UNIVERSITY OF WEST ATTICA
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τμημα Μηχανικων Πληροφορικης και Υπολογιστων

Διπλωματική Εργασία

“Εκτίμηση αποψίλωσης δασών για την ανάλυση της κλιματικής αλλαγής με χρήση μοντέλων βαθιάς μάθησης σε δορυφορικές εικόνες”

Ιωάννης Δασκαλόπουλος

επιβλέπων καθηγητής

Αθανάσιος Βουλόδημος

Επίκουρος Καθηγητής

Αθήνα, Μάρτιος 2021

UNIVERSITY OF WEST ATTICA
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Department of Informatics and Computer Engineering

MEng Thesis

"Deforestation monitoring for climate change analysis using deep learning techniques on satellite imagery"

Ioannis Daskalopoulos

supervised by

Athanasios Voulodimos
Assistant Professor

Athens, March 2021

Nikolaos Vasilas

Digitally signed by Nikolaos Vasilas
Date: 2021.03.11
18:37:38 +02'00'

....

Νικόλαος Βασιλάς
Καθηγητής

Athanasios Voulodimos

Digitally signed by Athanasios Voulodimos
Date: 2021.03.11
11:02:50 +02'00'

....

Αθανάσιος Βουλόδημος
Επίκουρος
Καθηγητής

Anastasios Kesidis

Digitally signed by Anastasios Kesidis
Date: 2021.03.11 15:58:04 +02'00'

....

Αναστάσιος Κεσίδης
Αναπληρωτής
Καθηγητής

Abstract

In the field of climate change analysis, a huge amount of information, derived from various sources and in various formats needs to be analyzed daily, for the production of accurate insights and predictions. Such a task, is heavily reliant on precise measurements and visual information. Thus, the instruments that are tasked with capturing this information are of high importance. Since climate change analysis is a wide field, the focus will be narrowed towards the detection of possible factors contributing to the phenomenon of deforestation. Furthermore, the type of information that will be processed is visual, making satellite imagery an ideal choice.

In this thesis, we first tackle the task of creating a pipeline for preprocessing said satellite imagery. The preprocessing step includes the possible transformations that will be performed on the images as well as the optimal set of bands with regards to the performance of the given model. Next, we will perform multi-label classification in an attempt to describe the content of the images in terms of the factors that contribute to the deforestation, using a set of tags. Taking into consideration the limited available resources, we employ EfficientNet, a lightweight Convolutional Neural Network which was found to achieve state-of-the-art results in image multi-label classification. Subsequently, as a baseline model we use VGG16 and as an experimental model we deploy the Vision Transformer, which seeks to integrate the Transformer layer that is widely used in natural language processing, into the field of computer vision. Furthermore, for variety purposes we finish our experiments by implementing the ResNet, DenseNet and MobileNet architectures. The results that are achieved are very promising, showcasing that there is high value in the visual information available with regards to the task of deforestation detection.

Περίληψη

Στον τομέα της ανάλυσης της κλιματικής αλλαγής, ένας μεγάλος όγκος πληροφορίας, ο οποίος προέρχεται από διάφορες πηγές και σε διαφορετικές εκδοχές, χρειάζεται να αναλυθεί με σκοπό την εξαγωγή των χρήσιμων συμπερασμάτων και προβλέψεων. Αυτή η διαδικασία εξαρτάται σε μεγάλο βαθμό από τις ακριβής μετρήσεις και το οπτικό υλικό. Ως αποτέλεσμα, τα όργανα που αποσκοπούν στην συλλογή της πληροφορίας αυτής είναι υψίστης σημασίας. Δεδομένου πως η ανάλυση της κλιματικής αλλαγής είναι ένα πεδίο με μεγάλο εύρος, η έρευνα θα εστιάσει στην ανίχνευση των παραγόντων που συμβάλλουν στο φαινόμενο της αποψίλωσης των δασών. Κλείνοντας, ο τύπος της πληροφορίας που θα χρησιμοποιηθεί θα είναι αποκλειστικά οπτικός, καθιστώντας το υλικό που προέρχεται από τους δορυφόρους ιδανικό.

Σε αυτή τη διπλωματική εργασία, εξετάζουμε αρχικά μεθόδους επεξεργασίας του υλικού από τους δορυφόρους στο οποίο αναφερθήκαμε πιο πριν. Το βήμα αυτό περιέχει τους πιθανούς μετασχηματισμούς που θα εφαρμόσουμε στο οπτικό υλικό, καθώς και την εξαργωγή των βέλτιστων συχνοτήτων όσον αφορά την απόδοση του δοθέντως μοντέλου. Έπειτα, θα εφαρμόσουμε κατηγοριοποίηση με πολλές ετικέτες σε μία προσπάθεια να περιγράψουμε τους παράγοντες της αποψίλωσης που λαμβάνουν χώρα σε κάθε εικόνα. Λαμβάνοντας υπ'όψη τους περιορισμένους διαθέσιμους πόρους για τη διεξαγωγή της διπλωματικής, χρησιμοποιούμε το EfficientNet, ένα ελαφρύ υπολογιστικά Συνελληκτικό Νευρωνικό Δίκτυο το οποίο επιτυγχάνει υπερσύγχρονα αποτελέσματα στην ταξινόμηση των εικόνων με πολλές ετικέτες. Επισπροσθέτως, ως σημείο αναφοράς για την απόδοση των υπολοίπων μοντέλων χρησιμοποιούμε το VGG16 και σαν πειραματικό μοντέλο χρησιμοποιούμε τον Vision Transformer, ο οποίος προσπαθεί να ενσωματώσει τα ”στρώματα μετασχηματισμού” τα οποία χρησιμοποιούνται ευρέως στον κλάδο της ανάλυσης κειμένου, στον κλάδο της όρασης υπολογιστών. Συνεχίζοντας, τελειώνουμε τα πειράματα υλοποιώντας τις αρχιτεκτονικές ResNet, DenseNet και MobileNet. Τα επιτευχθέντα αποτελέσματα είναι εξαιρετικά, υποδεικνύοντας πως υπάρχει μεγάλη αξία στην οπτική πληροφορία που μας είναι διαθέσιμη, όσον αφορά την χρήση της στην ανίχνευση της αποψίλωσης των δασών.

Acknowledgements

I would like to express my gratitude to my supervisor Athanasios Voulodimos for providing me with a great deal of support and assistance throughout the learning process of this thesis. His expertise and directions throughout this endeavor were invaluable. Furthermore, I would like to thank Maria Kaselimi for introducing me to the topic as well for the extended discussions and valuable suggestions which have contributed greatly to the improvement of the thesis.

Contents

List of Figures

List of Tables

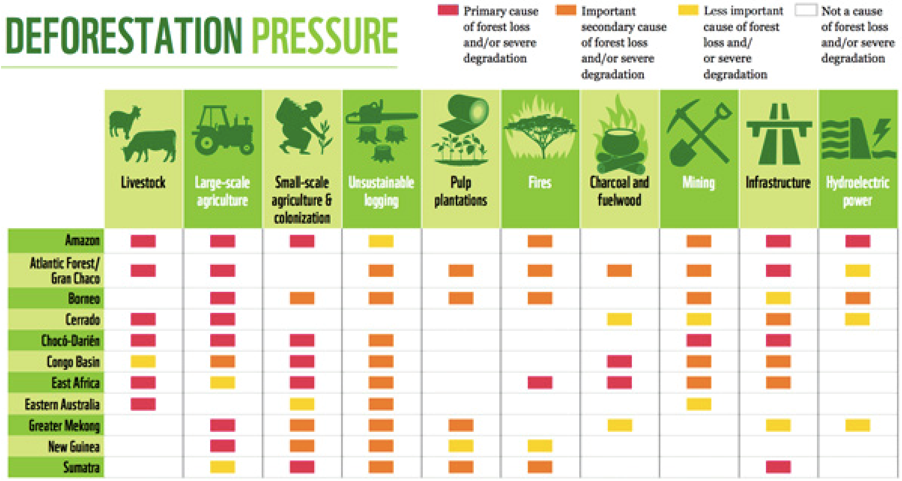
Chapter 1

Introduction

1.1 Deforestation Detection and Satellite Images

In recent times, the phenomenon of rapid climate change has been studied by a lot of scientific organizations that either simply monitor or seek solutions to the problem. Consequently, a lot of devices that monitor the earth's atmosphere have been set in motion in order to gather quality data, with the purpose of building high performance models that will detect and predict climate change. The aforementioned phenomenon is caused due to a plethora of contributing factors, with one of the major ones being deforestation, which will be the main topic of this thesis. This problem is approached from many different angles, which are primarily parametrical. Those approaches, although valuable, would gain a boost performance-wise if they took advantage of more of the sources of information that are available to us. One such source is the satellite imagery, which is broadcasted to earth from public and private satellites alike, in regular intervals. By capitalizing on this information, not only can we tackle the problem of detecting deforestation from a unique angle, but we can also enrich existing models that use other sources of information for achieving the same goal. However, achieving such a feat is no easy task as it is reliant upon understanding the factors that contribute to deforestation, as well as ways to extract and utilize the information hidden in the satellite imagery.

Deforestation, albeit indisputably concerning in our age, is not anything new. It has been happening for thousands of years, notably from the era of the first agricultural societies, when the need for more land-space was born. However, in our times, earth's population and consequently the need for more space, have increased exponentially at the expense of the existing forests and greenery. Therefore, a variety of problems direct and indirect have arisen. Firstly, a tragic consequence is the loss of plants and animals, even whole ecosystems. Then the reduction of the number of plants leads to less carbon dioxide filtration, thus contributing to another major climate problem, the Greenhouse Gases. Lastly, we have disruptions in the water cycle, soil erosion, flooding, etc, that threaten vulnerable indigenous people and in extent our society as a whole. Thereupon, it is only natural that detecting deforestation is a critical task of utmost importance for the field of climate change analysis. While the term deforestation is commonly used to describe the removal of trees or other greenery through artificial means, it is not limited to that. The loss of trees and other vegetation can also be attributed to accidental or natural means. Furthermore, directly removing the trees is only one way to cause artificial deforestation. Deeper inside the thesis we will thoroughly discuss about direct and other, more indirect factors that can cause it. Being able to categorize those factors is very beneficial in the process of understanding the gravity and reversibility of each one. With the rise of machine and deep learning a lot of effort has been directed into the quality of the categorization as well as the automation of the process for monitoring purposes. In our task, we will start by receiving satellite imagery, extract the parts of it that give the optimal value with regards to the performance and proceed to use deep learning models that take advantage of this information. Figure 1.1 illustrates some common causes of deforestation.



DEFORESTATION PRESSURE

🟥 Primary cause of forest loss and/or severe degradation

🟧 Important secondary cause of forest loss and/or severe degradation

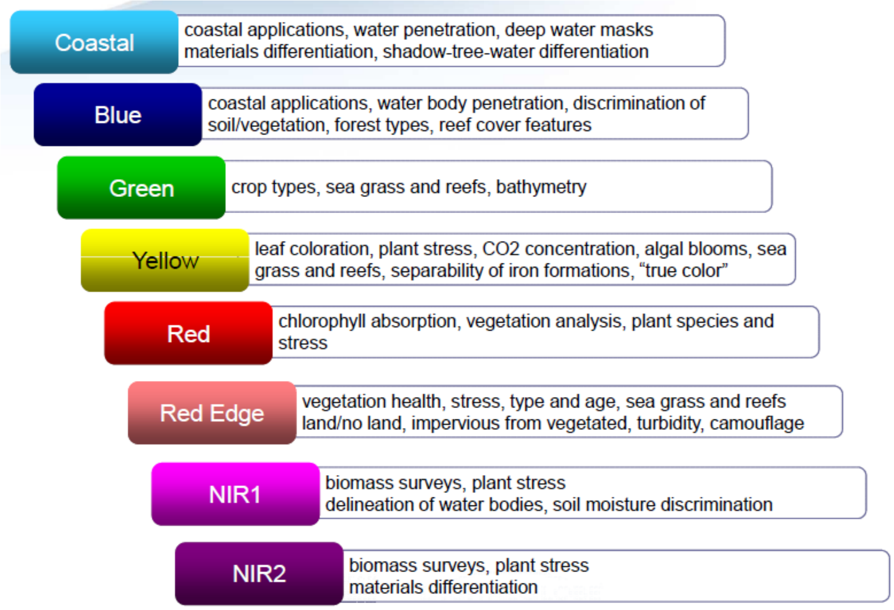
🟨 Less important cause of forest loss and/ or severe degradation

□ Not a cause of forest loss and/or severe degradation

	Livestock	Large-scale agriculture	Small-scale agriculture & colonization	Unsustainable logging	Pulp plantations	Fires	Charcoal and fuelwood	Mining	Infrastructure	Hydroelectric power
Amazon	🟥	🟥	🟥	🟨		🟧		🟧	🟥	🟥
Atlantic Forest/ Gran Chaco	🟥	🟥		🟧	🟧	🟧	🟧	🟧	🟥	🟨
Borneo		🟥	🟧	🟧	🟧	🟧		🟧	🟨	🟧
Cerrado	🟥	🟥					🟨	🟨	🟧	🟨
Chocó-Darién	🟥	🟥	🟥	🟧				🟥	🟥	
Congo Basin	🟨	🟧	🟥	🟧			🟥	🟧	🟧	
East Africa	🟥	🟨	🟥	🟧		🟥	🟥	🟧	🟧	
Eastern Australia	🟥		🟨	🟧				🟨		
Greater Mekong		🟥	🟧	🟧	🟧		🟨		🟨	🟨
New Guinea		🟥	🟧	🟧	🟨	🟨				
Sumatra		🟨	🟥	🟧	🟧	🟧			🟥	

Figure 1.1: Common factors that contribute to deforestation in different regions.

Typically, this approach utilizes satellite imagery, the reason being that satellites cover all of earth's surface in regular intervals and provide us with more depictions of the surface than the eye can see. Said depictions are called image bands and are created through the use of various amounts of electromagnetic radiation that are emitted from the satellite. This radiation is then reflected into the satellite and is measured by specialized sensors. The range of wavelengths measured is known as a band and is commonly described by the name and the wavelength of the energy being recorded. Thus, our task is to discern the bands that are beneficial for deforestation detection and use only them in an attempt to reduce training complexity and improve performance. The bands that are commonly captured by the satellite sensors are portrayed in figure 1.2.



Coastal	coastal applications, water penetration, deep water masks materials differentiation, shadow-tree-water differentiation
Blue	coastal applications, water body penetration, discrimination of soil/vegetation, forest types, reef cover features
Green	crop types, sea grass and reefs, bathymetry
Yellow	leaf coloration, plant stress, CO2 concentration, algal blooms, sea grass and reefs, separability of iron formations, “true color”
Red	chlorophyll absorption, vegetation analysis, plant species and stress
Red Edge	vegetation health, stress, type and age, sea grass and reefs land/no land, impervious from vegetated, turbidity, camouflage
NIR1	biomass surveys, plant stress delineation of water bodies, soil moisture discrimination
NIR2	biomass surveys, plant stress materials differentiation

Figure 1.2: Common bands that can be detected by most satellite sensory systems.

1.2 Goal of the thesis

As of yet, there has been no definitive solution to the problem of deforestation detection. The fact that the problem is still open, has allowed for the development of a variety of approaches towards its solution. A popular approach among them is the use of parametrical measurements which encapsulate distinct characteristics of the atmosphere and the ground of the area of interest. This approach, albeit successful, does not utilize the complete information available for each area. Utilizing the images could create a more complete representation of the area, enhancing the detection performance.

We utilize a dataset published in a Kaggle competition by a company named Planet, containing coarse-resolution imagery from Landsat (30 meter pixels) or MODIS (250 meter pixels). The area of interest is the Amazon forest as it is subject to a plethora of factors that contribute to deforestation, rendering the phenomenon there quite intense. Each entry contains imagery that comes in two formats, the first being the RBG bands in jpg, and the second being RGB plus the infrared band as tiff. All entries are described by a subset of 17 available labels that are named after popular factors of deforestation.

