Satellite Image Classification using deep learning

A THESIS

Submitted by

Sanjaykumar
(RCAS2021MDB009)

in partial fulfillment for the award of the degree of

MASTER OF SCIENCE
SPECIALIZATION IN
DATA SCIENCE AND BUSINESSES ANALYSIS

DEPARTMENT OF COMPUTER APPLICATION

RATHINAM COLLEGE OF ARTS AND SCIENCE

(AUTONOMOUS)

COIMBATORE - 641021 (INDIA)

MAY - 2023

RATHINAM COLLEGE OF ARTS AND SCIENCE
(AUTONOMOUS)
COIMBATORE - 641021

BONAFIDE CERTIFICATE

This is to certify that the thesis entitled **Intrusion detection and prevention system using zeek** submitted by **SanjayKumar (RCAS2021MDB009)**, for the award of the Degree of Master of Computer Science specialization in **"DATA SCIENCE AND BUSINESS ANALYSIS"** is a bonafide record of the work carried out by him under my guidance and supervision at Rathinam College of Arts and Science, Coimbatore

Dr.MOHAMED MALLICK
Supervisor

SIVAPRAKASH P
Mentor

-

Submitted for the university examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

RATHINAM COLLEGE OF ARTS AND SCIENCE
(AUTONOMOUS)
COIMBATORE - 641021

DECLARATION

I, **Sanjaykumar (RCAS2021MDB009)**, hereby declare that this thesis entitled **"SATELLITE IMAGE CLASSIFICATION USING DEEP LEARNING ",** is the record of the original work done by me under the guidance of **Dr.Mohamed Mallick**, Faculty Rathinam college of arts and science, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any University.

Place:Coimbatore **Sanjaykumar**

Date: **Signature of the Student**

COUNTERSIGNED

Dr.Mohamed Mallick

Supervisor

Contents

Acknowledgement

On successful completion for project look back to thank who made in possible. First and foremost, thank **"THE ALMIGHTY"** for this blessing on us without which we could have not successfully our project.I am extremely grateful to **Dr.Madan.A. Sendhil, M.S., Ph.D.,** Chairman, Rathinam Group of Institutions, Coimbatore and **Dr. R.Manickam MCA., M.Phil., Ph.D.,** Secretary, Rathinam Group of Institutions, Coimbatore for giving me opportunity to study in this college.

I am extremely grateful to **Dr.R.Muralidharan, M.Sc., M.Phil., M.C.A., Ph.D.,** Principal Rathinam College of Arts and Science(Autonomous), Coimbatore.

Extend deep sense of valuation to **Mr.A.Uthiramoorthy, M.C.A., M.Phil., (Ph.D),** Rathinam College of Arts and Science (Autonomous) who has permitted to undergo the project.

Unequally I thank **Mr.P.Sivaprakash, M.E., (Ph.D).,** Mentor and **Dr.Mohamed Mallick, M.E., (Ph.D).,** Project Coordinator, and all the Faculty members of the Department - iNurture Education Solution pvt ltd for their constructive suggestions, advice during the course of study.

I convey special thanks, to the supervisor **Dr.Mohamed Mallick** who offered their inestimable support, guidance, valuable suggestion, motivations, helps given for the completion of the project. I dedicated sincere respect to my parents for their moral motivation in completing the project.

List of Figures

List of Abbreviations

SSD	Single Shot Detector
CNN	Convolutional Neural Network
API	Application programming interface
R-CNN	Region Based Convolutional Neural Networks
YOLO	You Look Only Once
RPN	Region Proposel Network
GPU	Graphics Processing Unit
CPU	Central Processing Unit
IOT	Internet of Things
PYTESST	Python Tesseract
RESNET	Residual Neural Network
I-CNN	Improved Convolutional Neural Network
GPU	Graphical User Interface

Abstract

As the number of credit card transactions keep growing and represent an increasing share of the European payment system. Leading to several stolen account numbers and subsequent losses to banks, Also people believed that credit card transaction fraud is a growing threat with severe implications for the financial industry. Data mining plays a crucial role in detecting credit card fraud in both online and offline transactions. Credit card fraud detection which is a data mining problem becomes challenging for two main reasons. First, the characteristics of normal and fraudulent behaviour are continually changing, and second, the credit card fraud dataset is highly asymmetric. The performance of fraud detection in credit card transactions is greatly affected by the sampling method of the dataset and the choice of variables and the detection techniques used. This paper investigates the performance of logistic regression(LR), k-nearest-neighbour(KNN), Support vector machine(SVM)on credit card fraud data. Federated learning based transfer learning approach is applied on the best performing machine laerning algorithm. The dataset of credit card transactions obtained from European cardholders containing 284,807 transactions. A mixture of under-sampling and oversampling techniques applied to the unbalanced data. The three strategies used to the raw and pre- processed data, respectively. This work implemented in Python. The performance of the methods was evaluated based on accuracy, sensitivity, specificity, precision and recall rate. The results are shown in comparison. After cross-validation, the accuracy of the best classification algorithm model on transfer learning in FL + SVM, FL+ logistic regression 99.00% which using the over-sampling model.

Chapter 1

Introduction

This chapter provides an overview of information about the current deforestation trends in the Amazon rainforest. The past and current technological developments for mapping deforestation are also discussed.

1.1 1.1 BACKGROUND AND MOTIVATION

1.1.1 Deforestation in Amazon

The tropical forests have been cleared at an alarming rate (Hoang and Kanemoto, 2021). Loss of natural vegetation through deforestation is the second largest source of anthropogenic greenhouse gas emissions. The consequences of such phenomena include the loss of biodiversity, changes in hydrological cycles, local and global climate change, and disruption of rights and livelihood of local communities (D'Almeida et al., 2007; Giam, 2017; Hoang and Kanemoto, 2021; Houghton, 1999). Economic development, population growth, and international trade are primarily responsible for global deforestation, driven by commodity production, forestry, agriculture, and urbanisation (Rodrigues et al., 2009). The Amazon rainforest in Brazil encompasses the most extensive stretch

of tropical forest in the world (spanning 6.7 million km2, double the size of India). It is biologically the wealthiest region of our planet, hosting 25% of global biodiversity (Malhi et al., 2009; Nicolau et al., 2021). Despite the target set by National Policy on Climate Change (NPCC) to reduce deforestation in the Amazon, there has been an increasing trend in the rate of deforestation. The year 2019 saw an increase of 34% in deforestation, equating to an alarming rate of 10,129 km2 of clear-cut deforestation. The Brazilian Amazon Deforestation Monitoring Program (PRODES) reported the area of accumulated deforestation for 2020 to be 10,851 km2. This amount was 176% higher than the established NPCC target of 3925km2 (Silva Junior et al., 2021). It corresponds to 648 TgCO2 (648 million tons of CO2) released into the atmosphere due to deforestation (Silva Junior et al., 2021).

Furthermore, PRODES estimated deforestation in 2021 to be 13,235km2 based on 45% of the monitored area. Hence, this continued large-scale deforestation of the amazon would cause perpetual damage to the functioning and diversity of the biosphere. As deforestation increases rapidly (Werth, 2002), it is crucial to map it accurately and rapidly for managing tropical rainforests and undertaking effective containment policies (Maretto et al., 2021). In addition, it helps in monitoring deforestation and understanding its implications on local and global climate and the decline in global biodiversity (Cabral et al., 2018; de Bem et al., 2020; Werth, 2002). On a global scale, these maps help achieve the target 15.2 of sustainable development goal (SDG) number 15, which aims to promote the sustainable management of all types of forests, stop deforestation, restore the degraded forest, and significantly increase global reforestation

and afforestation.

