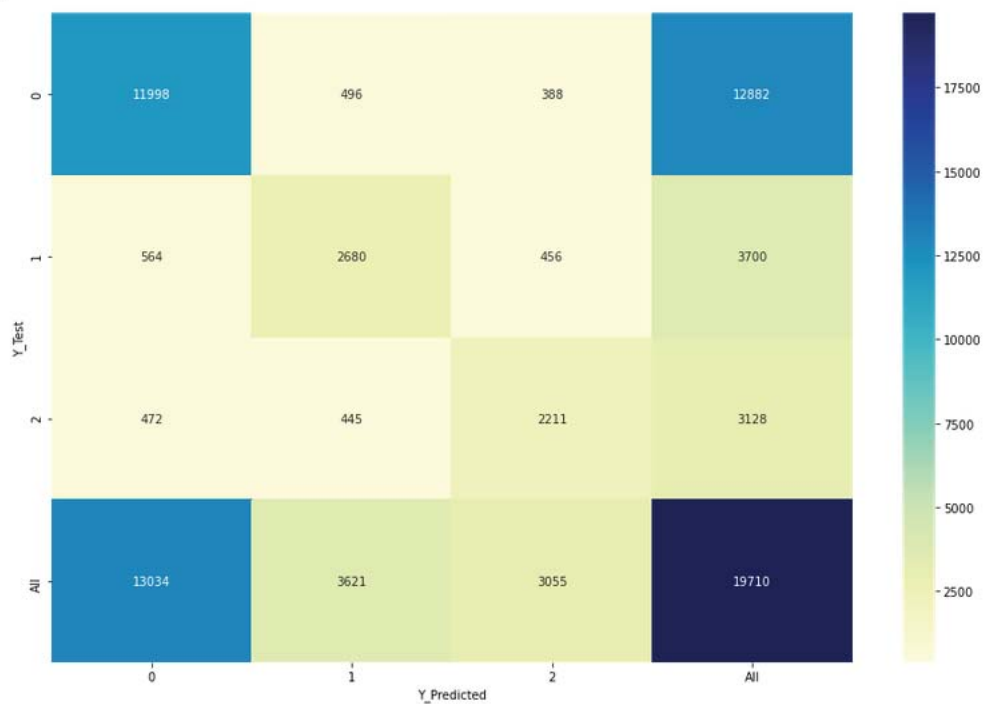


Figure 4.13: Model-2 Heatmap for Test Data

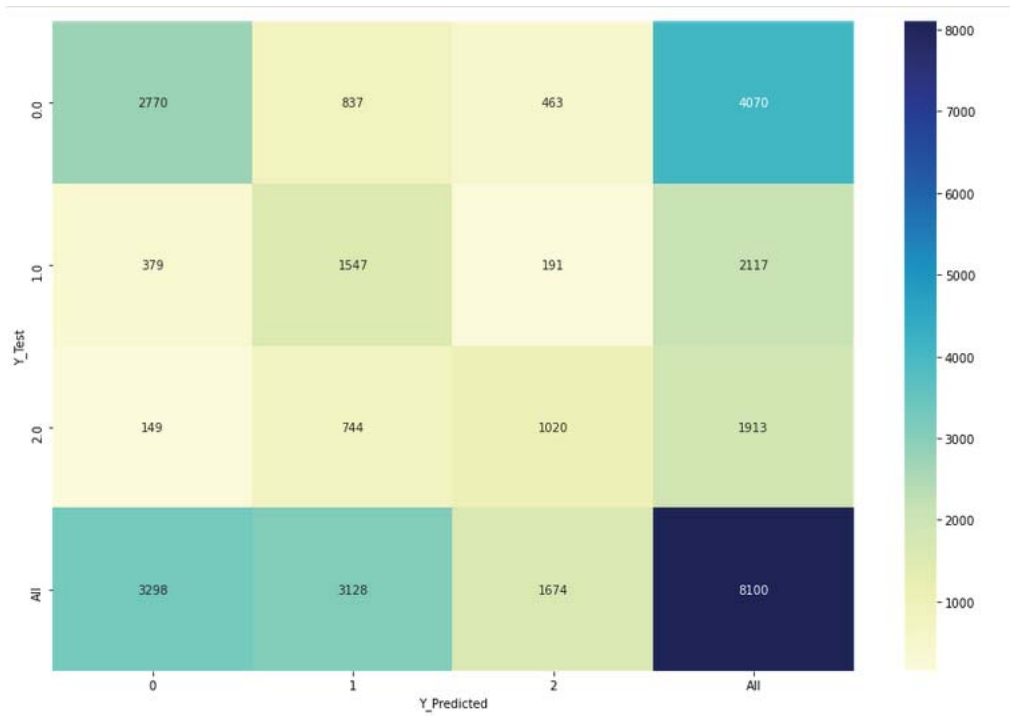


Figure 4.14: Model-2 Heatmap for NSW

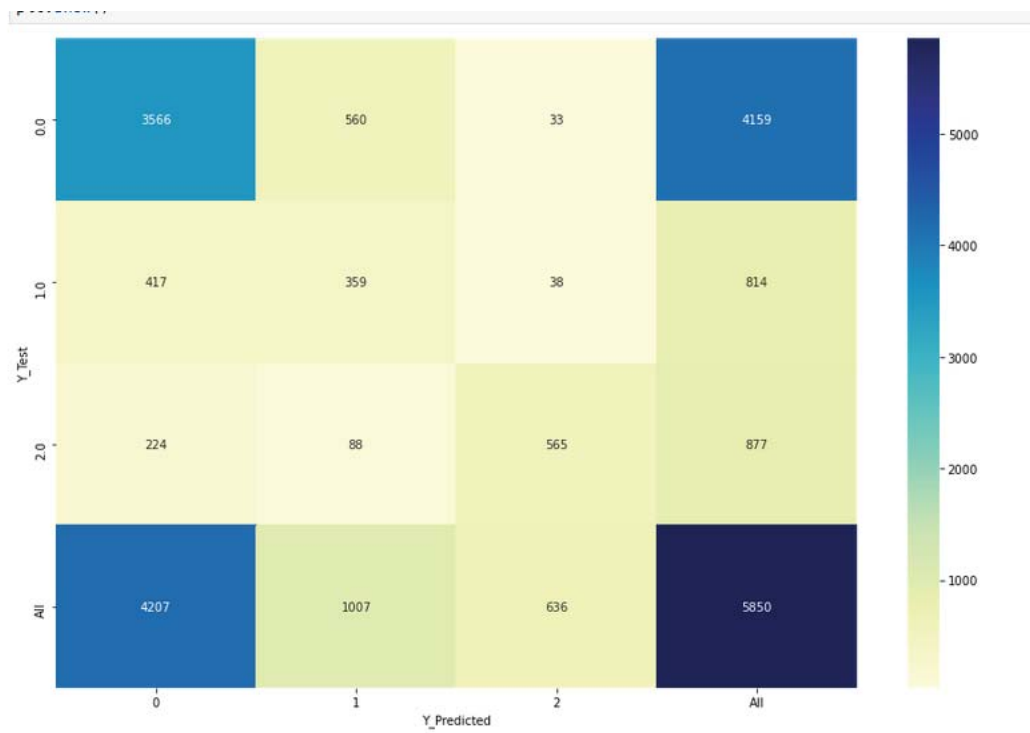


Figure 4.15: Model-2 Heatmap for California