To cope with the problem of deforestation detection, many organizations have developed systems exclusively relying on parametrical information. As such they are unable to handle cases with poor or no parametrical information. Our task is to explore the application of deep convolutional neural network architectures in the context of deforestation detection, in an endeavor to support the systems that are already implemented, as well as to provide a standalone solution that utilizes satellite data. We start by choosing the right bands that should be extracted from the images. Then, we refine the dataset into a proper format in order to train three neural networks, each one tackling the same problem of multilabel classification from a different angle. We then use the trained models to extract a set of tags that describe the factors of deforestation that have taken place in each image. As we show later, with the recent advancements in computer vision, the information obtained from the images is not only useful, but can also rival the performance obtained from the parametrical approaches. Concluding, the integration of the work provided by this thesis can greatly enhance the existing systems.

1.3 Outline

Due to the multidisciplinary nature of this thesis, it is divided into subsections which are organized as follows:

- Chapter 2 discusses related work. It begins with a closer look into the work with regards to the Amazon rainforest deforestation and explains how the use of artificial intelligence has been utilized in this context. Those works mostly monitor said rainforest by classifying the deforestation drivers for plots of land that stem from satellite imagery. Of course, the research is not limited into the Amazon rainforest as the rest of the papers perform similar work for Indonesia and various mangrove forests. The following subsection maintains the theme of forest monitoring but this time from the angle of other factors that affect climate change. The works in question, monitor the forest in order to predict the places in which fires will take place and the path which those fires will choose, by taking into account flammable natural resources that exist inside those forests. Throughout this section, a plethora of techniques and neural network architectures are utilized and compared, inspiring the work in our thesis. Besides the climate change related work, we mention two works that introduce convolutional neural network architectures, the EfficientNet and the Vision Transformer, as our work utilizes those. Lastly, in its own subsection, explainable AI, a framework used to explain the results of our neural networks, also notoriously known as "black boxes", is described.

- Chapter 3 describes the dataflow of the system from band extraction to model architectures. Transforming raw satellite data to probability vectors is a lengthy process, worthy of being thoroughly explained in its own section. Briefly, raw satellite data contain many bands with different meanings, from which the useful ones for our research should be extracted. Then, those bands which are represented as arrays, are modified with the use of a generator in a format that is easily understood by a neural network. Among, the 6 neural architectures used, (VGG16, DenseNet, Resnet, MobileNet, EfficientNet, Vision transformer) the state of the art models (EfficientNet and Vision Transformers) are thoroughly explained in their own subsections. Those architectures will finally produce probability vectors which will determine which drivers of deforestation are likely to have taken place for each image. In their own subsections, AdamW, a variant of the Adam optimizer and the binary cross-entropy loss function, a loss function integral to multilabel classification, are analysed in detail.

- Chapter 4 presents the methodology and experiments. We start by discussing the origins of the dataset, how it was handled and a quick analysis of what we expect to find inside the dataset. Then, we introduce each one of the drivers of deforestation for the Amazon rainforest, that act as our dataset's labels. Afterwards, the experiments subsection takes place. For each model, the hyperparameters used and the reasons behind said choices are discussed. Also, their loss functions and confusion matrices are presented. Finally, after an attempt to explain the reasons behind those results, a direct comparison between all models takes place.

- Chapter 5 draws conclusions and proposes new ideas for future work. We start with a brief overview of the thesis and then, compare the final results with a previous work that used the same dataset for tackling same problem. In the following subsection we describe Noisy student, a student-teacher model that can be built directly on top of our EfficientNet and potentially increase its performance. Furthermore, we list some techniques that could not be used due to our limited resources but could benefit our models' performance, more tuning and scaling our EfficientNet up to B7. Lastly, TResNet, a very promising model for multilabel classification tasks, is briefly introduced as a viable alternative to our models.

Chapter 2

Related Work

2.1 Neural network architectures to tackle deforestation

The work of this thesis is a followup of the work performed by Aaron Loh and Kenneth Soo [12] who experimented upon the exact same dataset that we will experiment upon. They attempted to describe 256x256x4 satellite image chips that stem from Planet, using a set of 17 labels that correspond to the prevalent factors of deforestation with regards to the Amazon forest. To achieve this purpose, they made use of the transfer learning technique, implementing models that were pretrained in the imagenet dataset. The aforementioned models are convolutional neural networks and more specifically the VGG16, the ResNet50 and the InceptionNet, thus, providing a comparison of how the models fare in that dataset. The best model of the ones used, produced an f2 score of 0.89.

In the context of the Amazon rainforest deforestation, M. X. Ortega et al. [15] combined all the previous research done into a paper, putting emphasis in the variety of neural network architectures that were implemented in order to tackle the problem.

The authors begin with brief reference to the problem and the most notable techniques that were used to monitor the rainforest. Out of those methods, the authors picked some change detection methods that they would implement and evaluate. The initial method was the Early Fusion [4] where the images are stacked along their spectral dimension to generate a unique input image for patch extraction. The second method was the Siamese Network [28] where two identical networks with shared weights receive as input pairs of images in order to create a concatenated feature vector that acts as the classifier. The choice of techniques was conscious as the techniques share some properties including the feature vector stacking. Those networks were trained using a dataset that comprises of a pair of Landsat 8-OLI images, with 30m spatial resolution that were subjected to atmospheric correction. The final images had 1100 ×2600 pixels and seven spectral bands (Coastal/Aerosol, Blue, Green, Red, NIR, SWIR-1,and SWIR-2). The experiments showed that both networks have the capacity to perform better than traditional machine learning techniques used in the field, such as the SVM (used as the baseline in their paper).

Adding to the research on Amazon rainforest deforestation, Rafael A. S. Rosa et al. [18] attempt to detect the phenomenon making use of use of SAR (synthetic aperture radar). They propose a new method of change detection in multitemporal SAR images using X- and P-band SAR images simultaneously to calculate a change detection indicator image (binary mask) based in the coherences between all the images used as attributes calculated from superpixel segments to define a change detection neural network. The data set was provided by Santo Antonio Energia S.A. acquired by the airborne sensor OrbiSAR-2 from

Bradar. They were collected in the period between 2012 and 2014 at X and P bands. Their resolution was 1000x1000 representing an area of one square kilometer. In order to detect change, each set of 17 images (one image for each month) dedicated to the same area, was condensed into 1 image consisting of superpixels through PCA. Furthermore, those superpixel images were given to a Multi-Layer Perceptron (MLP) with 10 folds of cross validation which produced as output change or not change in the image sets. The results varied greatly between image sets and were worse than previous work. Nevertheless, the authors conclude that this is normal, as their neural network was fed only one attribute (superpixels) while the previous work used 7 features. Thus, the produced superpixels were of greater quality and paired with other features would produce substantially better results.

Moving away from the Amazon rainforest, Jeremy Irniv et al. [8], perform a similar experiment for Indonesia. Indonesia is another country where the rate of deforestation is extremely high. In an attempt to detect the drivers of deforestation in this country, the research team implemented the ForestNet. This model is able to accept patches of satellite imagery of any size and classify them under the categories of plantation, agriculture, grassland and a default one for all the rest. The authors went through the effort of manually labeling Landsat 8 satellite imagery that is derived from google earth, from 2013 onward. All images were then converted to surface reflectance, 332x332 chips to account for atmospheric scattering or absorption. More than 50% cloudy images were discarded. All images were subject to per pixel classification. In order to achieve the desired performance, the authors employed data augmentation while experimenting with a variety of architectures like UNet[10], Feature Pyramid Networks[11], DeepLabV3[27] and EfficientNet. The best performing model, which took the name ForestNet, was a Feature Pyramid Network with an EfficientNetB2 as the backbone. The results were promising as the performance was generally high for all the drivers, albeit further improvements could be made, such as taking more drivers and the evolution of the landscapes into account.

Dillon Hicks et al. [7] focus on mangrove ecosystems that act as carbon sequestrators, limiting the effect that carbon emissions have on climate change in the process. While the reasons to preserve those forests are clear, those ecosystems seem to decline by 2% per year. Under those circumstances, the authors developed a system that can monitor areas of mangrove forests. The dataset consists of UAV satellite imagery from mangrove sites in Baja California Sur between 2018 and 2020 using a DJI Phantom 4 Pro UAV. The images are high quality (3840 x 2160) taken from an altitude of 120m using DJI GroundStation Pro. Then they produced additional features from the images such as the normalized difference vegetation index (NDVI) and the normalized difference water index (NDWI). Annotators created masks for each image with only 2 labels (magrove or other) as the purpose was only to detect the existence or not of mangrove forests. The architecture used was a hybrid CNN made of a perceptron and a pretrained EfficientNetB0 whose output embeddings are concatenated in order to produce the final output (mangrove, non mangrove). The performance of the hybrid model was high. However, the fact that this hybrid model needs both drone and satellite imagery of high resolution as an input, makes its usage very limiting.

An earlier work conducted by Thiago Nunes Kehl et al. [14] proposes a tool that can be used in the field of deforestation monitoring, as its purpose is to be able to perform deforestation detection in regular intervals. This dataset too, stems from the Amazon forest as it is captured by the MODIS/TERRA sensor of a satellite. The tool uses an artificial neural network for the processing of the dataset for which it provides parametrization and configuration capabilities, so that it can be adapted to more problems. This time, the performance evaluation of this model is done using a confusion matrix, so it is not directly comparable to the results of our thesis or the aforementioned related work. With this setup, a spectrum-temporal analysis of a region of the Amazon was made on 57 images. Finally, they conclude that such techniques that involve neural networks, have a strong potential with regards to deforestation detection, but as of the time the paper was released, the false alarms that would be fired would impact the consistency of this technique, making it difficult to be used for real world implementations.

2.2 Neural network architectures to tackle climate change

The phenomenon of climate change has been in the forefront of research for quite some time. Understanding the forestry can lead to solutions to issues that exceed the scope of deforestation itself. Pranoy Panda et al. [16] map vegetation in order to model the way wildfires will behave in the US west coast. For the purposes of this research, they use wildfire fuel data from nadir (downward-looking) images taken by drones or humans. Wildlife fuels can consist of grass, moss, and dead needles from conifer trees (this paper focuses on wildlife fuels that can be found in the ground). The dataset consists of 28 areas, each one being represented as a 4x4 grid of images for a total of 448 images across all areas. The images contained the labels of firewood, forb, grass, Lichen, Moss-Feath, Moss-Other, Moss-Sphag, shrub, non-fuel, and void. The model used was a Deeplabv3-Resnet101 which was pretrained on the COCO train2017 dataset. The need to use transfer learning came up due to the huge variety and the small amount of the available images. The model actually needed a huge amount of images in order to be properly trained and they just were not available. Additionally, data augmentation was used in order to quadruple the size of the dataset. Since the problem is one that requires segmentation, the images were masked according to the aforementioned labels and the model tried to approximate this masks. In overall, the number of correctly classified pixels is high, which is the metric that the authors prompt the reader to pay more attention to.

On a more global scale, Yongjia Song et al. [23] research techniques to predict wildfires using neural networks and non-linear models. The dataset consisted of statistical small fire data from Global Fire Emissions Database, meteorological data from National Centers for Environmental Prediction Climate Forecast System Reanalysis (CFSR) and three long-term climate ocean indices (ONI, AMO, PDO) from Earth System Research Laboratory of NOAA. Three non-linear statistical models, GLM, regression tree, and neural networks were applied to the aforementioned feature sets. Among those models, a combination were the GLM was used to select the best predictor parameters combination and the neural network produced a 1-year moving forecast seemed to yield the best results. The authors conclude that the model accounted for seasonal and regional patterns, showcasing a capacity for high performance.

2.3 Convolutional neural network architectures

A primary purpose of this paper is to showcase how interesting, state of the art neural network architectures fare with regards to this problem. For this reason, we will add the EfficientNet into our arsenal, introduced by Mingxing Tan et al. [25]. The authors delve deeper into the computational cost and performance trade-offs with regards to the convolutional networks. For this purpose, they devise a formula, that dictates a way to uniformly scale the convnets. This formula, introduces the compound coefficient which can scale baseline networks in discrete levels. They call the baseline network EfficientNetB0, while its largest counterpart is the EfficientNetB7. This formula acts as a rule of the thumb in convenet scaling procedures and as showcased can be used in other traditional networks such as the ResNet and the MobileNet. Additionally, using a mathematical formula to tackle the scaling problem, alleviates the need for lengthy tuning sessions, making EfficientNets attractive choices when low computational resources are available. EfficientNet surpassed in accuracy all other convolutional networks [2, 6] of its time on the popular ImageNet and Cifar-100 dataset benchmarks.

More recently, Alexey Dosovitskiy et al. [5] proposed a general purpose convolution neural network upon which we will experiment for the purposes of this thesis. The paper imports the notion of Transformers, which is an architecture that is widely used for natural language processing tasks, into the field of computer vision. Essentially, they seek to substitute specific components of existing neural networks in an effort to integrate the Transformer blocks in a non-invasive manner in an attempt top increase their performance. The reasoning behind those experiments is to show that image classification should not necessarily be reliant upon convolutional neural networks. If an image is tessellated into patches, a Transformer can train on those patches as well as it does for words. By managing to showcase the proposed model's superiority over other traditional CNNs with regards to its performance on benchmark datasets that are commonly used for classification, the authors conclude that this technique is viable for such tasks.

2.4 Explainable AI

Lastly, a vital part of every research is the explainability of the end results. Alejandro Barredo Arrieta et al. [1] analyze the various efforts that have gone into analyzing the results that are extracted from neural networks. Those techniques were coined under the umbrella term of explainable AI. Explainable AI is a broad field, so the authors study those techniques from the scopes of understandability, comprehensibility, interpretability, explainability, transparency. Generally, a black-box model can be explained textually, by simplification, by a visualization, by local explanation, by feature analysis and with an explanation by example. Having set the pillars of Explainable AI, the authors proceed to analyze each prevalent machine learning technique by using their framework. Notably, they state that convolutional neural networks can be explained in an easier fashion than other types of models. The techniques that try to explain them can be divided into those that map the output in the input space to see which parts of the input were discriminative for the output and delve into the intermediate layers of the network and extracts the information that it 'sees' at that point. To tackle the first issue, DeconvNet [21] was used, which when fed with a feature map from a selected layer, reconstructs the maximum activations. Those activations can show what parts of the image played a bigger role in the output. With regards to the second issue, the solution is to extract the photographic output of the intermediate convolutional layers, for the images that maximized the probability output in a particular class. A followup technique in the same direction was the Deep Generator Network (DGN) which generated the most representative image for a given output neuron in a CNN. The aforementioned and a plethora of other equally important works, proved to be the milestones of convolutional network explainability, which is integral for the research of this thesis.

Out of all the methods that are used to explain the convolutional neural networks that were listed previously, in our thesis we will choose the Grad-Cam, a work of Selvaraju et al. [20]. The technique's name stands for Gradient-weighted Class Activation Mapping. It uses the gradients of any target concept that flow inside a neural network until the last convolution in order to highlight the most important regions of the input image with regards to the classification decision. A major strength of this technique as the authors state, is its ability to be applied to a wide range of neural networks without the need for retraining or extra layers. Those visualizations are very useful for understanding the reasons behind classification failures and the classifications prowess of the model. Additionally, the authors propose an extension of Grad-Cam (which makes class based decisions when highlighting pixels), namely the Guided Grad-Cam, which aims to highlight some more fine-grained details in the image. This technique was initially used to interpret the results of a VGG16, demonstrating an increase in both the faithfulness and interpretability of the model. In this context, faithfulness is how accurately the function of the model is portrayed and interpretability has to do with the degree in which the results can be understood by humans. While there is a tradeoff between those two, this technique scored high in both metrics in a mixture of technical and human experiments.