1.1.2 Mapping Deforestation Using Satellite Imagery

Remote sensing technologies have played a significant role in recent decades by providing consistent, accurate, and timely information to study our planet (Cremer et al., 2020; Maretto, 2020). Land Use and Land Cover (LULC) change detection is one of the main uses of satellite remote sensing data (Syrris et al., 2019; Treitz, 2004). It consists of analysing and quantifying the state of an object at different times (Singh, 1989) and is an essential step in understanding deforestation processes. However, traditional manual analyses to study deforestation from the imagery are expensive for complex, large-scale analysis. So, producing an accurate, automated, fast, and responsive deforestation detection system with a reasonable accuracy has been an open challenge in the remote sensing community (Ball et al., 2017; Camps-Valls et al., 2014; Lu and Weng, 2007; Syrris et al., 2019). The presence of artefacts from cloud and cloud shadows, signal inconsistency due to varying environmental conditions, and phenological changes are a few challenges that may hinder mapping a LULC change phenomenon (Liu et al., 2020; Nguyen et al., 2020).

While optical satellite data is widely used in LULC mapping (Sefrin et al., 2020; Wang et al., 2020; Yin et al., 2018), Synthetic Aperture Radar (SAR) data is gaining popularity as data from SAR sensors become available freely. SAR sensors have enabled the ability to acquire images regardless of weather conditions. As part of the Copernicus program of the European Space Agency (ESA), the Sentinel-1 (S-1) satellite

with its C-band SAR provides a revisit frequency of six days. In the interferometric wide (IW) swath mode, Nominal land acquisition provides a spatial resolution of 5 m $\times$ 20 m in dual-polarization channels in the form of phase and amplitude information. The free, full, and open data policy enables users to access extensive scale data with rich source information. These open large-scale geodata represent a huge opportunity to create an advanced innovative methodology for different LULC mapping like deforestation. However, there are only a few reliable and automated methods for detecting deforestation using these big geodata with SAR images.

Several studies have looked at the viability of SAR imagery for LULC mapping, with an emphasis on polarimetric multitemporal (Bruzzone et al., 2004) and multi-frequency SAR in the L-band, C-band, and X- band (Lonnqvist et al., 2010; Waske and Braun, 2009), as well as the combining the use of SAR and optical data (Ullmann et al., 2014). The ability of a longer wavelength (L-band) to penetrate deeper into the forest structure is more appropriate for mapping forest cover. Unfortunately, there is no free SAR. L-Band time series dataset available worldwide.

1.1.3 Deep Learning for Image Classification

In recent years Deep Learning (DL) based models have shown a remarkable feature representation capability in various fields, including image scene classification from remote sensing satellite images (Cheng et al., 2017, 2016; Hu et al., 2015; Nogueira et al., 2017; Yao et al., 2016; Zou et al., 2015). DL refers to a set of Artificial Neural Networks (ANNs) with the ability to learn a hierarchical representation of data for

image classification, object detection, and many other applications (Lecun et al., 2015). DL models, especially Convolutional Neural Networks (CNNs), have achieved state-of-the-art results in the domain of remote sensing image classification (Yanfei Liu et al., 2018; Yu and Liu, 2018) and are most commonly utilised for pattern recognition from images (O'Shea and Nash, 2015).

CNNs are composed of a series of processing layers that perform three major tasks: 2D convolutions, unit- wise nonlinear activations, and spatial pooling with subsampling (Persello and Stein, 2017). Weights and biases of the convolution operations are learnt in a supervised way to reduce classification error. Standard architectures employ a sequence of convolutional layers that are flattened into a one-dimensional vector and fed to fully-connected layers. The Convolutional layers learn the spatial features, whereas fully- connected layers learn the classification rule that will be applied to the retrieved feature vector (Lecun et al., 2015). Because the network is trained from beginning to end, feature extraction and classification occur in the same framework. This method has been shown to be effective in various computer vision tasks, especially in image classification or object detection, where one label is assigned to the entire input scene. Deep CNNs have been successfully applied to image categorisation benchmarks, considerably outperforming techniques based on hand-designed features.

Furthermore, CNNs have been modified to perform pixel-wise classification, also known as semantic segmentation. The traditional patch-based method involves training the CNN to label the centre pixel of patches derived from the input picture (Bergado et al., 2016). Nevertheless, if applied to classify a large RS

image, this method will result in redundant processing and incur high computational costs. To overcome this computational issue, Fully Convolutional Networks (FCNs) (Shelhamer et al., 2014) are trained to infer the pixel-wise classification of an entire image or patch at once. In an FCN, the fully connected layers from CNNs are replaced with one or more upsampling layers that resample the feature map extracted by convolutional layers to the exact resolution of the input image (Badrinarayanan et al., 2017; Noh et al., 2015; Shelhamer et al., 2014).

The combination of diverse types of sensors like SAR and optical provides complementary information for the same target (Adrian et al., 2021). SAR sensors capture more of the structural properties from the backscatter energy of an object on the ground. However, they are more complex for interpretation due to the presence of speckle noise. On the other hand, the optical image has better spatial resolution and is easier to interpret but widely affected by atmospheric effects. A reliable approach is required to extract and fuse information from these two sensors. DL techniques have the potential to efficiently combine information from these two sensors because it has the advantage of automatically learning the hierarchical representation from the different modality of SAR and optical image. (Ramachandram and Taylor, 2017). It has gained a foothold and continues to gain rapid advancements in the field of human activity recognition Ebrahimi Kahou et al. (2015); Neverova et al. (2014); Radu et al. (2016); medical applications (Kiros et al., 2014; Tajbakhsh et al., 2017; Wu et al., 2013), and autonomous systems (Gu et al., 2016; Lenz et al., 2013). However, DL techniques have yet to be substantially investigated to fuse multimodal data from SAR and optical remote sensing

sensors.

Attention mechanisms, like many other in DLbased methods, attempt to emulate how the human brain or eye processes data (Ghaffarian et al., 2021). The human visual system does not perceive the entire image simultaneously; instead, it focuses on specific parts. The focused part of the image is perceived as in "higher resolution", whereas the part out of focus is "low-resolution" (Ghaffarian et al., 2021). The main idea behind the attention mechanism is to give higher weights to the most relevant information in the network. Inspired by this process, Bahdanau et al., (2014) developed an attention mechanism for natural language processing. Gradually, attention mechanisms have also been successfully applied to semantic segmentation tasks (Khanh et al., 2020; Oktay et al., 2018; Roy et al., 2019, 2018; Vahadane et al., 2021; Zhao et al., 2020; Zhou et al., 2020). In the case of convolutional networks, a spatial attention mechanism focuses on the local region from a given set of feature maps (Woo et al., 2018). It produces a rich representation of the relevant features of interest from the local domains and cut out the irrelevant information or noises (Zhang et al., 2019).

Chapter 2

Literature Survey

2.0.1 Image Fusion

Image fusion combines information from two or more images from the same or different sensors of different wavelengths of the same scene (Wang et al., 2005). Table 1.1 shows the commonly used existing image fusion techniques and their limitations.

Approach	Description	Limitation	Sample Literature
Simple average	Most basic approach for pixel- level image fusion.	No guarantee of an improved image.	(Malviya and Bhirud, 2009)
Simple Maximum	Compared to the average approach, results in a highly	Influenced by the blurring effect, which	(Malviya and Bhirud, 2009;
PCA	PCA is a tool that transforms the number of correlated variables into the number of uncorrelated variables, which is helpful in image fusion.	Strong correlation between the input image is required, and fused image will have lesser quality than any of the input images.	(Abdikan, 2018; Sun et al., 2005; Walker et al., 2010)
DWT	The DWT fusion method may surpass PCA in reducing spectral distortion. Has a higher signal-to-noise ratio than pixel-based methods.	Output image has a lower spatial resolution.	(Desale and Verma, 2013)
Combined DWT & PCA	Multilevel fusion yields better results when the image is fused twice using an efficient fusion technique. The final image had a high spatial resolution and high spectral quality.	Complex method. For a better result, a good fusing technique is required.	(Pajares and de la Cruz, 2004)

Table 2.1: Overview of various image fusion approaches

2.0.2 Deep Learning-based Image Fusion

Multiple DL-based fusion approaches have been proposed for image fusion in different application fields. However, the multimodal fusion of SAR and optical images is still an evolving research field. Table 1.2 shows a few approaches that utilise DL-based

networks for the fusion of multimodal images from various sources.