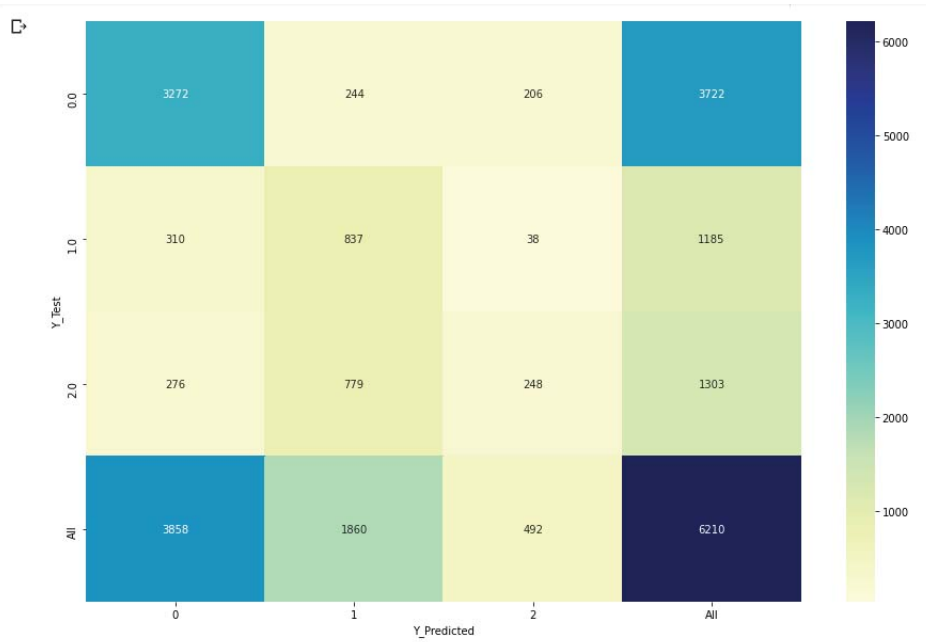


Figure 4.16: Model-2 Heatmap for Victoria

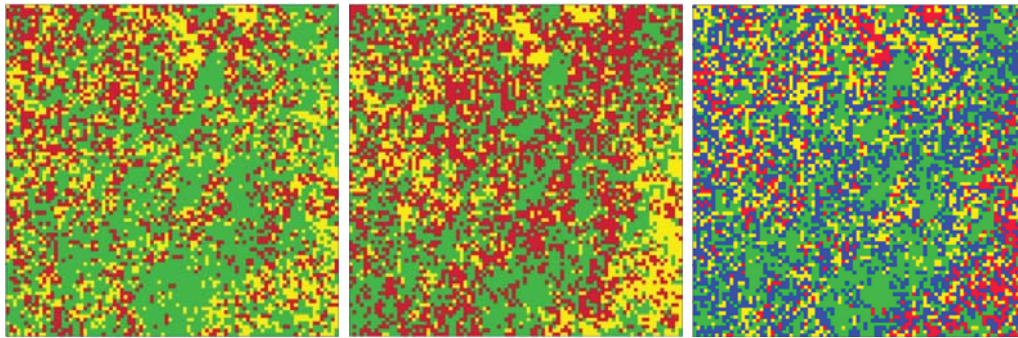


Figure 4.17: Model-2 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for NSW where red is severely damaged, yellow is mildly damaged, green is undamaged and blue is the misclassification