Chapter 3

Methods

3.1 Deforestation Detection Pipeline

The deforestation detection pipeline is fairly complex as we are tasked with producing probability vectors out of raw satellite imagery. The images that stem from the satellites are not in a form that can be directly understood by a neural network. A satellite image comes with a plethora of brands that represent different information in the form frequencies. It is our duty extract the useful bands for our problem, that means the ones with the most amount of information. However, the process does not end with the band extraction. Those bands should be refined to a format that is understood by the neural network, which is usually multidimensional arrays in [0-1]. Only under those conditions will the neural network model accept them in order to turn them into probability vectors. The probability vector indicates how likely it is for a deforestation driver to have taken place in an image.

Figure 3.1 shows our system's pipeline for detecting deforestation. The initial step is to read the appropriate bands from the satellite images given in tiff format. For this reason, a basic image generator has been implemented with the purpose of reading the first 4 channels of the images (RGB + Infrared). The images are stored locally and the corresponding dataset that matches the image names with their labels is transformed into a format that can be parsed from the aforementioned generator. Afterwards, the generator applies some transformations and forwards the images into a neural network. Finally, the neural network produces an array with the probability that each label has to describe each image.

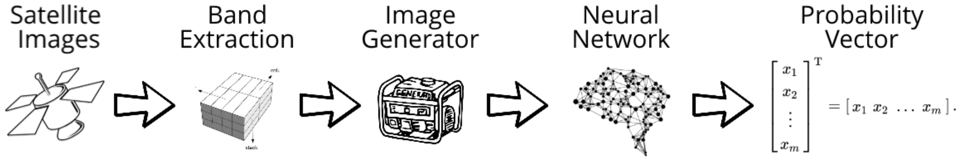

Satellite Images
Band Extraction
Image Generator
Neural Network
Probability Vector
$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}^{\mathrm{T}} = [\, x_1 \; x_2 \; \dots \; x_m \,].$

Figure 3.1: Flow of satellite information processing towards multilabel classification

3.2 Data Preprocessing

The first step is to find a dataset that is appropriate for our problem. With this in mind, we will use the one that was published by Planet, in whose specifics we will dive deeper in the next chapter. This dataset, is available in jpg format (RGB) and tiff format with 4 channels (RGB + near infrared). Most of the work with regards to image extraction is already done, what remains is to decide whether we need the extra near infrared band or not. For this thesis, it was decided that near infrared is a vital dimension of the image as it represents the existence and intensity of greenery, prevalent in more than half of the dataset's labels. With this decision, we will need more computational resources in order to process this information but will also get better performance. The images are paired a csv file that matches the image names with a subset of a total of 17 different labels that are the drivers of deforestation.

Having the images and the csv downloaded locally, we first replace the image names in the csv with the full paths that lead to the images. Then, we expand the label column (that holds the image labels as an array) into 17 columns with values the existence or the absence of each label from the image. This format is acceptable from the image generator for performing multilabel classification. Afterwards the csv is split into 3 datasets (train/valid/test) (60/20/20) and is stored as different csv files. Those files are ready to be read by the generators.

3.3 The Generator

Sometimes, even the most state-of-the-art configurations have not enough memory to process a large enough dataset. Datasets that consist of images are a prime example of this description. To deal with this problem we use a generator which will constantly load images from the local storage, transform them and push them into the neural network.

The generator will start by reading the images' locations in the local storage from the dataset and load the corresponding images. Afterwards, it will parse the tiff images that it just read, looking for the first 4 dimensions (RGB + Infrared in an RGBA setting) and resize them to a specified square resolution (i.e. 112x112 or 224x224). Also, the values contained in each channel will be normalized to [0, 1] to improve neural network performance as its activation function usually expect and output values inside this range. Optionally, more transformations can occur. In our case we used horizontal and vertical flip as they do not change the meaning and the context of each image (as they depict a patch of land). A total of three generators will be deployed for the train, valid, and test set respectively. In contrast with the other two, the test generator will not shuffle the images, so that we will be able to match the resulting labels with the true labels. Lastly, each generator will accept each the image batches in 'raw' mode with the specified tags given explicitly, so that we can get the multilabel behaviour that we want.

3.4 Models

3.4.1 Architectures

Throughout our experiments we will be making exclusive use of convolutional neural networks (CNN). This is a popular choice for image processing as all its benefits are geared towards such tasks. CNNs are fully connected feed forward networks that can accept as inputs of high dimensionality, reduce the number of parameters inside the system, but not the original input's information. This reduction happens due to the use of filters (windows smaller than the actual input), that slide through the input while retaining the information of the patches of the input in which it has already slid. The output of those filters is a more compact representation of the input. Images are such a high dimensional input, which would normally take many times more resources to compute if not for this technique. Furthermore, due to this benefit, a lot of research has been done in the field of image processing and neural networks to the point of the models becoming specialised and many times more efficient in this task. As a direct consequence, all the image classification benchmarks, be it multilabel or unsupervised, are dominated by the state of the art convolutional neural networks. Due to the sheer number of options available, a multitude of architectures was examined before choosing the best one for the problem.

VGG16 [22] was used as the baseline model as it produced decent results while costing relatively cheap in resources. In the same direction, DenseNet, ResNet and MobileNet were also tried as out of the box solutions that have stood the test of time. Afterwards, we used the more complex Convolutional Neural Network architectures EfficientNet [25] and Vision Transformers [5] increasing the system's performance in the process. All the aforementioned models were trained, validated and tested using exclusively the labeled images from Planet. In all the experiments the models with the randomly initialized weights produced superior results. Nevertheless, the EfficientNet outperformed the other models apart from the Vision Transformer by a respectable margin. The comparison will be discussed thoroughly in the experiments section. Figure 3.5 showcases a comparison between EfficientNet family and other popular architectures. All models in the figure were trained and tested in the popular Imagenet dataset.

3.4.2 EfficientNet

As previously mentioned, efficiency is a major factor when choosing for a model. Paired with our low computational resources and the use of the 4th channel the first choice was not very difficult. The first choice is the efficientNet. Tan et al. [25] approaches the process of finding better architectures by developing a basic low cost model which then can be scaled up in order to achieve better performance. Scaling up in this case means increasing the width and depth of the network and the resolution of the given images. Tuning those 3 parameters for the best performance usually requires a lot of manual tuning and resources. In this thesis we experiment with EfficientNet which allows the scaling process to be done in a more principled manner by using a compound coefficient to scale up the network. As a consequence, by tuning only one hyperparameter, the model can scale-up its width, depth and resolution uniformly by choosing from a set of fixed values for the previously mentioned compound coefficient. This allows for more efficient tuning while also producing models that can achieve state-of-the-art performance with 10 times more efficiency. The basic EfficientNet architecture is called EfficientNet B0 and its architecture is depicted in figure 3.2. EfficientNet will not only allow us to scale the model in the most cost efficient way, it is also a top performing model in both multiclass and multilabel classification problems, making it ideal for our task.

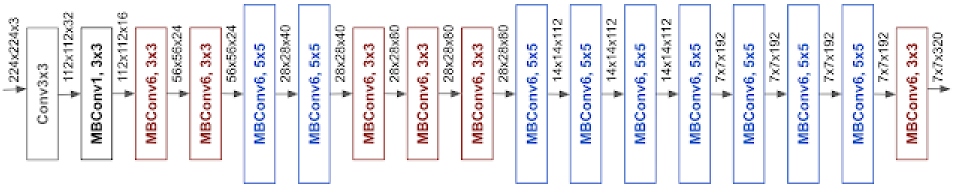


Figure 3.2: Architecture of Efficientnet-B0

3.4.3 Compound Scaling

Compound scaling is based on the idea that properly balancing width, depth and resolution according to the available resources will provide superior performance overall. To find the compound scaling coefficient one must perform a grid search to determine the relationship between those 3 hyperparameters always under the restriction imposed by the resources. Once the relationship is found, the coefficients must scale up until they take up all the available resources. The authors propose the following formula that describes the relationship between the three hyperparameters:

$$\begin{align} \textbf{depth: } d = \alpha^{\phi} \tag{3.1} \\ \textbf{width: } w = \beta^{\phi} \tag{3.2} \\ \textbf{resolution: } r = \gamma^{\phi} \tag{3.3} \\ \textbf{s.t. } \alpha * \beta^2 * \gamma^2 \approx 2 \tag{3.4} \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{3.5} \end{align}$$

ϕ is a user-specified coefficient that controls resources in terms of FLOPs (Floating Point Operations) and α, β, γ distribute the resources to depth, width, and resolution respectively. FLOPS of a regular convolution op is almost proportional to d, w^2, r^2, hence doubling the depth will double the FLOPS while doubling width or resolution increases FLOPS almost by four times. Hence, in order to make sure that the total FLOPS don't exceed 2^{ϕ}, the constraint applied is that $(\alpha * \beta^2 * \gamma^2) \approx 2$. Figure 3.3 visually expresses how compound scaling affects the network.

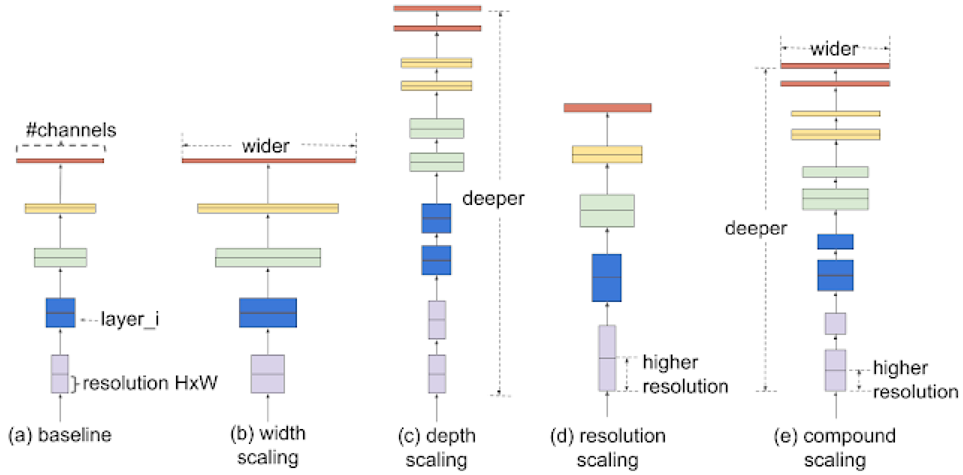
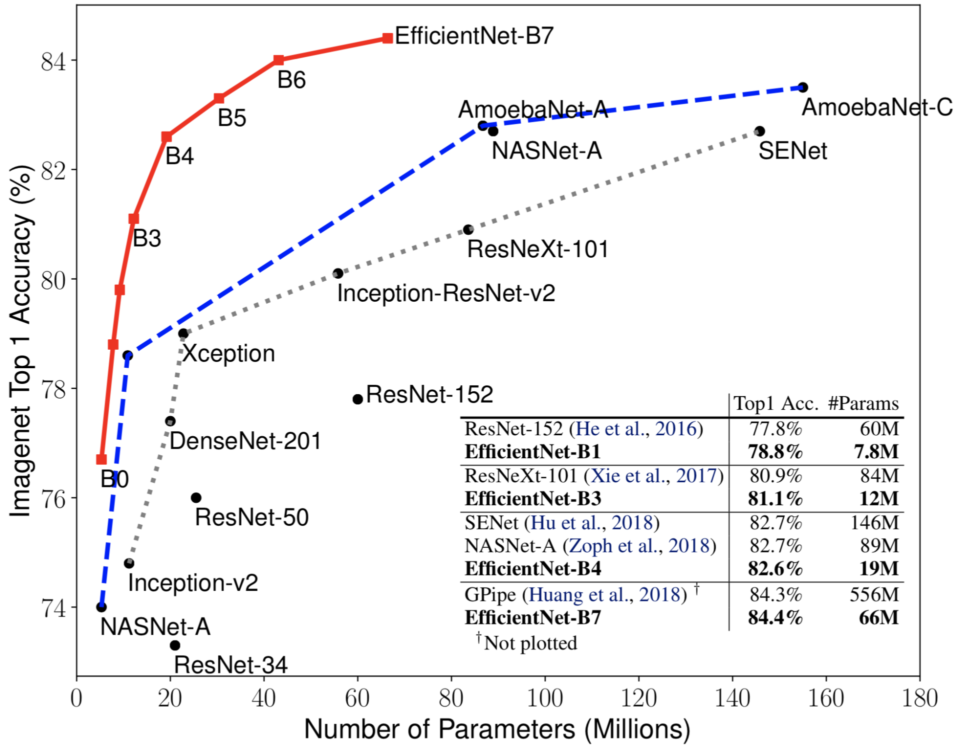


Figure 3.3: Scaling width, depth, resolution and Compound Scaling

This process continuously improves performance with more resources available. EfficientNet takes this idea a step further by providing a list of fixed coeffcients which have been trained in the imagenet dataset. Each set of coefficients has got a distinct name starting from efficient-B0 (base network). The largest network is the efficientnet-B7. As shown in the image 3.4, scaling the model further produces diminishing returns in performance.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Figure 3.4: Comparison between the EfficientNet and other state-of-the-art models in terms of performance

3.4.4 EffiecientNet Advanced Features

EfficientNet uses a variety of advanced techniques in order to achieve state-of-the-art efficiency. The most notable of them is model compression, which ensures the model remains lightweight even when scaled. Model compression is first and foremost achieved through pruning, where the parameters which do not help improve the performance are removed. Another part of model compression is quantization, where the parameters which are usually stored in 32-bit numbers, are converted to lesser precision if deemed necessary.

3.4.5 Vision Transformer

While the EfficientNet is an established state-of-the-art performer for image classification, Alexey Dosovitskiy et al. proposes the Vision Transformer as a way to tackle this task, challenging the traditional convolutional neural networks that are usually utilized. The primary reason for wanting to use this model, is that it is highly experimental, but has also shown very good performance in the initial tests. Additionally, as we will see later, it combines the transformers' ability of analyzing an input from many points of view (contexts), which is a very beneficial property for multilabel classification, while also retaining all the major benefits of convolutional neural networks that we described earlier.

Vision Transformer, as the name implies, attempts to integrate the well known notion in the field of natural language processing of Transformers, into the field of computer vision. Although there have already been some attempts in previous papers to do just that, with the most notable being the 'End-to-End Object Detection with Transformers' [3] by Carion el al. of Facebook, Vision Transformer is the first one to achieve significant success at that. In popular benchmarks, not only did it improve the performance, but it also claimed to have cut the training time set by another popular model published by Google, the Noisy Student [26] (which is actually a technique that uses a variant of the EfficientNet model at its core) by 80%.

Diving deeper into the Vision Transformer architecture, we can discern three major components. The first component is the layer that is responsible for the image preprocessing. This layer accepts the image as a three dimensional square matrix with values in [0, 1], divides it in square patches, which are then flattened into image embeddings. Next, we have 1 or more Transformer layers which treat those image embeddings the same as they would treat the text embeddings. This is a complicated layer and we will discuss it in detail in the next sections. Lastly, a series of feed forward layers constitute the head of the model. In the image 3.5, the high level architecture of the Vision Transformer is showcased.