Approach	Description	Sample Literature	Sample Literature
CNN based fusion	Adopts a Siamese-based CNN to fuse images from different modalities. Fuse images in a multi-scale manner via image pyramids.	(Liu et al., 2017)	(Malviya and Bhirud, 2009)
DenseFuse	DL-based multimodal fusion of infrared and visible images with encoder, decoder and a fusion block.	(Li and Wu, 2019)	(Malviya and Bhirud, 2009;
PMGI	Fast unified fusion network based on proportional maintenance of gradient and intensity (PMGI). Can handle various tasks like medical image fusion, visible and infrared image fusion, multi-exposure image fusion and pan-sharpening.	(H. Zhang et al., 2020)	(Abdikan, 2018; Sun et al., 2005; Walker et al., 2010)
U2Fusion	Fusion of medical images based on an end-to-end unsupervised fusion network. Uses an information preservation degree of the extracted feature to evaluate the importance of each source image.	(Xu et al., 2022)	(Desale and Verma, 2013)
cGAN	Conditional Generative Adversarial— Network(cGAN) based approach for fusion of multimodal SAR and optical imagery to synthesise cloud-free optical images.	(Bermudez et al., 2019; Y. Li et al., 2020)	(Pajares and de la Cruz, 2004)

Table 2.2: Overview of DL-based image fusion approaches

2.0.3 Attention Mechanisms

Attention mechanisms were initially introduced for natural language processing (Bahdanau et al., 2014). Nevertheless, it has widely been used in various fields since its introduction, especially for medical image segmentation. However, limited research utilises attention mechanisms in remote sensing applications. Moreover, according to our best knowledge, only a few researches use spatial attention mechanisms to fuse SAR and optical images and potentially substitute the cloudy optical images with SAR images automatically. Table 1.3 shows the literature related to our study incorporating attention mechanisms.

2.1 PROBLEM ANALYSIS

Most researches on mapping deforestation still rely heavily on the ability of an optical sensor to capture the phenomenon. Optical sensors have a huge advantage in terms of the interpretability of the images. However, as discussed in section 1.1.2, there is a

Approach	Description	Reference	Sample Literature
AttentionU- Net	Uses a soft-attention gate inside a generic U-Net to produce attention maps that emphasise the location of the pancreas for medical image segmentation.	(Oktay et al., 2018)	(Malviya and Bhirud, 2009)
CBAM	Convolutional Block Attention Module (CBAM) uses a channel and spatial attention module in a feed-forward CNN to refine the encoder features.	(Woo et al., 2018)	(Malviya and Bhirud, 2009;
SCAU-Net	Enhances U-Net encoder and decoder framework with spatial and channel attention modules for medical image segmentation.	(Khanh et al., 2020; Zhao et al., 2020)	(Abdikan, 2018; Sun et al., 2005; Walker et al., 2010)
SCAttNet	Uses an end-to-end semantic segmentation network with a lightweight channel and spatial attention module for feature refinement in high-resolution remote sensing images.	(H. Li et al., 2021)	(Desale and Verma, 2013)
TAFFN.	Triplet Attention Feature Fusion Network (TAFFN) for the fusion of SAR and optical image. Uses spatial, channel and cross attention based on a self-attention mechanism to extract and integrate long-range and complementary information from the images and perform a land cover classification.	(Xu et al., 2021)	(Pajares and de la Cruz, 2004)

Table 2.3: Overview of literature related to attention mechanisms

limitation to using optical remote sensors to map deforestation due to their inability to penetrate the cloud. The presence of persistent clouds covering the Amazon rainforest season-wide hinders the ability to detect deforestation, especially during the wet season, when it is nearly impossible to get a cloud-free image over some regions (Griffiths et al., 2018).With the advancement of SAR sensors and their ability to penetrate the clouds, monitoring the rainforest even during seasons with persistent cloud covers has presented an alternative way to monitor deforestation. However, there is always a trade-off for only using the SAR sensors to study deforestation. SAR images are complex to interpret and have a different modality than optical sensors. Therefore, in recent years, a substantial amount of research has focused on optimally fusing complementary and correlated data of multimodal sensors. Various attempts to fuse SAR and optical data include the wavelet-merging technique (Abdikan, 2018; Hong and Zhang, 2008; Lu et al., 2011), Principal Component Analysis (PCA) (Abdikan, 2018; Pereira et al., 2013; Walker et al., 2010) and intensity-hue-saturation (IHS) (Abdikan, 2018). However, these approaches fuse the image at a pixel level, which suffers from spectral distortion

and fails to maintain the spatial resolution of input images (Yu Liu et al., 2018).

Chapter 3

Dataset description

The Kaggle dataset that we use to train our CNN model comes from satellite imagery of the Amazon Basin that wascollected over a 1-year span starting in 2016. The training set and test set consist of 40,479 and 61,191 256×256 images respectively. Each image is available both in a 3-channel (RGB) JPEG format and in a 4-channel (RGB+near-IR) TIFF format. There are 17 classes and each image can belong to multiple classes. The labels can broadly be broken into three groups: atmospheric conditions, common land cover/land use phenomena, and rare land cover/land use phenomena. The atmospheric condition labels are: clear, cloudy, partly cloudy, and haze. The common labels are: primary, agriculture, cultivation, habitation, water and roads. The rare labels are: slash-and-burn, selective logging, blooming, bare ground, conventional mining, artisinal mining, and blow-down. Distribution of training images across different class labels. It is evident that one of the more prominent challenges is dealing with the class data imbalance–that is, some of the classes have very few training images associated with it. Upon closer inspection, we see that there is a structure in the labels and the inter-relationships among labels can be exploited to design better classifiers and dur-

ing postprocessing to weed out some misclassifications. The co-occurrence matrix for weather labels. We see that the weather labels are mutually exclusive. Figure 4 and 5 show the co-occurrence matrices for the common labels and rare labels respectively, where it is evident that common labels have heavy overlap while rare labels have very minimal overlap.

Figure 3.1: Overview of the dataset

The class labels for this task were chosen in collaboration with Planet's Impact team and represent a reasonable subset of phenomena of interest in the Amazon basin. The labels can broadly be broken into three groups: atmospheric conditions, common land cover/land use phenomena, and rare land cover/land use phenomena. Each chip will have one and potentially more than one atmospheric label and zero or more common and rare labels. Chips that are labeled as cloudy should have no other labels, but there may be labeling errors.

3.1 Data analysis

Some basic data analysis was performed on the dataset which have been described in details below.

3.1.1 Distribution of Training Labels

Firstly, the histogram as present in Figure showing the distribution of training labels was constructed. It has been found that the dataset is not balanced in nature, i.e. , all labels are not present in uniform quantity. Labels such as primary, clear and agriculture are present in significantly more number than the other ones. Whereas, some other labels like slash burn, blow down and conventional mine are present in very less quantity. Note that in the dataset, a single image may have multiple classes. The histogram must be seen keeping this in mind