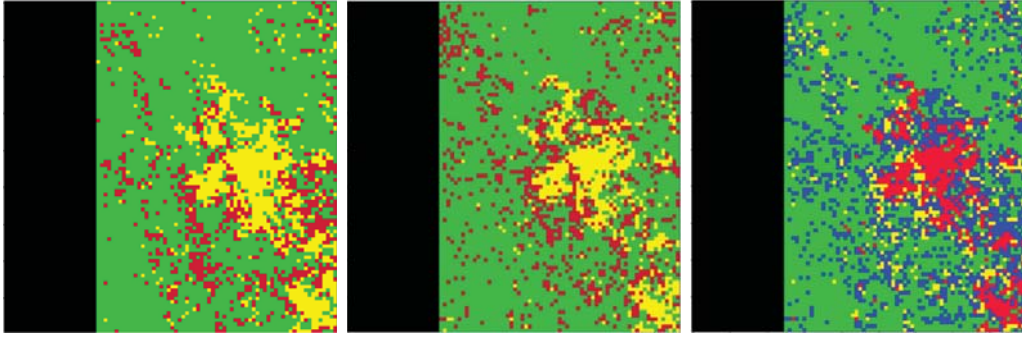


Figure 4.18: Model-2 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for California where red is severely damaged, yellow is mildly damaged, green is undamaged and blue is the misclassification