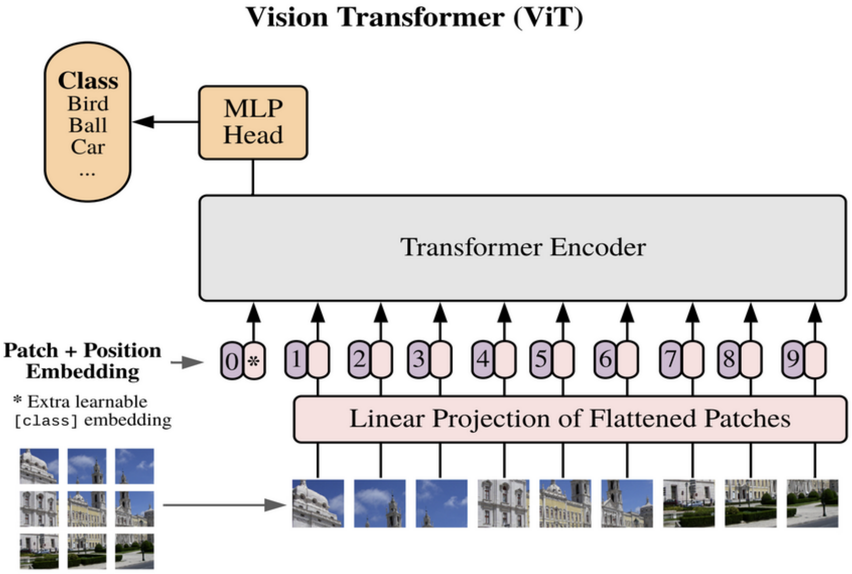

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 3.5: High level architecture of the Vision Transformer

3.4.6 The Transformer Layer

In order to understand the reason behind the claim that the Vision Transformer is different from the other convolutional neural networks, we first need to understand the Transformer, which is its core building block. In the field of natural language processing, the Transformer behaves in a way such that for each word inputted, its relationship with each other word is examined and a matrix containing said relationships as values is outputted. Naturally, this means that the order between those words is not important. The Transformer layer of the Vision Transformer does the exact same thing, but this time, it processes image patches instead of words. The image patches are simple tessellations of the original image. This bidirectional architecture of the Transformer allows for a great level of parallelization, thus avoiding the usual bottlenecks commonly found in CNNs. This parallelization allows for the full exploitation of the GPU/TPU resources, which is incidentally responsible for the major decrease in the training time when compared to other CNNs. Image 3.6 showcases the architecture of a typical Transformer layer. It is a self-attention layer followed by a series of addition, normalization and feed-forward layers. The self-attention layer is the secret behind the calculations of the correlations between all possible image patch pairs which constitute the ”context” of the image. We will focus on the self-attention layer in the next section. Furthermore, add the possible meanings of each patch together in order to normalize them and perform further calculations. The feed-forward layer in-between exists for the fine tuning of the weights received from the self-attention layer.

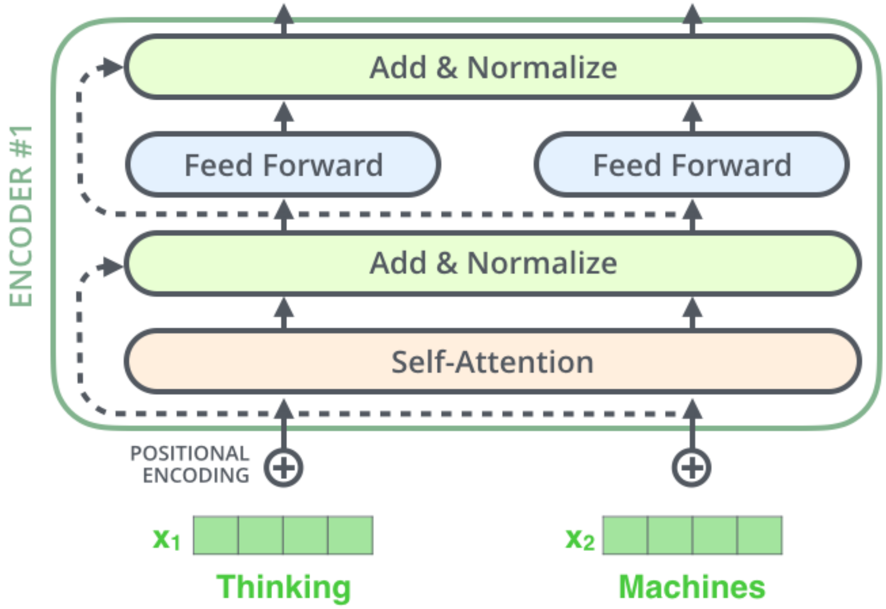


Figure 3.6: Transformer’s internal architecture

3.4.7 The Self-attention Mechanism

Again, using natural language processing as an example, simply parsing a sentence word by word has been proven to not provide optimal results. Apart from the meaning of the words themselves, it is important to note that there is an underlying context upon which they are ”glued” together. Self-attention attempts to do just that. It interprets the context in which the words are used, by trying to identify the correlation between those words. It is actually a layer that exists inside the broader Transformer layer. In the context of the Vision Transformer, where instead of words we have patches of images, the correlation between those images is what is measured, thus trying to identify the context in which its patch exists relative to the other patches. On left of the image 3.6 we see the input to a self-attention layer which is a set of flattened image patches. Those patches together form the query-key matrix which we can observe on the right. Each patch is one query, each pixel position is a key. The summation of the resulting queries by key, after the self attention layer, results in the importance vector which holds the importance of each key in the context of the image.

The self-attention mechanism is theorised to be beneficial for multilabel classification and the reason this model was chosen over other experimental models. Instead of having one input that carries all the information throughout the model, the self-attention mechanism breaks this output into several different outputs that should ideally carry different pieces of information before being glued together again. This means, that the process could break the processed image into patches that contain information almost exclusive to each one of our labels, that remain discrete upon the combination phase, making the decision for the probability for each label on the final output level much clearer and the model more confident. As a direct consequence, it might also be able to retain information about rarer labels, that could be easily obscured by the more prevalent labels in traditional models.

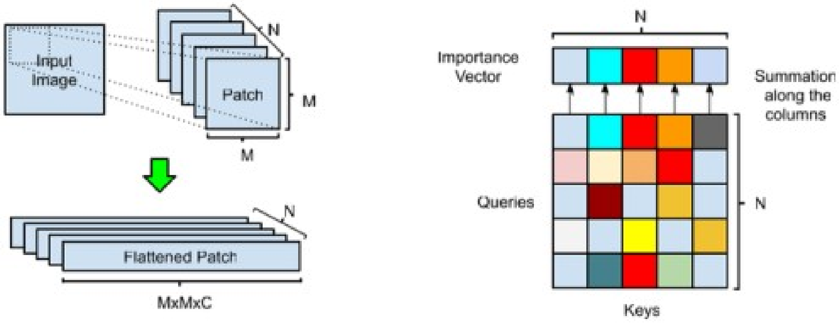


Figure 3.7: One the left we can observe the input given to a self-attention layer. On the right we can see the input's matrix representation

3.4.8 The AdamW Optimizer

A very important factor that defines the success of a neural network model is the optimizer being used. Algorithms like Adam [9] achieve fast convergence with an element-wise scaling term on learning rates. Despite their success, they have been observed to generalize poorly compared to SGD as suggested by Pedro Savarese et al. [19]. A possible cause of this issue is that L2 regularization and weight decay delay are equivalent for standard stochastic gradient descent (when rescaled by the learning rate), whereas this is not the case for adaptive gradient algorithms. AdamW [13] is a variant of Adam which has the goal of decoupling weight decay from L2 regularization for adaptive gradient algorithms with regards to the loss function. This approach retains the speed of its predecessor while also resulting in better generalization capacity. Another benefit of this optimizer is that the neural network becomes less sensitive to hyperparameter changes. This is very useful in our case study as the resources that are available for training are limited, restricting us from tuning.

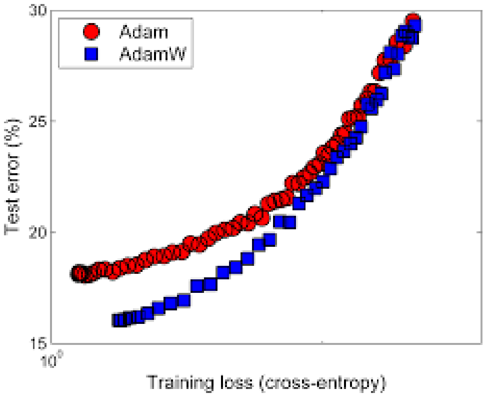


Figure 3.8: Comparison between Adam and AdamW with regards to the loss function

3.4.9 The Binary Cross-entropy Loss Function

Moving forward, an integral part to every deep learning problem is the loss function that will be chosen in order to evaluate the training phase of the model. The loss function that is commonly used for the problem of multilabel classification is binary cross-entropy and it is important to understand why. For starters, binary cross-entropy is a function that as its name implies, is used to evaluate how well the model is trained in a binary problem. So given the classes y=[0, 1], if the true class is 1, log(p(y)) is added to the loss and conversely if the true class is 0, log(1 - p(y)) is added to the loss. That said, it is becoming clear that we should strive for the least amount of loss possible during the training of the model. Shifting from the binary problem to the multilabel one, instead of [1, 0] being our only possible classes, we have multiple classes that consist of multiple [0, 1] labels. To make this more clear, an example of the true labels for an image could be [0, 1, 0, 1, 0, 0, ...] where each element represents the existence or not of a label. This is the reason we previously converted our labels in this format during the preprocessing phase. Since the premise of each label getting the values 0 or 1 is the same, we can treat this problem as multiple binary classifications where for each image, the existence or not, of every one of the possible labels adds up to the total loss. Of course it is more punishing for the model than in a regular binary classification, but this is beneficial to us, as the model will strive for precision without neglecting any label. The binary cross-entropy is best paired with the sigmoid, as the resulting probabilities of each label will not affect the others, unlike other activation functions like softmax where the dominant output probability will skew the rest.

3.4.10 Explainable AI and Grad-Cam

Finally, before moving to the experimental section, it is worth discussing the techniques that will be used in the context of explaining the results of our neural network architectures. It is well known that neural networks in general, while being highly permormant, are often viewed as a black-box. Nevertheless, being able to peek into their inner workings, would give us a huge advantage towards thoughtfully improving them in contrast to blind tuning. Since this thesis makes exclusive use of convolutional neural network variants, we will uniformly use 2 different methods to achieve our purposes.

Firstly, we need to be able to discern, which parts of the image, play the biggest part in the classification process for each class. To achieve that, we implement the Grad-Cam, which uses the gradient produced from the input image's flow inside the network for highlighting the parts of the image that are likely to have excited the activation functions the most. All of our models share the requirement for this technique to be used, they feature a global max pooling layer after their last convolution. By examining the results, we will seek for patterns followed across all our neural network architectures, as well as for reasons that some classes have potentially failed.

Secondly, instead of mapping the activation functions to the input, it might also be useful to peek into individual slices of the model. Unfortunately, our models and especially the Vision Transformer vary dramatically in their inner architecture. Thus, we will use the only common ground between them, their last convolutional layer before being maxpooled. This is not to be confused with the previous technique that maps all the network's activation functions from beginning to end. This is a mere slice in order to find out which neurons were excited at a stage very close to the network's output. Hopefully, some patterns shared by the models might emerge.

Chapter 4

Experiments

4.1 Datasets

When it comes to satellite imagery, there is no shortage of it as it can be found in various repositories around the globe. The real issue is that only a minor subset of those images are labeled in order to support classification tasks. Since we would like to perform multilabel classification, the dataset offered from Planet seems to be an ideal choice as it is large, it is cleaned up to a degree and it is also labeled. The size of the dataset is approximately 40k labeled and 40k non-labeled images. For our purposes we will use the labeled images only. The format in which the images are given is either in a 3-channel jpg or 4-channel tiff.

Since the fourth channel, which is the infrared is very useful for discerning greenery, we will use the images in the tiff format. All the redundant GeoTiff information has been stripped so that only the image channel values and the metadata that correspond to the image format remain. he imagery has a ground-sample distance (GSD) of 3.7m and an orthorectified pixel size of 3m.The data comes from Planet's Flock 2 satellites in both sun-synchronous and ISS orbits and was collected between January 1, 2016 and February 1, 2017.

Now, let's analyze the class labels with which we are tasked to train the model. The images have been labeled by the company's teams and crowd-sourcing. There is a total of 17 labels. Plot 4.1 shows the names of the 17 labels and how many times they are found in the dataset. Judging by this plot, we will have to deal with an unbalanced label distribution.

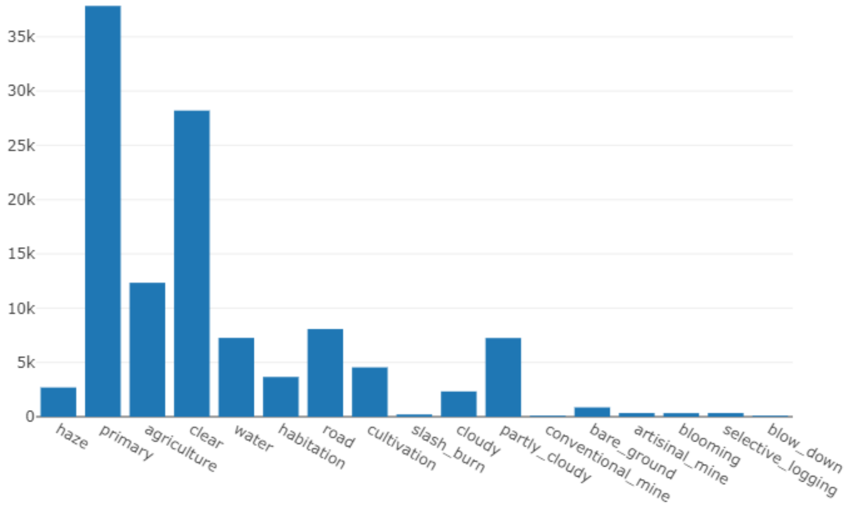


Figure 4.1: Label distribution

Now that we have seen the labels by name, a quick introduction of each one will follow.

4.1.1 Cloudy

Cloudy images pose a major difficulty for satellite imagery. Although a lot of techniques have been created specifically for the reason of clearing the clouds off the images, a lot of information is still lost in the process. Thus, it is very difficult for a computer vision model to discern any of the characteristics that make up the rest of the labels. So, to avoid making an uneducated guess, we will simply categorize those images as cloudy.

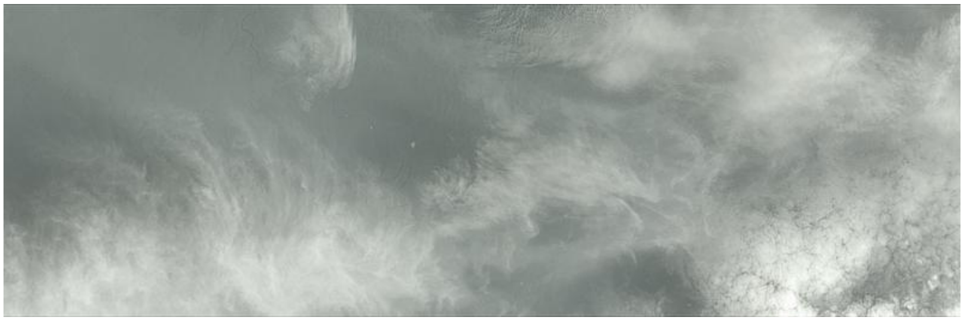


Figure 4.2: Example of a cloudy image

4.1.2 Partly Cloudy

In the same manner, partly cloudy images, completely obscure some of the areas in the image. The information in those places is lost, but it still leaves room for the model to make a detection based on some finer details.

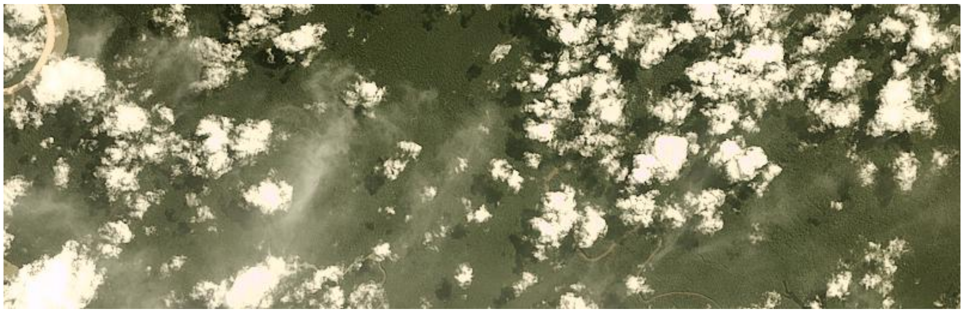


Figure 4.3: Example of a partly cloudy image

4.1.3 Hazy

Hazy images, still pose a great issue for satellite imaging, but depending on the level of the haziness, some details may still be observable. Usually, the model cannot detect some finer details in those images, but some major causes of deforestation such as agriculture can very well be detected.