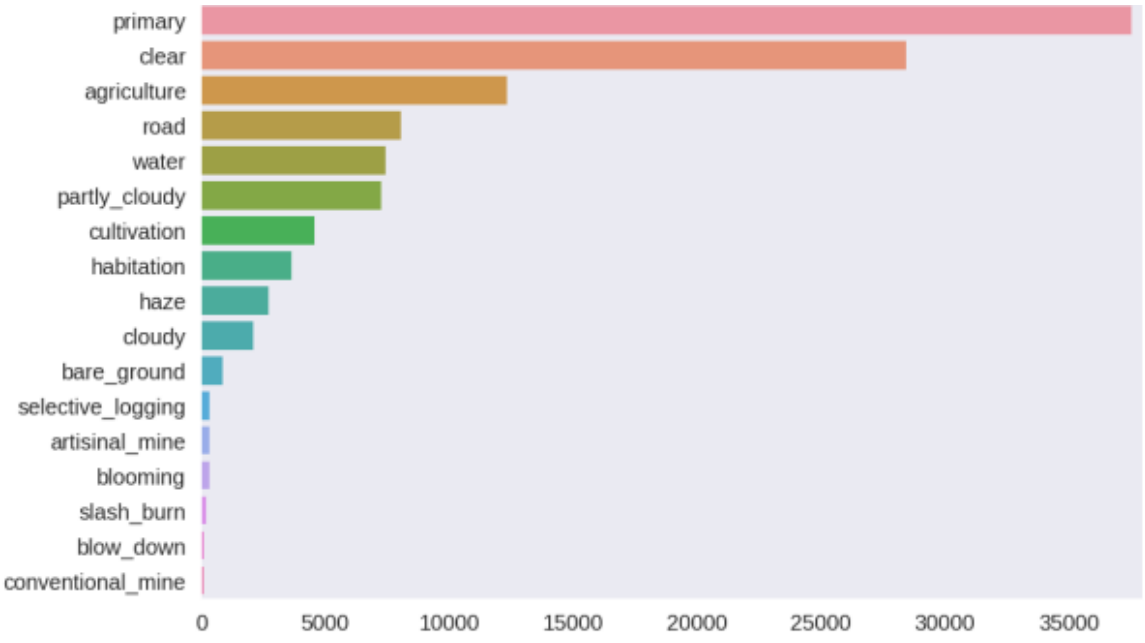

Figure 3.2: Distribution of Training Labels

3.1.2 Correlation Matrix

The correlation matrix was plotted, as shown in Figure 5, to understand the occurrence of the classes with respect to each other. Here, redder is the label, more is the value of the correlation for any given pair of classes. After studying this plot, some interesting results were observed. Some of them are:

• The label primary is associated with almost all classes. This means that most chips have some degree of primary forests along with other labels.

• The label agriculture is also associated with a few labels like road, habitation and cultivation.

3.1.3 Preprocessing of data

Even after converting to JPG, the dataset was quite large in size. It would have been computationally expensive to train the model on such a large dataset. Besides, the obtained dataset contained images of various dimensions. Hence, all images were resized to a standard size, in this case, 128x128 pixels. This is also an important step as it helps in speeding up the training. Since the downloaded VGG16 model did not contain the top layer, it was possible to train with images with dimensions (128x128x3) that were different from the dimensions of images used in the original VGG16 model (224x224x3). In this dataset, 40479 images for training and 40669 images for testing were used. Each image may be classified into multiple classes.

Chapter 4

Methodology

Deep learning is actually a subset of machine learning. It technically is machine learning and functions in the same way but it has different capabilities. The main difference between deep and machine learning is, machine learning models become well progressively but the model still needs some guidance.A branch of machine learning called deep learning appeared. The popularity of machine learning and the development of the computing capacity of computers enabled this new technology. Deep learning as a concept is very similar to machine learning but uses different algorithms. While machine learning works with regression algorithms or decision trees, deep learning uses neural networks that function very similarly to the biological neural connections of our brain.

If a machine learning model returns an inaccurate prediction then the programmer needs to fix that problem explicitly but in the case of deep learning, the model does it by him.The first advantage of deep learning over machine learning is the needlessness of the so-called feature extraction. Automatic car driving system is a good example of deep learning.Deep learning that enable the network to learn from unsupervised data

and solve complex problems. Deep Learning approaches such as Convolutional Neural Network, Auto Encoder, Deep Belief Network, Recurrent Neural Network, Generative Adversal Network and Deep Reinforcement Learning are the algorithms used in Deep Learning.In our project we are using Convolutional Neural Networks.

4.1 Convolution Neural Network

One of the most popular deep neural networks is Convolutional Neural Networks. It is a class of deep neural networks, most commonly applied to analyze visual imagery and specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data.Convolutional neural networks are composed of multiple layers of artificial neurons.The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision.It can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. CNN utilizes spatial correlations which exist with the input data. Each concurrent layer of the neural network connects some input neurons.

4.1.1 Convolution Neural Network in image processing

A convolutional neural network is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.CNNs are fully connected feed forward neural networks. CNNs are very effective in reducing the number of parameters without losing on the quality of models. Images have high dimensionality (as each pixel is considered as a feature) which suits the above described abilities of CNNs.It as a machine learning algorithm that can take in an input image, assign importance weights and biases to various objects in the image, and then can differentiate one from the another.It works by extraction features from the image. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

We use Convolutional Neural Network and Deep Learning based yolo for Real Time Detection and Recognition of Human Faces, which is simple face detection and recognition system is proposed in this paper which has the capability to recognize human

faces in single as well as multiple face images in a database in real time with masks on or off the face. Pre-processing of the proposed frame work includes noise removal and hole filling in colour images. After pre-processing, face detection is performed by using CNNs architecture. Architecture layers of CNN are created using Keras Library in Python. Detected faces are augmented to make computation fast. By using Principal Analysis Component (PCA) features are extracted from the augmented image. For feature selection, we use Sobel Edge Detector.

4.1.2 The Input Image

Real-time input images are used in this proposed system. Face of person in input images must be fully or partially covered as they have masks on it. The system requires a reasonable number of pixels and an acceptable amount of brightness for processing. Based on experimental evidence, it is supposed to perform well indoors as well as outdoors i.e. passport offices, hospitals, hotels, police stations and schools etc.

4.1.3 The Pre-processing Stage

Input image dataset must be loaded as Python data structures for pre-processing to overturn the noise disturbances, enhance some relevant features, and for further analysis of the trained model. Input image needs to be pre-processed before face detection and matching techniques are applied. Thus pre-processing comprises noise removal, eye and mask detection, and hole filling techniques. Noise removal and hole filling help eliminate false detection of face/ faces. After the pre-processing, the face image is cropped and re-localised. Histogram Normalisation is done to improve the quality of

the pre- processed image.

4.1.4 Image analysis Stage

We perform face detection usingHAAR Cascade algorithm.This system consists of the value of all black pixels in greyscale images was accumulated. They then deducted from the total number of white boxes. Finally, the outcome is compared to the given threshold, and if the criterion is met, the function considers it a hit.In general, for each computation in Haar-feature, each single pixel in the feature areas can need to be obtained, and this step can be avoided by using integral images in which the value of each pixel is equal to the number of grey values above and left in the image.

4.1.5 The Feature-Extraction Stage

Feature Extraction improves model accuracy by extracting features from pre-processed images and translating them to a lower dimension without sacrificing image characteristics. This stage allows for the classification of human faces.

4.1.6 The Classification Stage

Principal Component Analysis(PCA) is used to classify faces after an image recognition model has been trained to identify face images. Identifying variations in human faces is not always apparent, but PCA comes into the picture and proves to be the ideal procedure for dealing with the problem of face recognition. PCA does not operate classifying face images based on geometrical attributes, but rather checks which all factors would influence the faces in an image. PCA was widely used in the field of

pattern recognition for classification problems.PCA demonstrates its strength in terms of data reduction and perception.

4.1.7 Training Stage

The method is based on the notion that it learns from pre- processed face images and utilizes CNN model to construct a framework to classify images based on which group it belongs to. This qualified model is saved and used in the prediction section later. In CNN model, the stages of feature extraction are done by PCA and feature selection done by Sobel Edge Detector and thus it improves classification efficiency andaccuracy of the training model.

4.1.8 Prediction Stage

In this stage, the saved model automatically detects theoftheface maskimagecaptured by the webcam or camera. The saved model and the pre-processed images are loaded for predicting the person behind the mask. CNN offers high accuracy over face detection, classification and recognition produces precise and exactresults.CNN model follows a sequential model along with Keras Library in Python for prediction of human faces.

4.2 Stochastic Gradient Descent

Although any method can be used to train the optimization convolutional networks, one of the most common is the stochastic gradient descent using the mini-batch of samples.