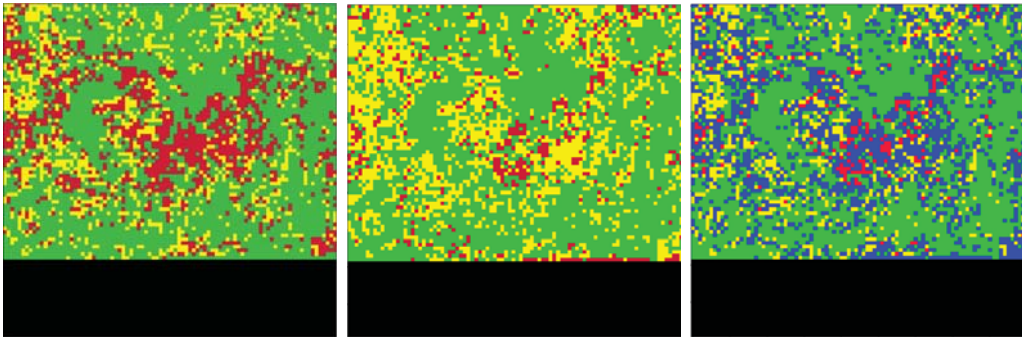


Figure 4.19: Model-2 Comparison of "Test image"(far left), "Predicted image"(center) and "Unmatched labels image"(far right) for Victoria where red is severely damaged, yellow is mildly damaged, green is undamaged and blue is the misclassification

Data	Accuracy	Precision	Recall	Micro f1-score	kappa	OE
Test data	85.69	85.69	85.69	85.69	72.87	0.14
NSW	65.89	65.89	65.89	65.89	47.17	0.34
California	76.75	76.75	76.75	76.75	48.16	0.23
Victoria	70.16	70.16	70.16	70.16	46.13	0.30

Table 4.3: Model-2 metrics results

Data	Mildly Damaged Area(km ²)	Severely Damaged Area(km ²)
NSW	1910.545	1259.7
California	443.365	697.775
Victoria	1033.695	306.28

Table 4.4: Predicted Damaged Areas for Model-2

4.3.3 Analysis

Comparing Model-1 and Model-2 results show that in all metrics, Model-1 outperformed Model-2 by approximately 4%. This is evident since Model-2 is a specialization of Model-1 as it detects levels of damage too and hence more number of classes will correspond to less accuracy comparatively. In terms of the three individual predictions, the Model-2 performed more poorly on NSW dataset compared to Model-1 since in Model-1 all the scores except kappa statistic were in close range of values but in Model-2 NSW is lagging behind in metrics by a large margin. On Victoria dataset, Model-1 performed better by a small amount than on California dataset and on California dataset Model-2 performed better on Victoria dataset by a larger percentage compared to California dataset. Both models perform better at classifying undamaged areas compared to other classes. This is obvious since the number of crops for undamaged regions is higher. This is due to the damage covering only portions of the whole area that each image contained. The reason for classification accuracy of both the models being lower than most classifiers in this field could be due to the fact that levels of damage contained complex features which were not fully represented by ResNet-50 since it is pretrained on daily object dataset and not satellite images. Thus, this could have led to anomalous K-Means clustering which led to multiple wrong labels. Another reason could be the fact that not all the locations contained whole forest areas and there were roads, buildings and city parts in some images which could have been mixed with some clusters during K-Means.

Chapter 5

Conclusion and Future Work

In this research, we have constructed two CNN model architectures, Model-1 and Model-2 that classify the flattened feature maps into undamaged and damaged classes and undamaged, mildly damaged and severely damaged classes respectively. We collected before and after Sentinel-2 images from very recent fire damaged vegetation areas from 12 locations which are then used to form three channel DI using green, red and NIR bands. The difference images are then cropped into 122x122x3 image crops to form a dataset of 85,860 image crops. Among these crops 20,160 of NSW, California and Victoria were separately kept for predicting and the remaining data was split into 70% for training the models and 30% for testing. Our results showed a training accuracy of 92.84% and 89.26% for Model-1 and Model-2 respectively. For Model-1, the test accuracy reached 89.77% and recorded 89.77%, 76.99% and 0.10 in micro f1-score, kappa statistic and OE respectively. For Model-2 the test accuracy was 85.69% and had 85.69%, 72.87% and 0.14 values in micro f1-score, kappa statistic and OE respectively. Our Model-1 outperformed Model-2 by 4% in all metrics which was expected as the number of classes was higher for model-2. This report uses ResNet-50 for transfer learning which was pretrained on Imagenet dataset to classify objects instead of satellite images classification which can be more complex and this may have reduced the performance but still obtained very promising results. This method of using high resolution satellite imagery to classify and approximate damaged vegetation caused by forest fire has the potential to drastically alleviate the current loss of vegetation and wildlife which in the long run will improve the overall ecosystem and may also aid in slowing down the global warming rates. In the future, other robust pretrained networks such as VGG19 and AlexNet can be used as a comparative analysis to find the best network for feature extraction from multispectral satellite imagery. The report can further use normalized difference indices calculated from the bands such as NDVI, PSR and NDWI. in-place of the actual bands or along with the bands to analyze if the inclusion of these indices improves the result. Furthermore, other satellite images such as Landsat-8 images or aerial images can be used to observe if results improve.