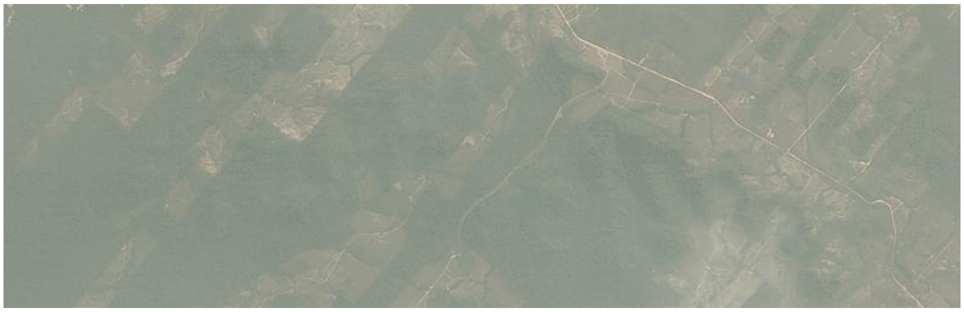


Figure 4.4: Example of a hazy image

4.1.4 Clear

Clear is every image which does not have any form of clouds or haze, which fortunately is the vast majority of the images as we saw in the label distribution. Those, are the ideal images for deforestation detection.

4.1.5 Water

Water is a component that predicts the areas where we will expect the areas to be full of greenery. Manipulating the flow of the rivers, often leads to an increase of the tree growth speed in damaged areas. Water also reflects a lot of the infrared frequencies, especially during summer, so it is important to be able to discern it from actual plants.

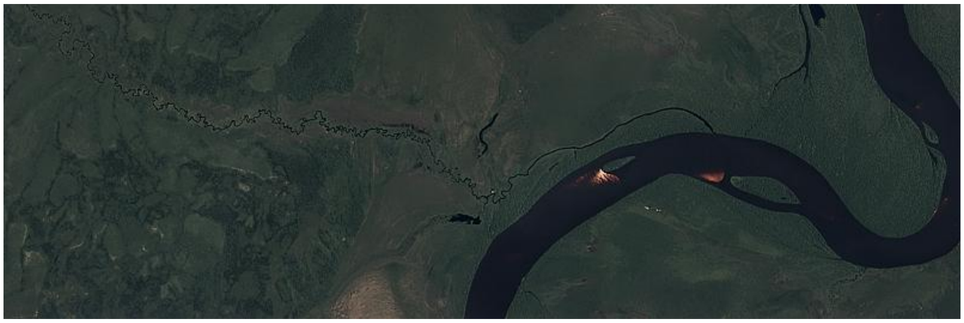


Figure 4.5: Example of an area with water

4.1.6 Habitation

Habitation is wherever there are human shelters, be it small villages or large urban areas. They usually appear as white pixels in the images, with small villages being harder to detect. Although they are not a major cause of deforestation, it is useful to monitor the expanse of the large urban areas at the expense of the greenery.

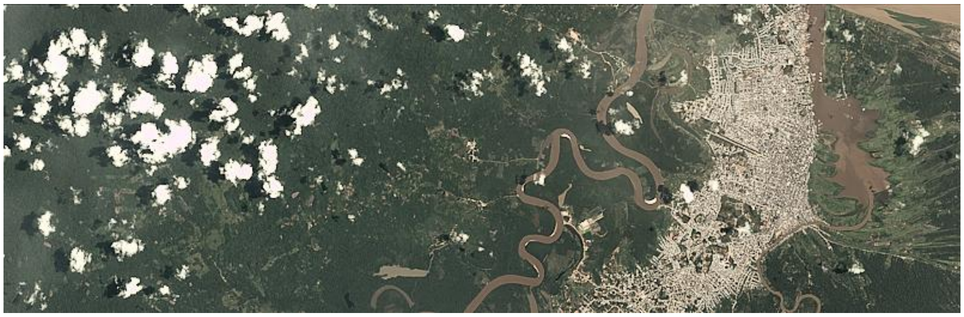


Figure 4.6: Example of an area with habitation

4.1.7 Agriculture

Agriculture is a major driver of deforestation. It consumes large chunks of forests in attempt to make more space for coffee plants and more. The problem is that all those plants that are used for human consumption filter orders of magnitude less CO2 than the large trees they replace, adding to the greenhouse gas problem, loss of oxugen, etc.

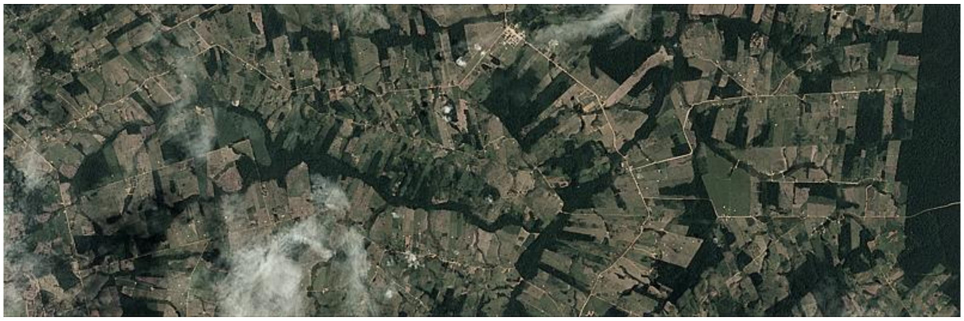


Figure 4.7: Example of an area with agriculture

4.1.8 Road

The roads by themselves are a driver of deforestation but not so much because of the land that they clear of trees during their construction. They are usually good indication of where deforestation will happen in the future, as they make the access easier for devices that are used for tree cutting. It could be for logging or creating new towns or agricultural areas.

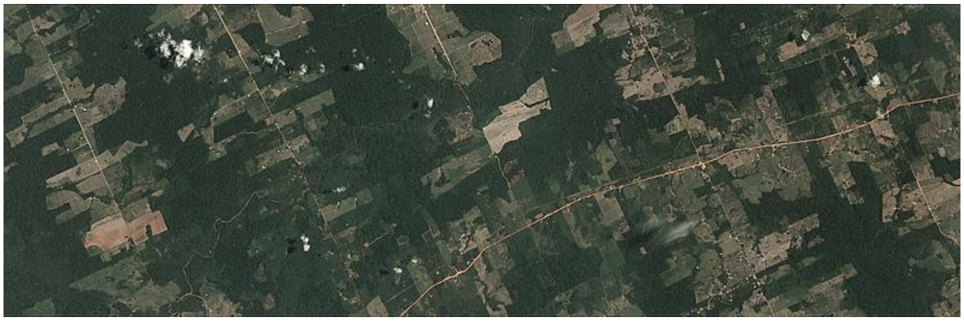


Figure 4.8: Example of an area with road

4.1.9 Cultivation

Cultivation is a subset of agriculture. It can usually be discerned from agriculture in most cases by the satellite images. It is not a major driver of deforestation as it is usually an area that families in small villages use for sustenance.

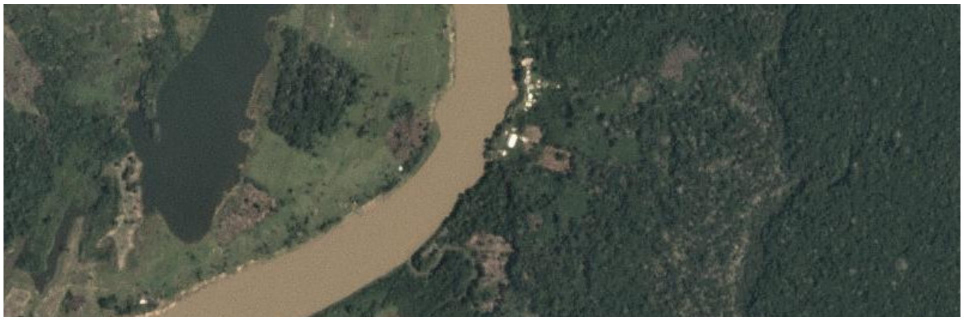


Figure 4.9: Example of an area where cultivation occurs

4.1.10 Bare Ground

This label is used to describe all forms of land that do not have any trees in them. Note that the absence of trees must be natural and not caused by humans.

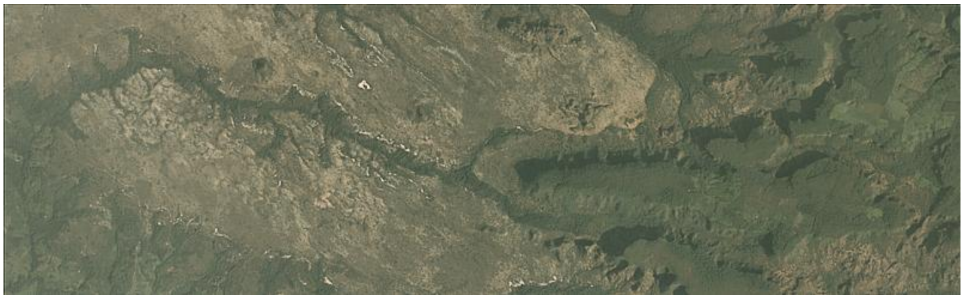


Figure 4.10: Example of an area without greenery

4.1.11 Slash and Burn

Another cause of deforestation are forest fires. Usually they leave the area black, so they are easy to detect. Sometimes the burnt trees are cut in order to allow for the faster growth of new trees. It is important to monitor those areas as it is not uncommon to have been burnt by artificially started fires in order to create an opening that can be later exploited by companies.

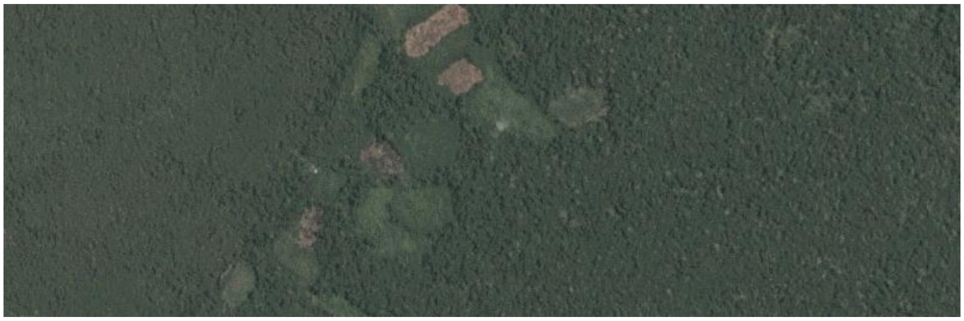


Figure 4.11: Example of an area that lost greenery due to fires

4.1.12 Selective Logging

Selective logging is the legal form of logging, where only selected trees can be cut. They usually are the high value trees, and the amount is regulated.

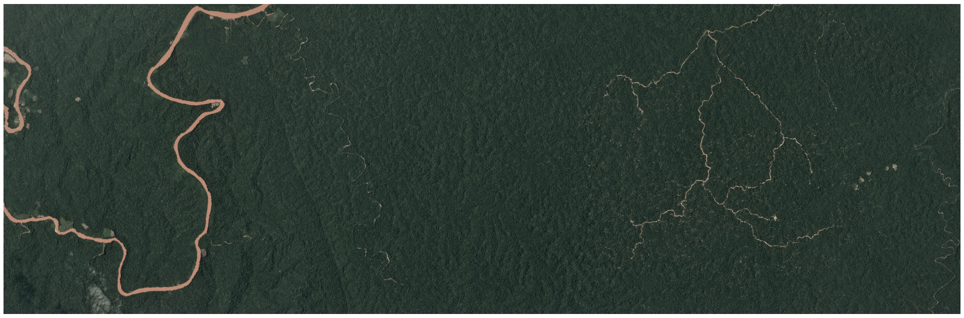


Figure 4.12: Example of an area that is used for logging

4.1.13 Blooming

Although, most of the blooming cannot be seen from space, the most extreme of the instances can. Large trees bloom, fruit, and flower at the same time to maximize the chances of cross pollination.

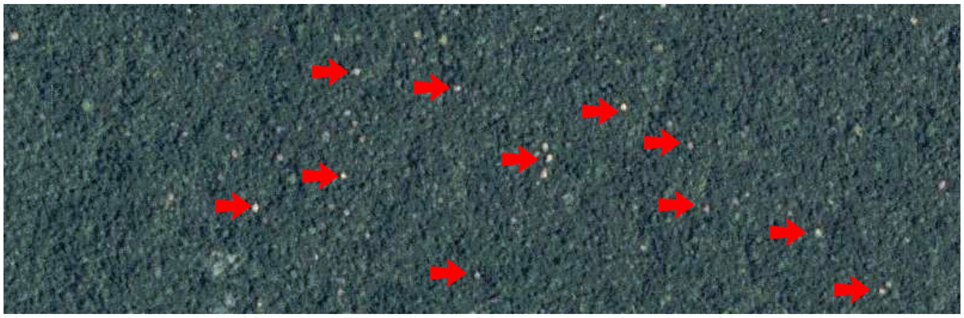


Figure 4.13: Example of an area where large trees will bloom

4.1.14 Conventional Mining

There is a great amount of resources available in the ground under the Amazon. This label follows the legal mining operation that although expanding, are a controllable cause of deforestation.

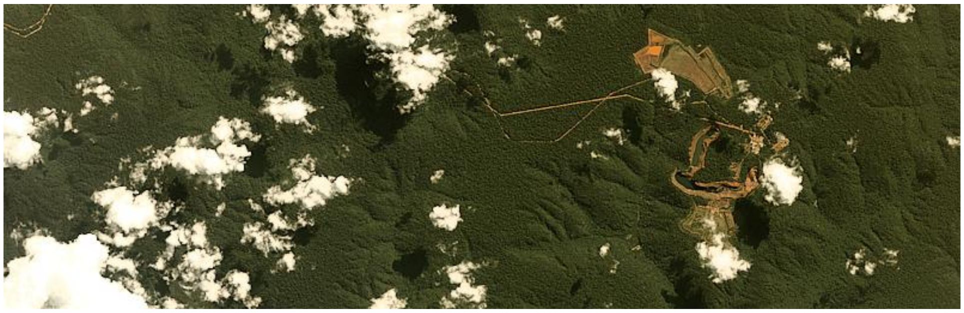


Figure 4.14: Example of an area that is used for traditional mineral mining

4.1.15 Artisinal Mining

This label is mostly used for small scale mining operations for gold, most prevalent at the foothills of Andes. In this activity sometimes illegal workers partake. The ways the valuable minerals are mined there, require the used of other heavy minerals such as mercury, which is extremely harmful to the forest. The whole process leaves all nearby areas barren for many centuries.

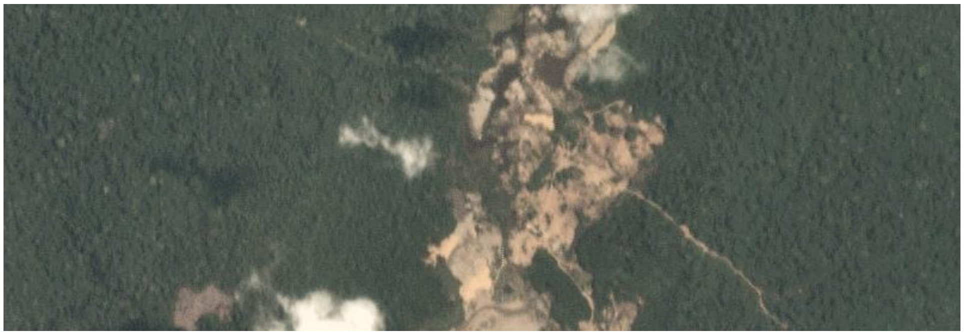


Figure 4.15: Example of an area where illegal mining of heavy elements is taking place

4.1.16 Blow Down

A natural phenomenon where high speed cold air that stems from the Andes, blasts the large trees, leaving the area open. The open area recovers fast, as other plants rush in the open space to take advantage of the sunlight.