The gradient calculation requires the error of the last layer to previous layers to be backpropagated. The backpropagation of errors [44,45] is a method of calculating

gradients that can be used in the method of stochastic gradient descent to train neural networks grouped in layers. This is really a simple implementation of the chain rule of derivatives, speeding the calculations of all required partial derivatives. As mentioned, once a pattern has been applied to the input of the network as a stimulus, this propagates from the first layer through the upper layers of the network, to generate an output. The output signal is compared to the desired output and an error signal for each of the outputs (error vector) is calculated.

The error outputs are propagated backwards from the output layer to all neurons in the hidden layer contributing directly to the output. This process is repeated layer by layer, until all neurons in the network have received an error signal describing their relative contribution to the total error. There are many optimizations of this method, such as Momentum, Adagrad, RMSProp, Adam, Nesterov, Adadelta, etc. In our case, we use the first-mentioned, incorporating the term known as momentum (which can be understood as the average of the previous gradients), which reduces oscillations that cause local minima, thus accelerating convergence. There are other optimizations, such as weight decay (regularization term), which penalizes changes in the weights and prevents them from being too large.

The importance of this process is that, as the network is trained, the neurons in the intermediate layers organize themselves in such a way that the different neurons learn to recognize different characteristics of the total input space. After training, when these neurons are presented with an arbitrary input pattern that contains noise or is incomplete, the neurons in the hidden layers of the network will respond with an active

output if the new input contains a pattern that resembles the feature that the Individual neurons have learned to recognize during their training.

4.3 Hyperparameter Optimization

In both cases (full training or fine-tuning), the training of these networks requires the adjustment of certain variables called hyperparameters (momentum, weight decay, learning rate, etc.), specifically in the context of algorithms based on stochastic gradient descent (which are the most common). To optimize this setting, it is interesting to consult [48]. The hyperparameters that are usually considered in the first place are: the initial learning rate, its decay value and the intensity of regularization, but there are many others that can also be important, such as the momentum, the decay of the weights, the number of iterations, etc.

Regarding the hyperparameters themselves, we can say the following. Learning rate: This is one of the most important, if not critical, hyperparameters, as it determines the amplitude of the jump to be made by the optimization technique in each iteration. If the rate is very low it will take a long time to reach convergence and if it is very high it could fluctuate around the minimum or even diverge. The asymptotic convergence rates of SGD are independent of sample size. Therefore, the best way to determine the correct learning rates is to perform experiments using a small but representative sample of the training set. When the algorithm works well with that small set of data, the same learning rates can be maintained and trained with the complete dataset [49]. Another possible option is to use dynamic learning rates (which are reduced when

converging to the solution). This dynamic must be predefined and must therefore be adapted to the specific characteristics of each dataset. Momentum: As the parameters approach a local optimum, improvements can slow down, taking a long time to finally reach the minimum. Introducing a term that "boosts" the optimization technique can help to further improve model parameters towards the end of the optimization process. This term, called momentum, will consider how the parameters were changing in recent iterations, and will use that information to keep moving in the same direction. Specifically, the momentum term increases for dimensions whose gradients are pointing in the same directions and reduces updates for dimensions whose gradients change direction. As a result, faster convergence is achieved and oscillation is reduced. Size of the mini-batch: In our case, we use the stochastic gradient descent method with a random subset (mini-batch) of the training data at each iteration. If the size of the mini-batch is too small, convergence will be slow and it is also not possible to take advantage of some type of highly efficient operations (intelligent matrices). If the size is too large, the speed advantages offered by this method are reduced, as larger subsets of training data are used. In any case, its impact mainly affects the training time and hardly affects the results obtained. A value of 32 may be a good initial approximation. Weight decay: This value is an additional term in the weight update rule that causes the weights to drop exponentially to zero and determines the importance of this type of regularization in the gradient calculation. Generally, the more examples of training you have, the weaker this term will be and the more parameters you have to adjust (very deep nets, large filters, etc.), the higher this term should be.

Number of iterations: One way to know the number of iterations to perform (without reaching overfitting) is to extract a subset of samples from the training set (note that the test set has previously been removed from the complete dataset) and to use it in an auxiliary way during training. This subset is called the validation set. The role of the validation set is to evaluate the network error after each epoch (or after every certain number of epochs) and determine when it begins to increase. Since the validation set is left out during training, the error committed on it is a good indication of the network error over the entire test set. Consequently, the training will be stopped when this validation error increases and the values of the weights of the previous epoch will be retained. This stopping criterion is called early-stopping. Early-stopping is a simple way to avoid overfitting, i.e., even if the other hyperparameters cause overfitting, early-stopping will greatly reduce overfitting damage that would otherwise occur. It also means that it hides the excessive effect of other hyperparameters, possibly hindering the analysis that one might want to do when trying to figure out the effect of individual hyperparameters.

In addition to the criteria discussed for each hyperparameter, certain general details must be taken into account. Implementation: Larger neural networks often require a lot of training time, so tuning the hyperparameters can be very time-consuming. One option is to design a system that generates random hyperparameters (within reasonable ranges) and performs training, evaluating the performance achieved and storing model control points (along with their corresponding statistics). Subsequently, these control points can be inspected and analyzed to outline the appropriate hyperparameter

optimization strategies.

Use cross validation or not: In most cases, if the validation set is large enough, cross-validation is not required. Search intervals for hyperparameters: It is advisable to search for hyperparameters using a logarithmic scale; at least for the learning rate and for the strength of regularization, as they have multiplicative effects on the training dynamics. Random search or search by grid: Randomized trials are more efficient for hyperparameter optimization than grid-based assays. In addition, this is also generally easier to implement. Border values: A hyperparameter can sometimes be searched at an inappropriate interval. Therefore, it is important to check that the adjusted hyperparameter is not at one end of that range, since the optimum value of the hyperparameter might be is outside our search range.

Initialization of the parameters: This operation can be deceptively important. In general, we can say that bias terms can often be initialized to 0 without problems. The weight matrices are more problematic, for example, if all values are initialized to 0, the activation function may generate null gradients; if all weights were equal, the hidden units would produce the same gradients and behave the same (thus wasting parameters). A possible solution is to initialize all elements of the weight matrix following a zero-centered Gaussian distribution with a standard deviation of 0.01.

The initial learning rate is often the most important hyperparameter and therefore its correct adjustment should be ensured. Its value is usually less than 1 and greater than 106. Usually, 0.01 is used as a typical value, but this logically depends on each case. Following this methodology, the hyperparameters used in the different trainings

shown in the next section have been selected.

4.4 Full Training of a Residual Network (ResNet)

We also decided to use the original residual network developed by He et al., of Microsoft [15], which has led to a growing adoption of this specific type of network due to its good results. The depth of the networks has a decisive influence on their learning, but adjusting this parameter optimally is a very difficult task. In theory, when the number of layers in a network increases, its performance should also improve. However, in practice, this is not true for two main reasons: the vanishing gradient (many neurons become ineffective/useless during the training of such deep networks); and the optimization of parameters is highly complex (by increasing the number of layers, it increases the number of parameters to adjust, which makes training these networks very difficult, leading to higher errors than in the case of shallower networks).

The residual networks seek to increase the network's depth without such problems affecting the results. The central idea of residual networks is based on the introduction of an identity function between layers. In conventional networks, there is a nonlinear function y = H (x) between layers (underlying mapping), as shown on the left of Figure 5. In residual networks, we have a new nonlinear function y = F (x) + id (x) = F (x) + x, here F (x) is the residual (on the right of Figure 5). This modification (called shortcut connections) allows important information to be carried from the previous layer to the next layers. Doing this avoids the problem of the vanishing gradient.