Bibliography

- [1] C. Nunez, *Deforestation and its effect on the planet*, 2019, Feb. 2019. [Online]. Available: <https://www.nationalgeographic.com/environment/global-warming/deforestation/>.
- [2] A. Woodward, *Australia's fires are 46% bigger than last year's brazilian amazon blazes. there are at least 2 months of fire season to go.* [Online]. Available: <https://www.insider.com/australia-fires-burned-twice-land-area-as-2019-amazon-fires-2020-1>.
- [3] M. Gong, H. Yang, and P. Zhang, "Feature learning and change feature classification based on deep learning for ternary change detection in sar images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 129, pp. 212–225, 2017, ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2017.05.001>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271616301459>.
- [4] S. Mishra, P. Shrivastava, and P. Dhurvey, "Change detection techniques in remote sensing: A review," *International Journal of Wireless and Mobile Communication for Industrial Systems*, vol. 4, pp. 1–8, Apr. 2017. DOI: 10.21742/ijwmcis.2017.4.1.01.
- [5] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, Mar. 2005, ISSN: 1057-7149. DOI: 10.1109/TIP.2004.838698.
- [6] A. Asokan and J. Anitha, "Change detection techniques for remote sensing applications: A survey," *Earth Science Informatics*, vol. 12, no. 2, pp. 143–160, Jun. 2019, ISSN: 1865-0481. DOI: 10.1007/s12145-019-00380-5.
- [7] A. Albert, J. Kaur, and M. Gonzalez, "Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale," *arXiv:1704.02965 [cs]*, Sep. 2017, arXiv: 1704.02965. [Online]. Available: <http://arxiv.org/abs/1704.02965>.
- [8] M. A. Kadhim and M. H. Abed, "Convolutional neural network for satellite image classification," in *Intelligent Information and Database Systems: Recent Developments*, M. Huk, M. Maleszka, and E. Szczerbicki, Eds. Cham: Springer International Publishing, 2020, pp. 165–178, ISBN: 978-3-030-14132-5. DOI: 10.1007/978-3-030-14132-5_13. [Online]. Available: https://doi.org/10.1007/978-3-030-14132-5_13.

- [9] S. N. K. B. Amit and Y. Aoki, “Disaster detection from aerial imagery with convolutional neural network,” in *2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, IEEE, Sep. 2017, pp. 239–245, ISBN: 9781538607169. DOI: 10.1109/KCIC.2017.8228593. [Online]. Available: <http://ieeexplore.ieee.org/document/8228593/>.
- [10] V. Yaloveha, D. Hlavcheva, and A. Podorozhniak, “Usage of convolutional neural network for multispectral image processing applied to the problem of detecting fire hazardous forest areas,” *Advanced Information Systems*, vol. 3, no. 1, pp. 116–120, May 2019, ISSN: 2522-9052. DOI: 10.20998/2522-9052.2019.1.19.
- [11] N. Audebert, B. Le Saux, and S. Lefevre, “Deep learning for classification of hyperspectral data: A comparative review,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 159–173, Jun. 2019, ISSN: 2168-6831, 2473-2397, 2373-7468. DOI: 10.1109/MGRS.2019.2912563.
- [12] M. I. Sameen, B. Pradhan, and O. S. Aziz, “Classification of very high resolution aerial photos using spectral-spatial convolutional neural networks,” *Journal of Sensors*, vol. 2018, pp. 1–12, Jun. 2018, ISSN: 1687-725X, 1687-7268. DOI: 10.1155/2018/7195432.
- [13] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, “DeepSAT: A learning framework for satellite imagery,” in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, Nov. 2015, pp. 1–10, ISBN: 9781450339674. DOI: 10.1145/2820783.2820816. [Online]. Available: <https://dl.acm.org/doi/10.1145/2820783.2820816>.
- [14] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, “Deep learning classification of land cover and crop types using remote sensing data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, May 2017, ISSN: 1545-598X, 1558-0571. DOI: 10.1109/LGRS.2017.2681128.
- [15] Z. M. Hamdi, M. Brandmeier, and C. Straub, “Forest damage assessment using deep learning on high resolution remote sensing data,” *Remote Sensing*, vol. 11, no. 17, p. 1976, Aug. 2019, ISSN: 2072-4292. DOI: 10.3390/rs11171976. [Online]. Available: <http://dx.doi.org/10.3390/rs11171976>.
- [16] J. Alzubi, A. Nayyar, and A. Kumar, “Machine learning from theory to algorithms: An overview,” *Journal of Physics: Conference Series*, vol. 1142, p. 012012, Nov. 2018, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1142/1/012012.
- [17] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: from theory to algorithms*. Cambridge University Press, 2014, ISBN: 9781107057135.
- [18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539.
- [19] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, ISSN: 0899-7667, 1530-888X. DOI: 10.1162/neco_a_00990.