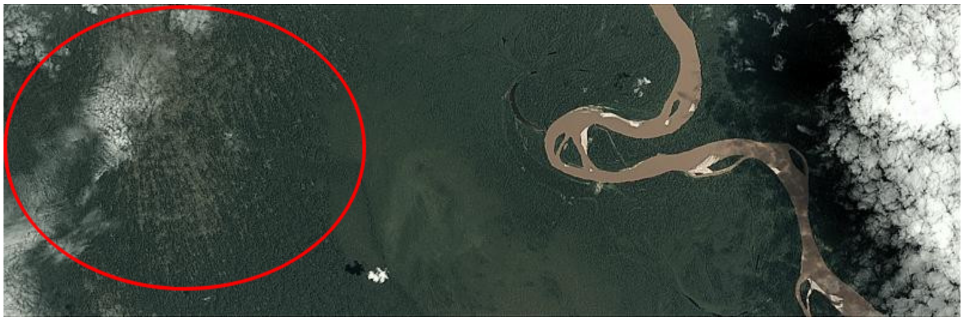


Figure 4.16: Example of an area where extreme natural phenomena destroy the trees

4.1.17 Dataset Samples

Image gallery 4.17 depicts how some of the images that are available to us look like, matched with one or more labels that describe them.

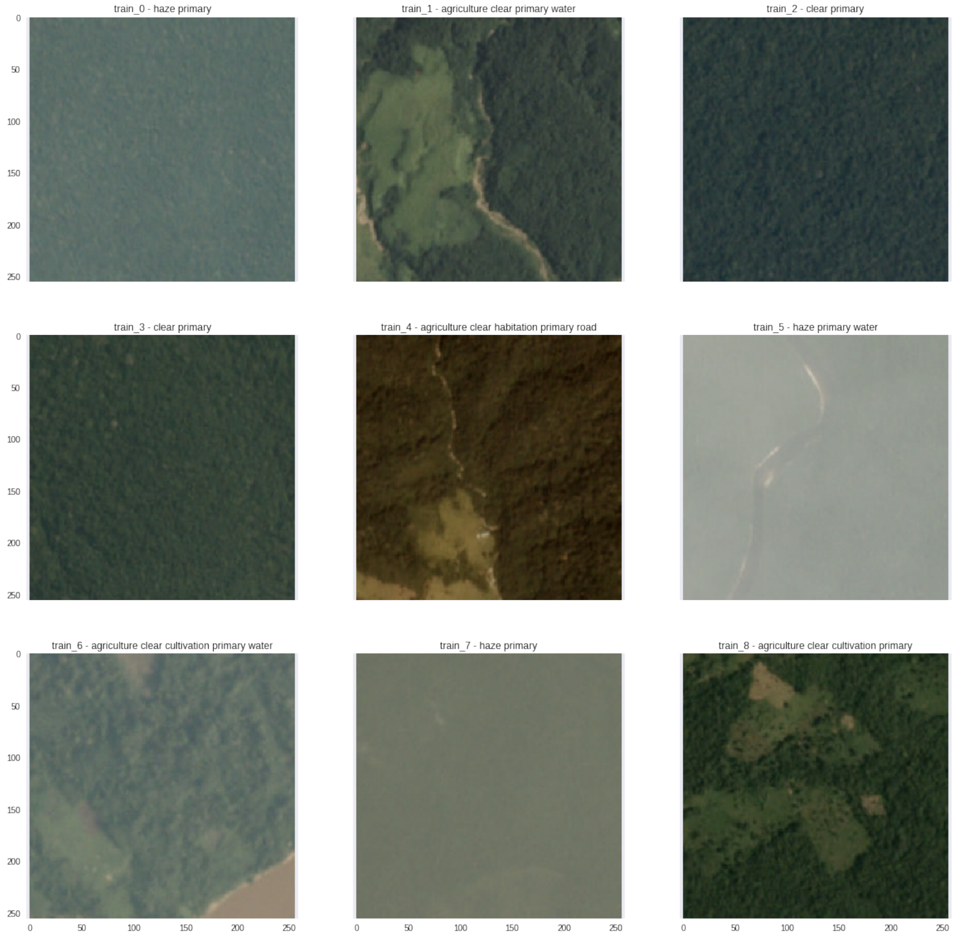

train_0 - haze primary
train_1 - agriculture clear primary water
train_2 - clear primary
train_3 - clear primary
train_4 - agriculture clear habitation primary road
train_5 - haze primary water
train_6 - agriculture clear cultivation primary water
train_7 - haze primary
train_8 - agriculture clear cultivation primary
0
50
100
150
200
250

Figure 4.17: RGB images of the Amazon forest

4.2 Experimental Setup

4.2.1 Section structure

Before starting this section, it is worth discussing the structure that each experiment will follow.

We first state the specifics, meaning how the dataset was split, the resolution of the images in which it was trained and the model specific hyperparameters. Afterwards, we examine the training phase of the model by showcasing a plot of the loss versus epoch. Then we delve into the specifics of the output, by analyzing the results into a heatmap for each one of our classes.

In an attempt to further explain the output beyond its numerical substance, we attempt to depict the functionality inside the network that lead to those results. To achieve this, we first pick the image that was correctly classified with the most confidence for each class (placed on the left). Then, we highlight the areas of the image that had the bigger role in this decision (placed in the middle). Lastly, we extract a map showing the parts of the last convolutional layer that were excited prior to this decision (placed on the right). Thereafter, we discuss any notable observations.

It is important to note that while the chosen images achieved high confidence for their respective class, most of those images contain multiple labels and thus, we expect to see more highlight or excitations that correspond to those other labels. Also, the Vision Transformer does not contain any convolution in the traditional sense, so an experimental version of this technique was used (we will explain it in the Vision Transformer experiment) that might provide us with results that are not directly comparable to those of the other networks or even accurate. Finally, a section will follow that cross-examines the results between all of our experiments.

4.2.2 VGG16

Experiment preparations

Table 4.1 lists the hyperparameters used for the multi-label experiment using the VGG16. As the name of the architecture implies, it consists of 16 layers. The resolution chosen is 256x256, which is the maximum resolution available for the given images. The batch size of choice was the largest possible that fit the GPU, which was 64. The dataset of 40k images was split into 50/20/30 training/validation/test sets. Its simplicity as well as its good performance in general, makes it a prime candidate to be the baseline model

Resolution	256x256
Layers	16
Batch size	32

Table 4.1: Hyperparameter setup for the multi-label experiment using the VGG16.

Training process

In figure 4.18, we can observe the training process for the classification. Due to our limited resources early stopping was implemented with patience 4. It seems that convergence between the training and validation score is happening from the 17th epoch onwards. Also, we reduce the learning rate in each plateau with patience 3 in order to make sure that the model will rapidly train during its first stages and then it will slow down during its last stages, in order to learn as many details as possible. This explains the incremental drop in the loss volatility during the training phase.

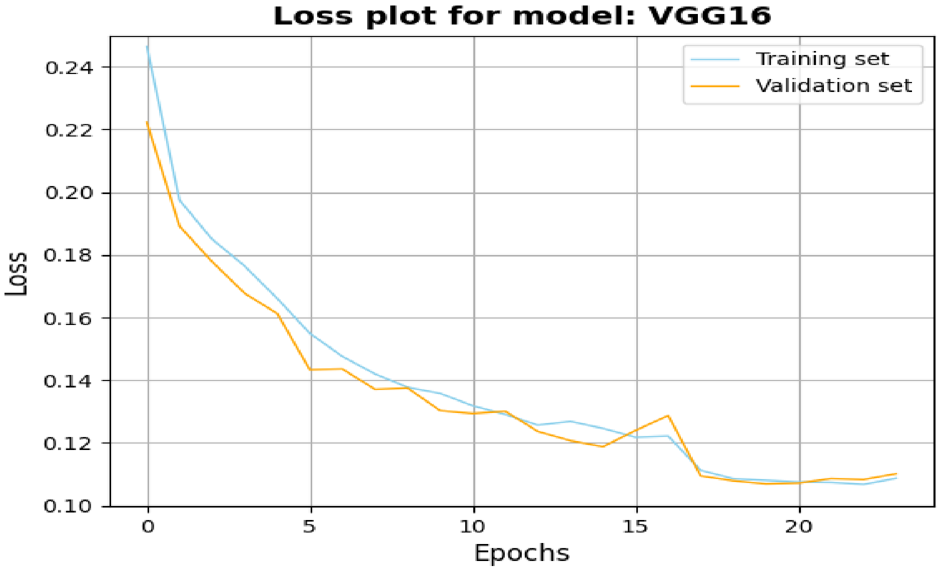


Figure 4.18: Training and validation loss during the model's training phase

Experiment results

Figures 4.19 and 4.20 show that haze, agriculture, water, habitation, road and cultivation were some widely used labels that the model tended to miss-classify. The model also did not have a lot of success in trying to classify the rare labels. Nevertheless, it had success in determining the image clarity labels.