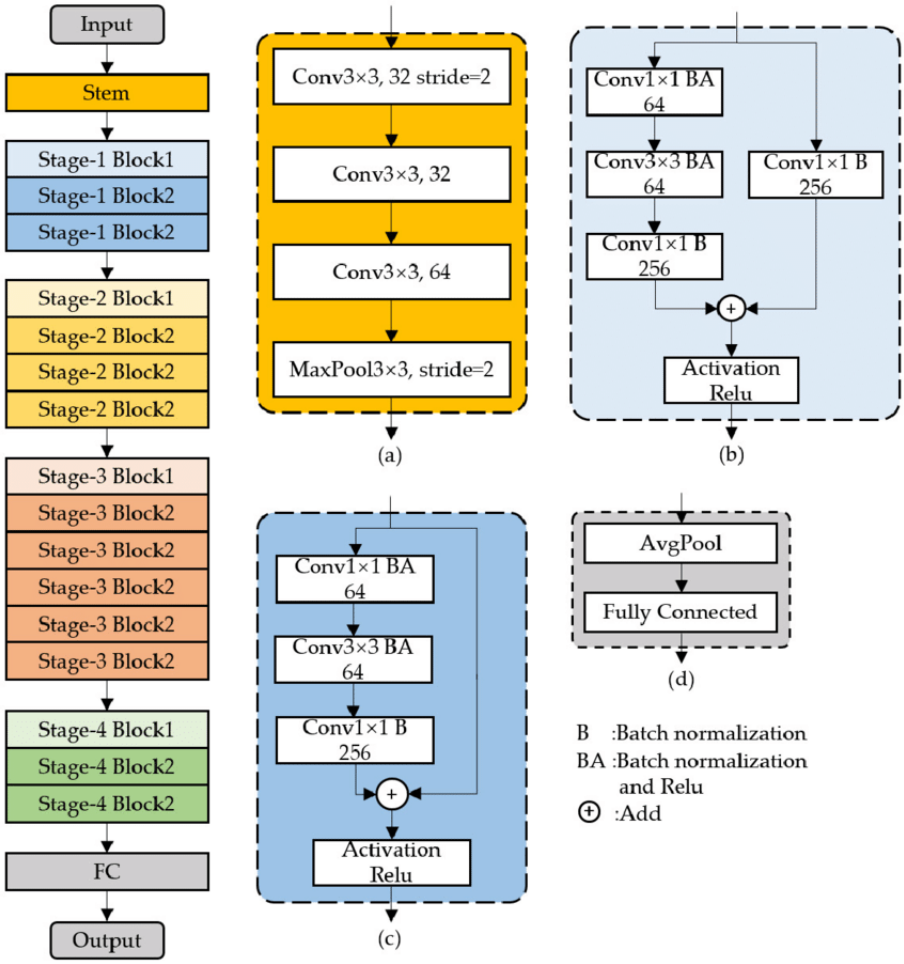

Figure 4.1: Resnet Architecture

Chapter 5

Result and Discussion

In this section the experimental setting is introduced first, to establish the basic idea of the work

5.1 MODEL PERFORMANCE METRICS

5.1.1 Overall Accuracy

Model overall accuracy calculated using Eq. (5.1) is one of the performance metrics used to evaluate the model. The overall accuracy of a classifier was used in two different scenarios. (a) During the training of the model, monitor its performance. Furthermore, (b) during the prediction phase where the adaptability of the different trained models was evaluated.

5.1.2 Precision, Recall and F1 Score

Apart from the overall accuracy, confusion matrices resulting from all the classifier's variations were also investigated. Using Eq. (3.3), we can calculate the precision and recall of each class using the confusion matrix

$$precision = tp/tp + fp \tag{5.1}$$

$$recall = tp/tp + fn \tag{5.2}$$

$$precision = tp + fn/tp + tn + fp + fn \tag{5.3}$$

where,tp is the total number of true positives, fp is the number of false positives, and fn is the number of false negatives. Precision indicates the proportion of deforestation areas correctly identified by the classifier. Recall indicates the proportion of deforested areas in the reference data correctly identified by the classifier. A harmonic mean of precision and recall parameters is used to calculate the F1 score in Eq. (5.5) (Flach and Kull, 2015)

$$F1score = precision.recall/precision + recall \tag{5.4}$$

5.2 Result

First extracted the features from the lower convolutional layer of ResNet50v2 giving us a vector of shape (7, 7, 2048) which is flattened to a shape of (49, 2048). This vector is passed through the CNN encoder,the encoder input is passed through the RNN decoder, that attends over the image to predict the next word. The used attention mechanism is based on Bahdanau's additive attention**. This frees the model from having to encode

the whole input feature into a fixed-length vector, and lets the model focus only on information relevant to the generation of the next target word.

```
Model: "cnn__encoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                multiple                  524544
=================================================================
Total params: 524,544
Trainable params: 524,544
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.1: Summary of model

```
Model: "rnn__decoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        multiple                  115200
_________________________________________________________________
gru (GRU)                    multiple                  1575936
_________________________________________________________________
dense_1 (Dense)              multiple                  262656
_________________________________________________________________
dense_2 (Dense)              multiple                  230850
_________________________________________________________________
bahdanau_attention (Bahdanau multiple                  394753
=================================================================
Total params: 2,579,395
Trainable params: 2,579,395
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.2: Summary of model

The extracted features stored in the respective .npy files are passed through the CNN

encoder.The encoder output, hidden state (initialized to 0) and the decoder input(which is the start token) is passed to the RNN decoder.The decoder returns the predictions and the decoder hidden state.The decoder hidden state is then passed back into the model and the predictions are used to calculate the loss.Teacher forcing is used, to decide the next input to the decoder Calculate the gradients, apply them to the optimizer and backpropagate.Model trained for 11 epochs achieved F-beta score 83.39% Validation loss reached 0.2962 at 11th epoch, while training loss reached 0.2924 After the first epoch the validation loss decrease was only occurring at 3rd decimal place

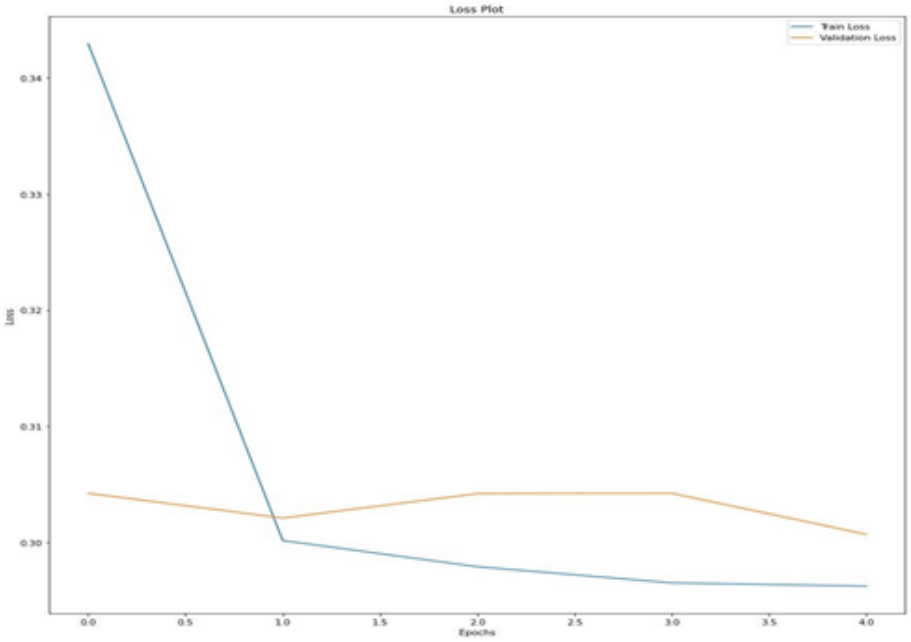

Figure 5.3: Validation graph

Model trained for 20 epochs achieved F-beta score 83.9%Validation loss reached 0.2915 at 20th epoch, while training loss reached 0.2918 After the first epoch the validation loss decrease was only occurring at 3rd decimal place