- [20] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv:1511.08458 [cs]*, Dec. 2015, arXiv: 1511.08458. [Online]. Available: <http://arxiv.org/abs/1511.08458>.
- [21] Y. Wang, X. Wang, and J. Jian, *Remote sensing landslide recognition based on convolutional neural network*, Sep. 2019. DOI: <https://doi.org/10.1155/2019/8389368>. [Online]. Available: <https://www.hindawi.com/journals/mpe/2019/8389368/>.
- [22] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv:1811.03378 [cs]*, Nov. 2018, arXiv: 1811.03378. [Online]. Available: <http://arxiv.org/abs/1811.03378>.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385 [cs]*, Dec. 2015, arXiv: 1512.03385. [Online]. Available: <http://arxiv.org/abs/1512.03385>.
- [24] R. Nainggolan, R. Perangin-angin, E. Simarmata, and A. F. Tarigan, "Improved the performance of the k-means cluster using the sum of squared error (sse) optimized by using the elbow method," *Journal of Physics: Conference Series*, vol. 1361, p. 012015, Nov. 2019, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1361/1/012015.
- [25] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conference Series: Materials Science and Engineering*, vol. 336, p. 012017, Apr. 2018, ISSN: 1757-8981, 1757-899X. DOI: 10.1088/1757-899X/336/1/012017.
- [26] Q. Wang and P. M. Atkinson, "Spatio-temporal fusion for daily sentinel-2 images," *Remote Sensing of Environment*, vol. 204, pp. 31–42, Jan. 2018, ISSN: 00344257. DOI: 10.1016/j.rse.2017.10.046.
- [27] *Sentinel 2 bands and combinations 2019*, Nov. 2019. [Online]. Available: <https://gisgeography.com/sentinel-2-bands-combinations/>.
- [28] *Copernicus open access hub*. [Online]. Available: <https://scihub.copernicus.eu/dhus/>.
- [29] *User guides - sentinel-2 msi - level-2 processing - sentinel online - sentinel*. [Online]. Available: <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/processing-levels/level-2>.
- [30] B. Petrovska, E. Zdravevski, P. Lameski, R. Corizzo, I. Štajduhar, and J. Lerga, "Deep learning for feature extraction in remote sensing: A case-study of aerial scene classification," *Sensors*, vol. 20, no. 14, p. 3906, Jul. 2020, ISSN: 1424-8220. DOI: 10.3390/s20143906.
- [31] C. Yuan and H. Yang, "Research on k-value selection method of k-means clustering algorithm," *J*, vol. 2, no. 2, pp. 226–235, Jun. 2019, ISSN: 2571-8800. DOI: 10.3390/j2020016.
- [32] K. Potdar, T. S., and C. D., "A comparative study of categorical variable encoding techniques for neural network classifiers," *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7–9, Oct. 2017, ISSN: 09758887. DOI: 10.5120/ijca2017915495.

- [33] L. Prechelt, “Early stopping - but when?” In *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Springer Berlin Heidelberg, 1998, vol. 1524, pp. 55–69, ISBN: 9783540653110. DOI: 10.1007/3-540-49430-8_3. [Online]. Available: http://link.springer.com/10.1007/3-540-49430-8_3.
- [34] C. Ferri, J. Hernández-Orallo, and R. Modroiu, “An experimental comparison of performance measures for classification,” *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, Jan. 2009, ISSN: 01678655. DOI: 10.1016/j.patrec.2008.08.010.