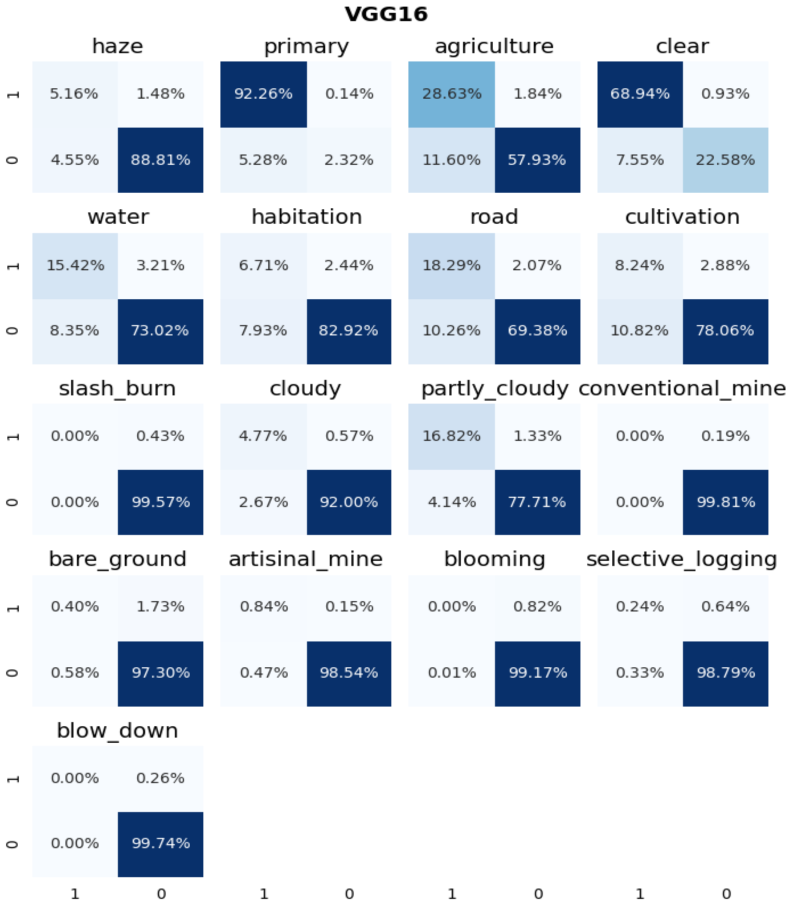


Figure 4.19: Relative performance for each individual label

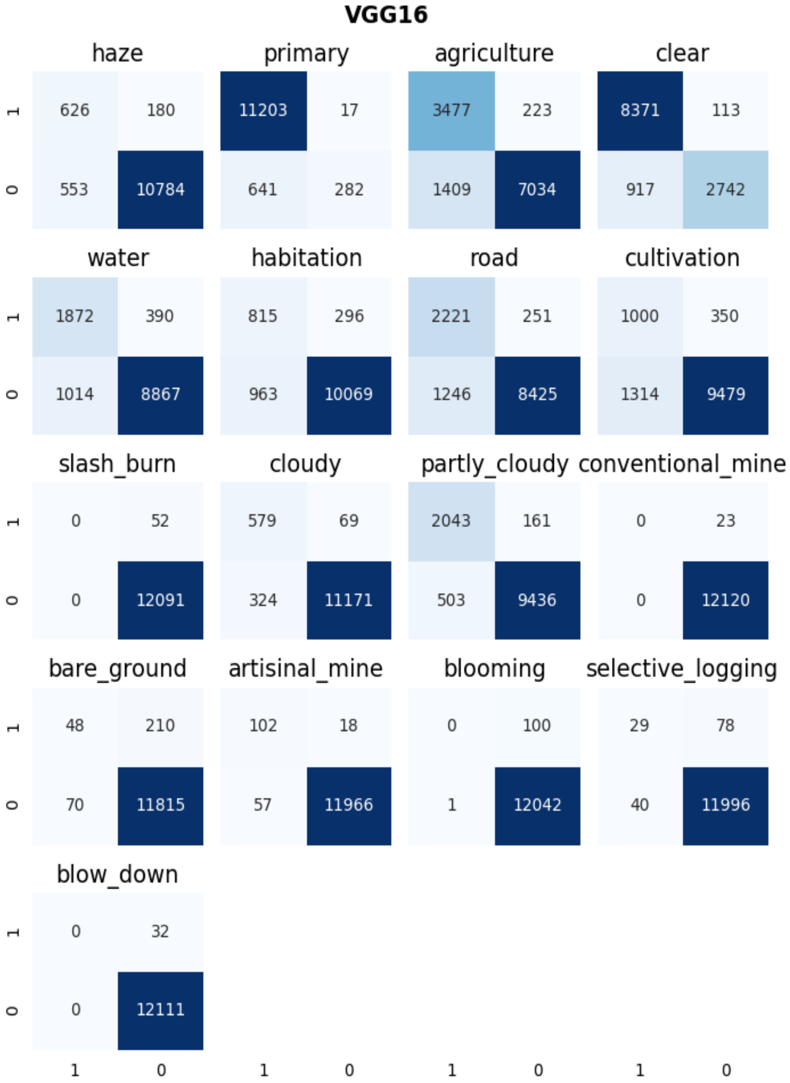


Figure 4.20: Absolute performance for each individual label

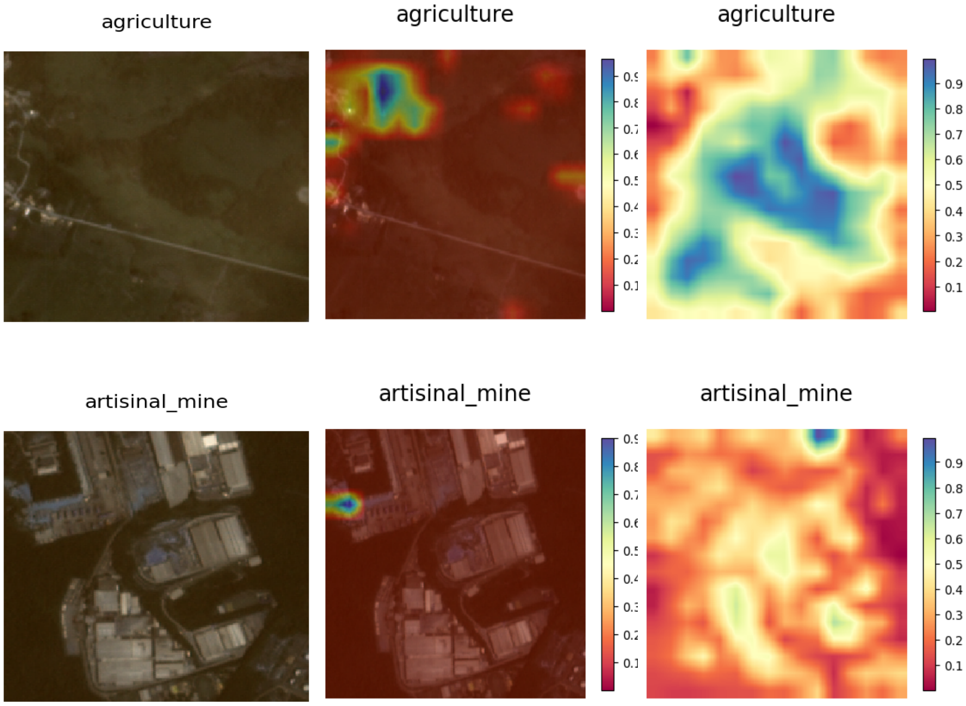
Possible explanations for the results

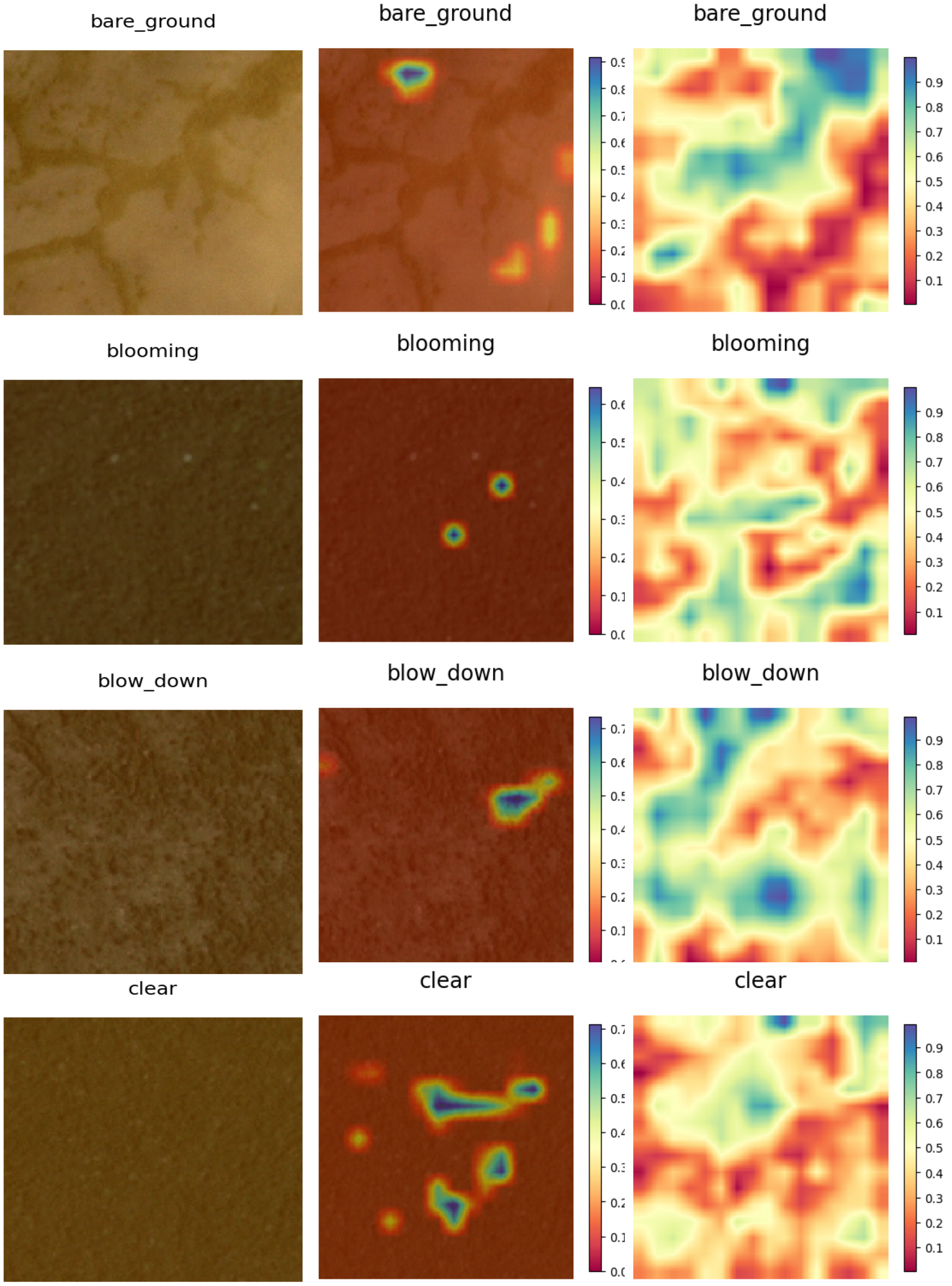
Starting with the weather labels, we can observe that the model created a good understanding for most of them. The gradcam for partly cloudy in particular shows that the activation functions were spot on throughout the model and in the final layer a very good representation arrived. Also, model decided to output uniform values for usually uniform images such as haze. In the case of other uniform images such as cloudy and clear, the model activated in a lot of arbitrary places which also works towards differentiating them from more detailed labels.

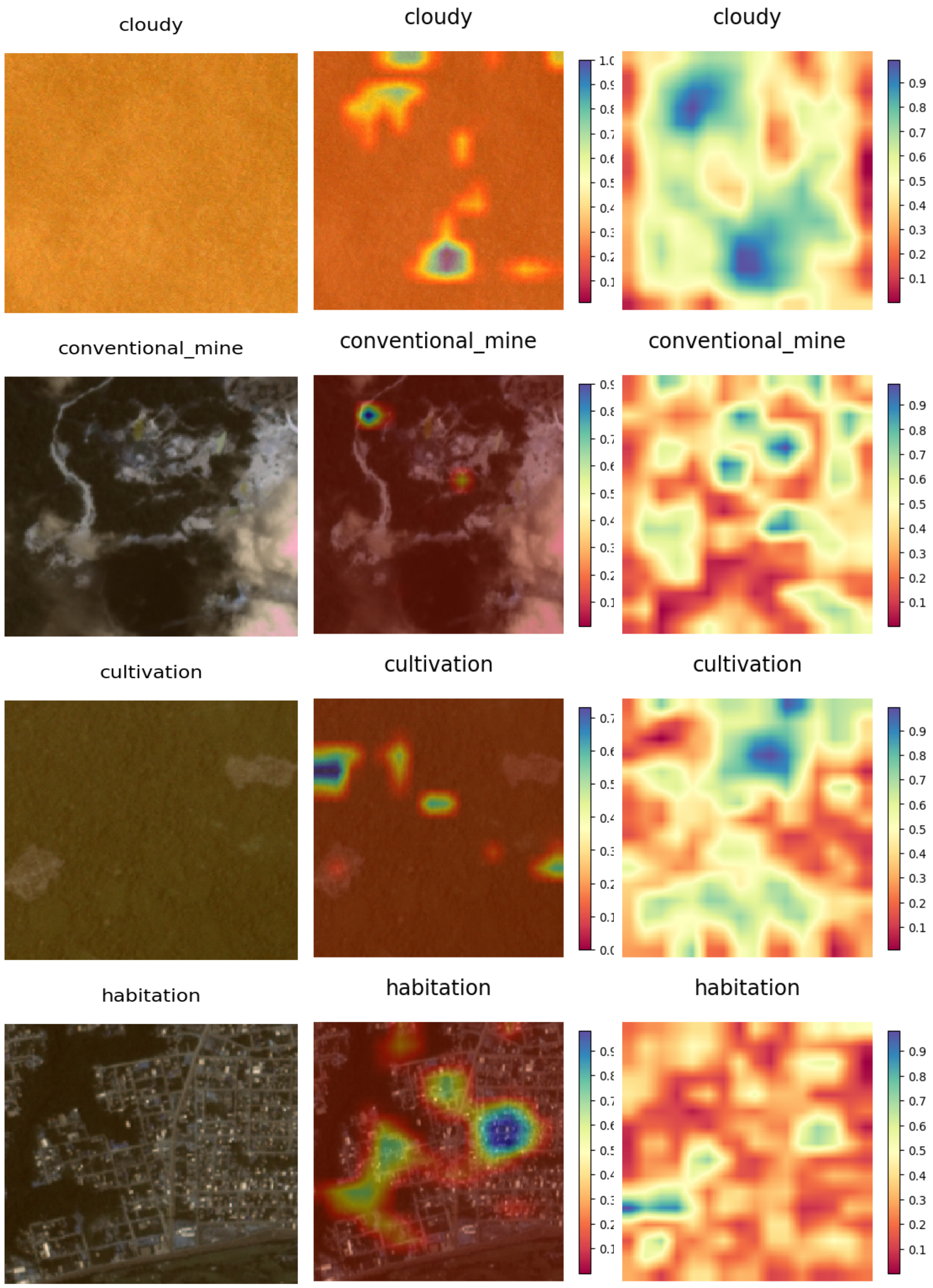
Moving on to the landscapes, this was another group in which the model performed adequately. Primary was represented with a lot of arbitrary activations as most uniform image labels. Water was detected by the network, which focused in the middle of large water bodies. What is more interesting is that the last activation layer took a form that matches exactcly the lakes and rivers in the area. However, selective logging and cultivation seem to have the network firing in random spots without interest both in the gradcam and the final layer explaining the model's poor performance in those labels.

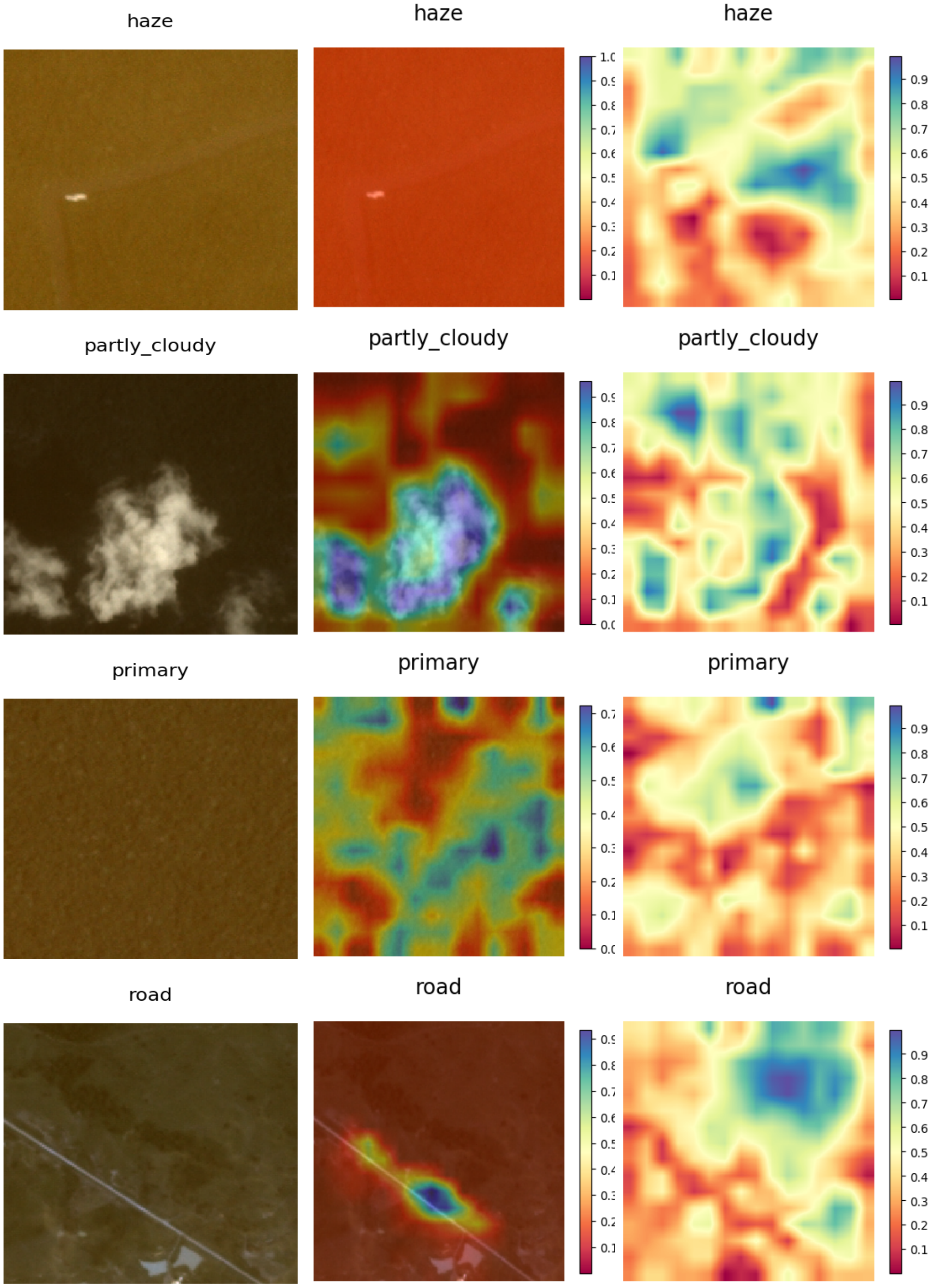
With regards to the infrastructure, the precision of the model for the road stands out. Also note that the last layer represents the road with close to 0 values which is quite common for networks to do as they just want to map numerical spaces with labels without caring if the values they associate them with are high or low (the blue in the image might be for another label such as primary). The same mapping is happening with agriculture which is successfully pinpointed by the model. Another interesting pattern is that of habitation where the activations point in its center but the last layer has found that almost all of the image is habitation in which values in the middle are assigned. Habitation, agriculture and road achieve great performance fairly.

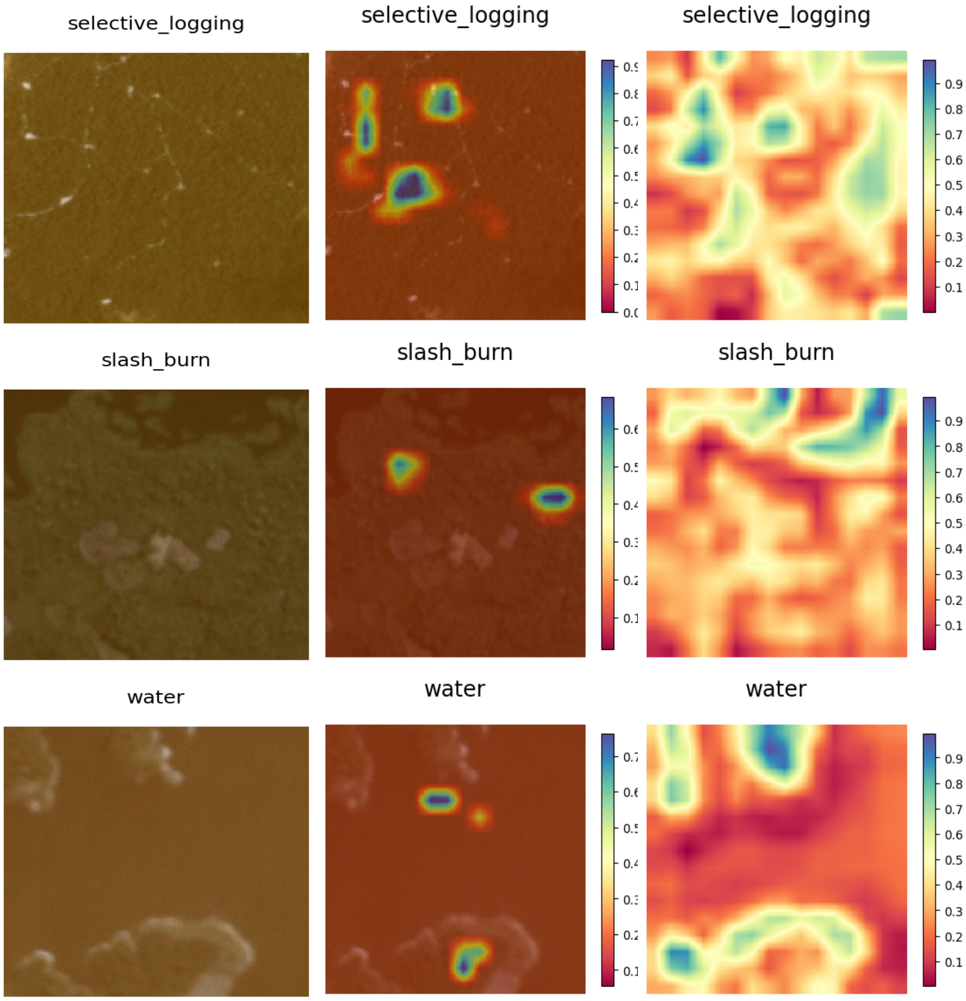
Lastly, we can observe that the model achieves poor performance across most rare labels. Blooming, blow down and slash burn invoke small and precise activations in the model that do not pinpoint towards where the actual labels are. Furthermore, they all translate to seemingly arbitrary values in the last layer. Artisinal mine is the only exception where the mine is located intensely and precisely leading to a somewhat better score for the label.




bare_ground
bare_ground
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
bare_ground
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
blooming
blooming
0.6
0.5
0.4
0.3
0.2
0.1
0.0
blooming
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
blow_down
blow_down
0.7
0.6
0.5
0.4
0.3
0.2
0.1
blow_down
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
clear
clear
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
clear
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


cloudy
cloudy
cloudy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
conventional_mine
conventional_mine
conventional_mine
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
cultivation
cultivation
cultivation
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
habitation
habitation
habitation
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


haze
haze
haze
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
partly_cloudy
partly_cloudy
partly_cloudy
primary
primary
primary
road
road
road


selective_logging
selective_logging
selective_logging
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
slash_burn
slash_burn
slash_burn
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
water
water
water
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

4.2.3 EfficientNet

Experiment preparations

Table 4.2 lists the hyperparameters used for the multi-label experiment using the EfficientNet. Width and Depth coefficients are set to indicate an EfficientNet of B0 complexity. The default resolution for the specified coefficients is 256x256 which is the maximum resolution available for the given images. The reason for choosing the B0 model architecture is that through the tuning process, it was discovered that models of higher complexity do not necessarily produce better results for the dataset and in most cases their generalization ability is actually lower than their less complex counterparts. That said, the higher the resolution of the images that were fed to the model, the better the performance. So, it made a lot of sense to choose this specific setup given the limited computational resources available (GTX 1060 6GB). The batch size of choice was the largest possible that fit the GPU which was 16. The dataset of 40k images was split into 50/20/30 training/validation/test sets.