5.2.1 Testing Output

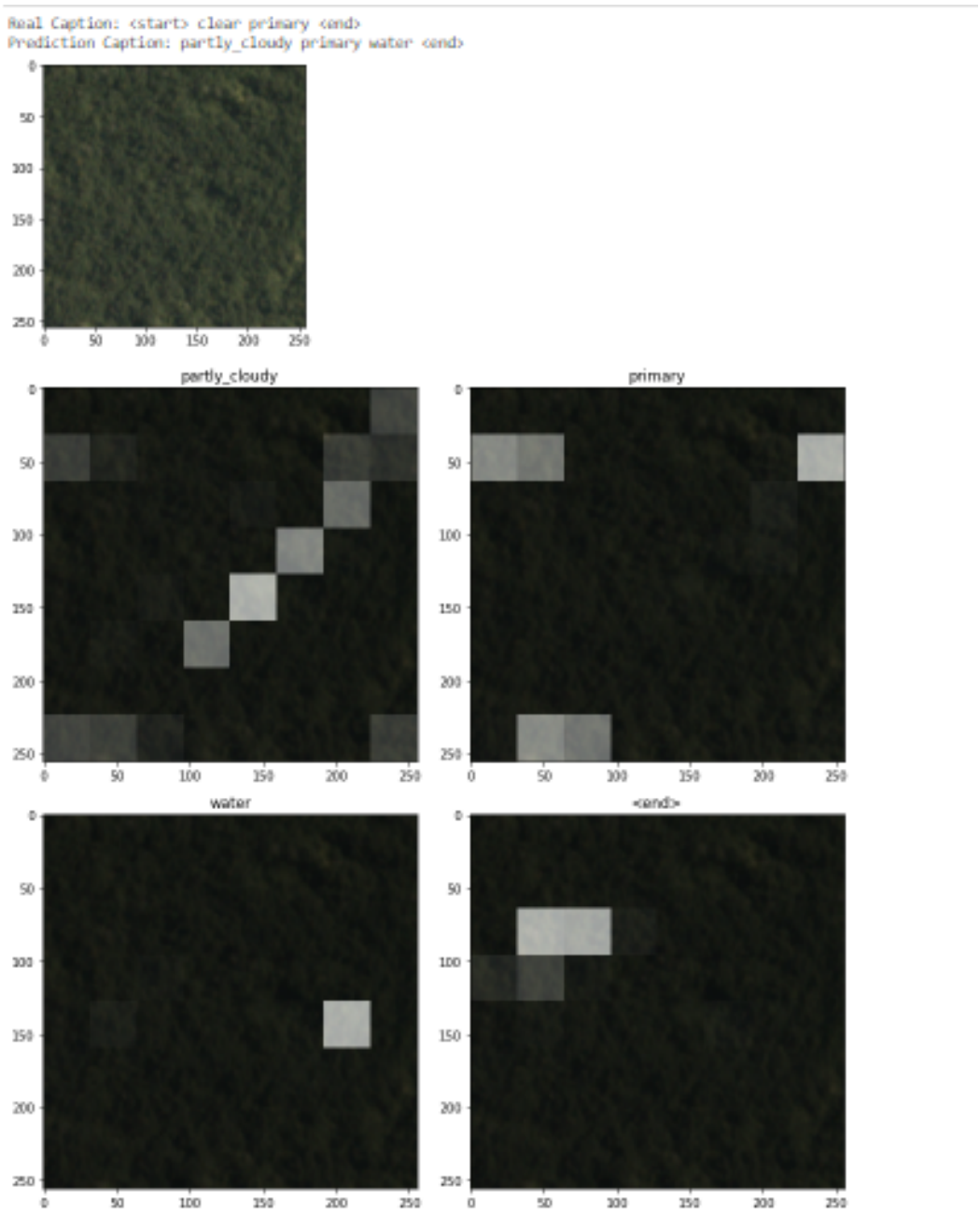

Figure 5.4: Test image

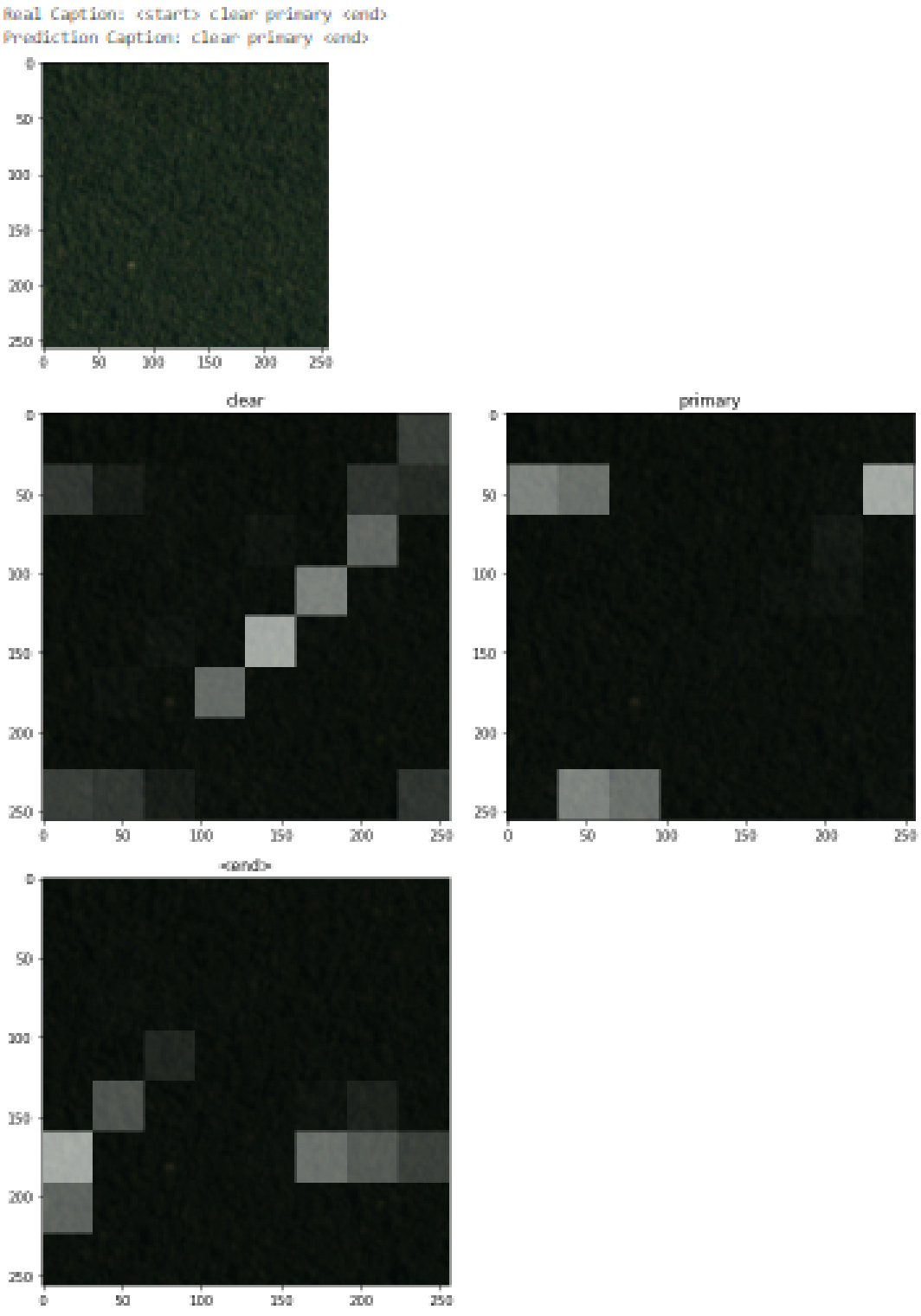

Figure 5.5: Test image

Chapter 6

Conclusion

Satellite image classification using attention mechanism is a feasible approach. The model achieved F-beta score greater than 83% (highest F-beta score from the respective Kaggle competition around 93%).An approach that utilizes the TIF version would probably provide better results, however it would be resource demanding.We used the ResNet pretrained at the ImageNet dataset, as feature extractor. Probably, fine-tuning the model to the satellite image dataset before proceeding with the feature extraction would provide better results. Model training took place inside a GPU enabled Colab notebook (limitations at resources). Future improvements include selecting a different pre-trained network as feature extractor, modifying model architecture by replacing the GRU layer with an LSTM one and/or selecting a different attention mechanism.

References

1. Abdikan, S., 2018. Exploring image fusion of ALOS/PALSAR data and LANDSAT data to differentiate forest area. Geocarto International 33, 21–37. https://doi.org/10.1080/10106049.2016.1222635

2. Adrian, J., Sagan, V., Maimaitijiang, M., 2021. Sentinel SAR-optical fusion for crop type mapping using deep learning and Google Earth Engine. ISPRS Journal of Photogrammetry and Remote Sensing 175, 215–235. https://doi.org/10.1016/J.ISPRSJPRS.2021.02.018

3. Almeida-Filho, R., Rosenqvist, A., Shimabukuro, Y.E., Silva-Gomez, R., 2007. Detecting deforestation with multitemporal L-band SAR imagery: a case study in western Brazilian Amazônia. International Journal of Remote Sensing 28, 1383–1390. https://doi.org/10.1080/01431160600754591

4. Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 2481–2495.

https://doi.org/10.1109/TPAMI.2016.2644615

5. Bahdanau, D., Cho, K.H., Bengio, Y., 2014. Neural Machine Translation by Jointly Learning to Align and Translate. 3rd International Conference on Learning

6. Representations, ICLR 2015 - Conference Track Proceedings. Ball, J.E., Anderson, D.T., Chan, C.S., 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. Journal of Applied Remote Sensing 11, 1.
https://doi.org/10.1117/1.JRS.11.042609

7. Bergado, J.R., Persello, C., Gevaert, C., 2016. A deep learning approach to the classification of sub- decimetre resolution aerial images, in: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 1516–1519.
https://doi.org/10.1109/IGARSS.2016.7729387

8 Bermudez, J.D., Happ, P.N., Feitosa, R.Q., Oliveira, D.A.B., 2019. Synthesis of Multispectral Optical Images from SAR/Optical Multitemporal Data Using Conditional Generative Adversarial Networks. IEEE Geoscience and Remote Sensing Letters 16, 1220–1224.
https://doi.org/10.1109/LGRS.2019.2894734

9 Bragagnolo, L., da Silva, R.V., Grzybowski, J.M.V., 2021. Amazon and Atlantic

Forest image datasets for semantic segmentation.
https://doi.org/10.5281/ZENODO.4498086

10 Bragagnolo, L., da Silva, R.V., Grzybowski, J.M.V., 2019. Amazon Rainforest dataset for semantic segmentation.
https://doi.org/10.5281/ZENODO.3233081

11 Bruzzone, L., Marconcini, M., Wegmüller, U., Wiesmann, A., 2004. An advanced system for the automatic classification of multitemporal SAR images. IEEE Transactions on Geoscience and Remote Sensing 42, 1321–1334.
https://doi.org/10.1109/TGRS.2004.826821

12 Cabral, A.I.R., Saito, C., Pereira, H., Laques, A.E., 2018. Deforestation pattern dynamics in protected areas of the Brazilian Legal Amazon using remote sensing data. Applied Geography 100, 101–115.
https://doi.org/10.1016/j.apgeog.2018.10.003

13 Camps-Valls, G., Tuia, D., Bruzzone, L., Benediktsson, J.A., 2014. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. IEEE Signal Processing Magazine 31, 45–54.
https://doi.org/10.1109/MSP.2013.2279179

14 Cheng, G., Han, J., Lu, X., 2017. Remote Sensing Image Scene Classification: Benchmark and State of the Art. Proceedings of the IEEE 105, 1865–1883.
https://doi.org/10.1109/JPROC.2017.2675998

15 Cheng, G., Zhou, P., Han, J., 2016. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing 54, 7405–7415. https://doi.org/10.1109/TGRS.2016.2601622

16 Cremer, F., Urbazaev, M., Cortes, J., Truckenbrodt, J., Schmullius, C., Thiel, C., 2020. Potential of Recurrence Metrics from Sentinel-1 Time Series for Deforestation Mapping. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 13, 5233–5240. https://doi.org/10.1109/JSTARS.2020.3019333

17 D'Addabbo, A., Refice, A., Pasquariello, G., Lovergine, F., 2016. SAR/optical data fusion for flood detection. International Geoscience and Remote Sensing Symposium (IGARSS) 2016-November, 7631–7634. https://doi.org/10.1109/IGARSS.2016.7730990

18 D'Almeida, C., Vörösmarty, C.J., Hurtt, G.C., Marengo, J.A., Dingman, S.L., Keim, B.D., 2007. The effects of deforestation on the hydrological cycle in Amazonia: A review on scale and resolution. International Journal of Climatology 27, 633–647. https://doi.org/10.1002/joc.1475

19 de Bem, P., de Carvalho Junior, O., Fontes Guimarães, R., Trancoso Gomes, R., 2020. Change Detection of Deforestation in the Brazilian Amazon Using Landsat Data and Convolutional Neural Networks. Remote Sensing 12, 901. https://doi.org/10.3390/rs12060901

20 Desale, R.P., Verma, S. v., 2013. Study and analysis of PCA, DCT DWT based image fusion techniques. International Conference on Signal Processing, Image Processing and Pattern Recognition 2013, ICSIPR 2013 1, 66–69. https://doi.org/10.1109/ICSIPR.2013.6497960

21 Durieux, A.M., Calef, M.T., Arko, S., Chartrand, R., Kontgis, C., Keisler, R., Warren, M.S., 2019. Monitoring forest disturbance using change detection on synthetic aperture radar imagery, in: Zelinski, M.E., Taha, T.M., Howe, J., Awwal, A.A., Iftekharuddin, K.M. (Eds.), Applications of Machine Learning. SPIE, p. 39. https://doi.org/10.1117/12.2528945

22 Ebrahimi Kahou, S., Bouthillier, X., Lamblin, P., Gulcehre, C., Michalski, V., Konda, K., Jean, S., Froumenty, P., Dauphin, Y., Boulanger-Lewandowski, N., Chandias Ferrari, R., Mirza, M., Warde- Farley, D., Courville, A., Vincent, P., Memisevic, R., Pal, C., Bengio, Y., 2015. EmoNets: Multimodal deep learning approaches for emotion recognition in video. Journal on Multimodal User Interfaces 10, 99–111. https://doi.org/10.1007/s12193-015-0195-2

23 ESA, n.d. Copernicus Sentinel data 2018-2021 for Sentinel data [WWW Document]. URL https://sentinel.esa.int/web/sentinel (accessed 12.6.21a). ESA, n.d. STEP – Science Toolbox Exploitation Platform [WWW Document]. URL http://step.esa.int/main/ (accessed 5.16.22b).

24 Flach, P., Kull, M., 2015. Precision-Recall-Gain Curves: PR Analysis Done Right,

in: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc.

25 Gao, J., Yuan, Q., Li, J., Zhang, H., Su, X., 2020. Cloud Removal with Fusion of High Resolution Optical and SAR Images Using Generative Adversarial Networks. Remote Sensing 12, 191. https://doi.org/10.3390/rs12010191

26 Ghaffarian, S., Valente, J., van der Voort, M., Tekinerdogan, B., 2021. Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review. Remote Sensing 13, 2965. https://doi.org/10.3390/RS13152965/S1

27 Giam, X., 2017. Global biodiversity loss from tropical deforestation. Proc Natl Acad Sci U S A 114, 5775–5777. https://doi.org/10.1073/pnas.1706264114

28 Gibril, M.B.A., Bakar, S.A., Yao, K., Idrees, M.O., Pradhan, B., 2017. Fusion of RADARSAT-2 and multispectral optical remote sensing data for LULC extraction in a tropical agricultural area. Geocarto International 32, 735–748. https://doi.org/10.1080/10106049.2016.1170893