Resolution	256x256
Width coeff	1.0
Depth coeff	1.0
Batch size	16

Table 4.2: Hyperparameter setup for the multi-label experiment using the EfficientNet.

Training process

In figure 4.21, we can observe the training process for the classification. Due to our limited resources early stopping was implemented with patience 5. It seems that convergence between the training and validation score is happening from the 15th epoch onwards. Also, we reduce the learning rate in each plateau with patience 2, in order to make sure that the model will rapidly train during its first stages and then it will slow down during its last stages, in order to learn as many details as possible. This explains the incremental drop in the loss volatility during the training phase.

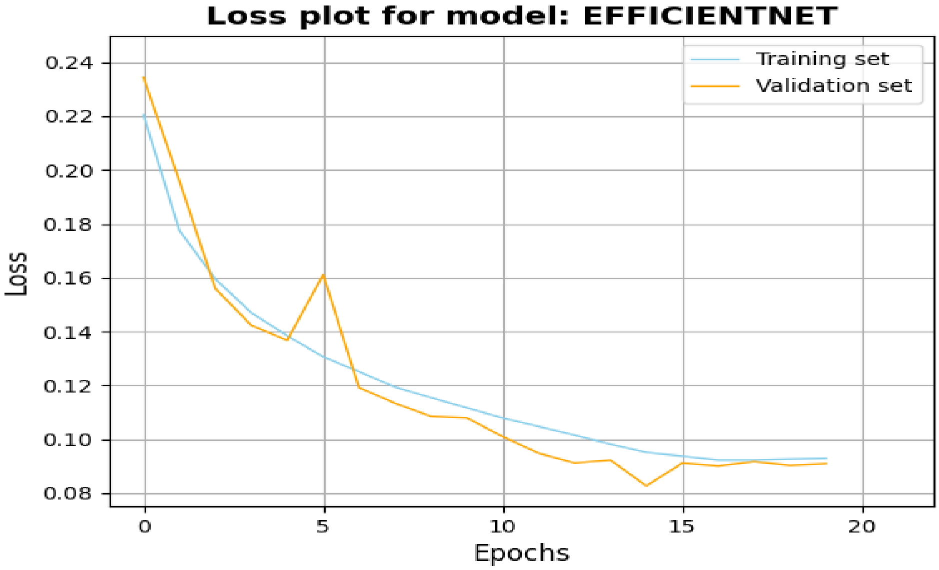


Figure 4.21: Training and validation loss during the model's training phase

Experiment results

Figures 4.22 and 4.23 show that agriculture, water, habitation and cultivation were some widely used labels that the model tended to miss-classify. On the positive side, it seems that the model tried and had some success in classifying some rare labels instead of overlooking them completely.

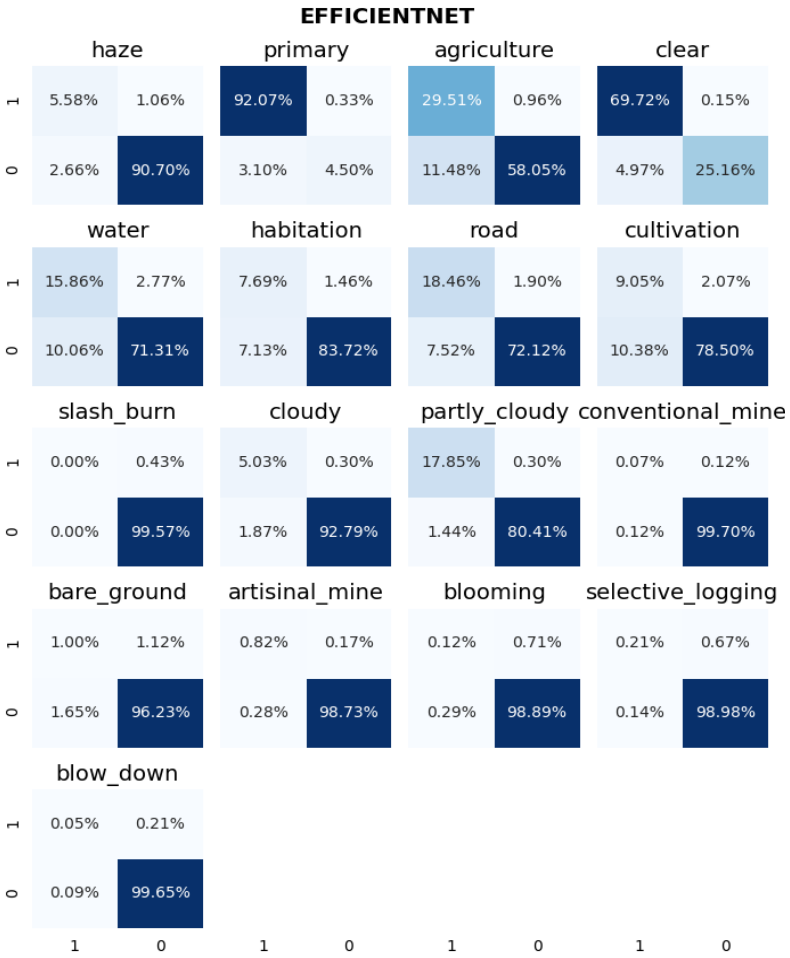


Figure 4.22: Relative performance for each individual label

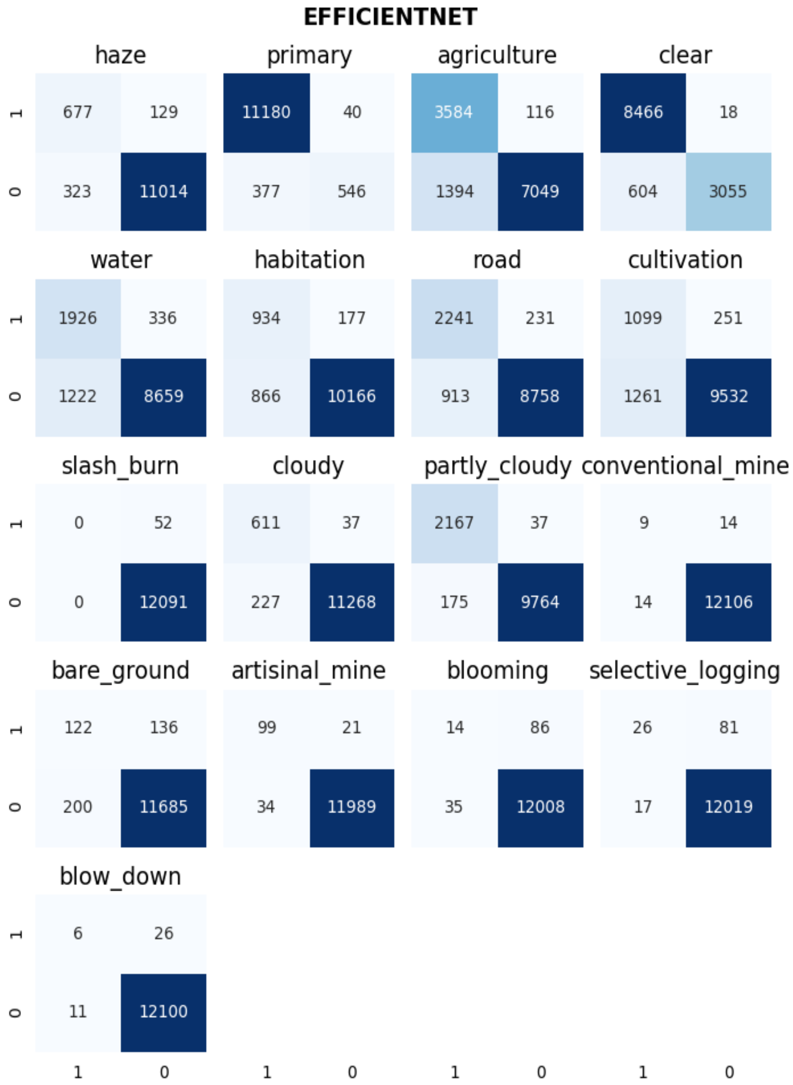


Figure 4.23: Absolute performance for each individual label

Possible explanations for the results

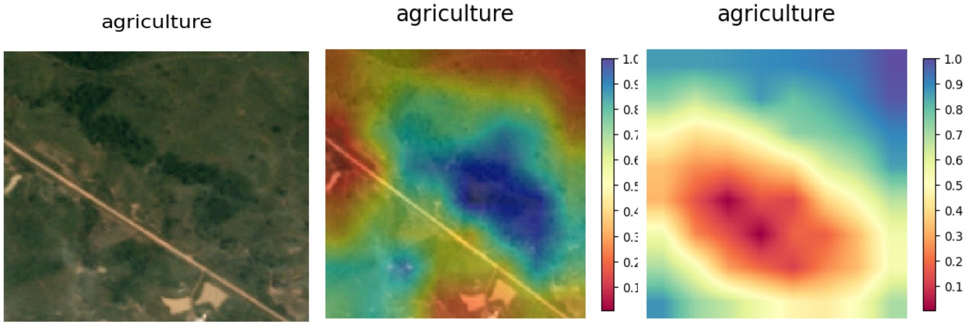
To start with, we can generally observe that the last layer of the efficientnet, adopted in large, close to zero values for the areas of interest. So for the last layer, when we refer to the term activation, we talk about the red spots. Also, the gradcam output and the contents of the last layer seem to agree together for the majority of the inputs.

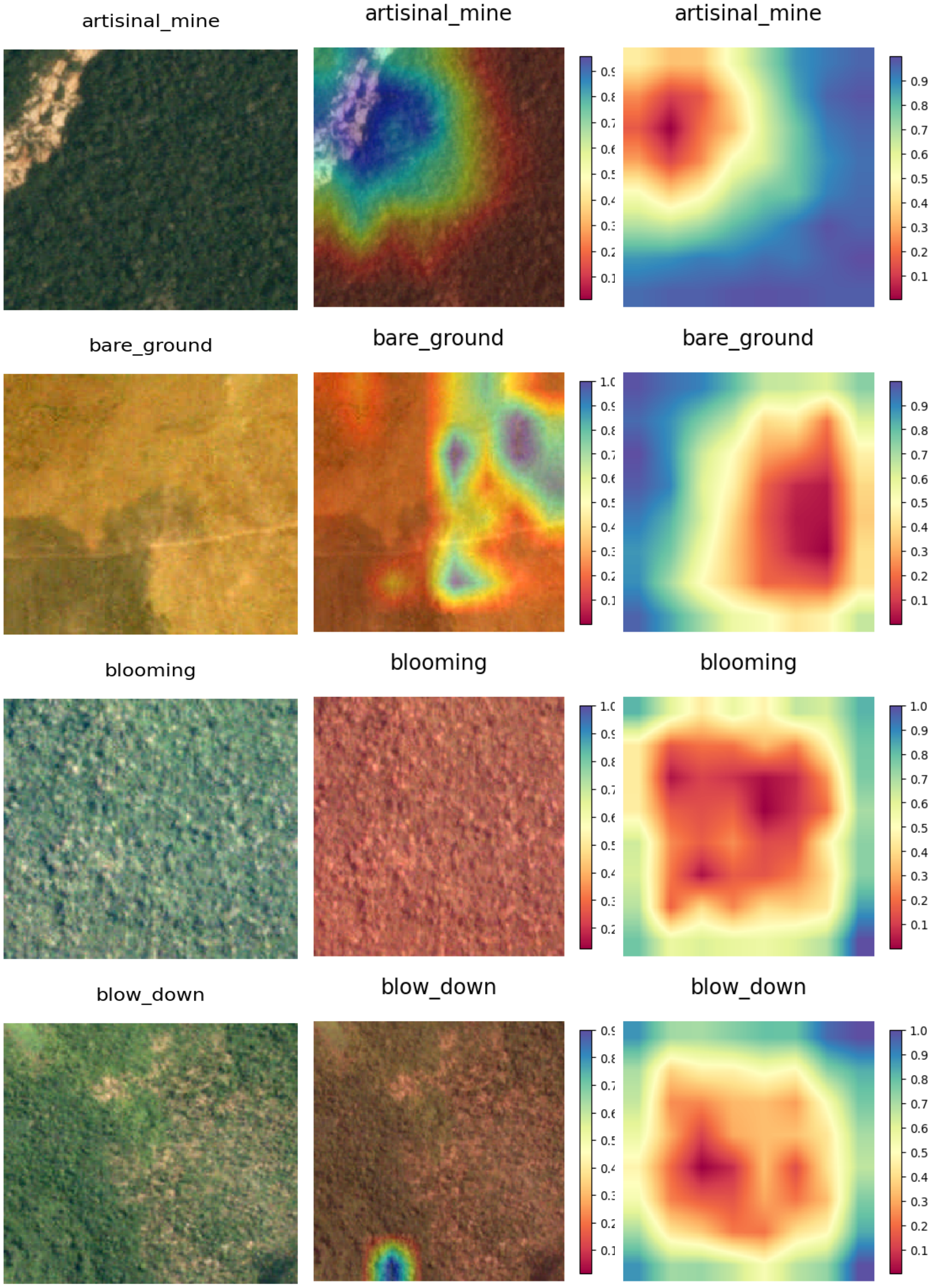
When it comes to the weather labels, which are mostly uniform elements, the model mapped them with uniform values. An interesting fact is that most of the uniform labels, where depicted as a square in the center of the last layer, with the main activation being a different red spot inside the square for each corresponding label. As such, the model was able to accurately map uniform activation to labels in an orderly manner. Partly cloudy was located accurately and the particular image for haze might have confused the model a bit due to the various shapes inside it. Nevertheless, the image for haze was depicted in the same manner as other labels with uniform values, so we can assume that the model understood it. In general, the model understood how to handle the weather labels, achieving great performance in the process.

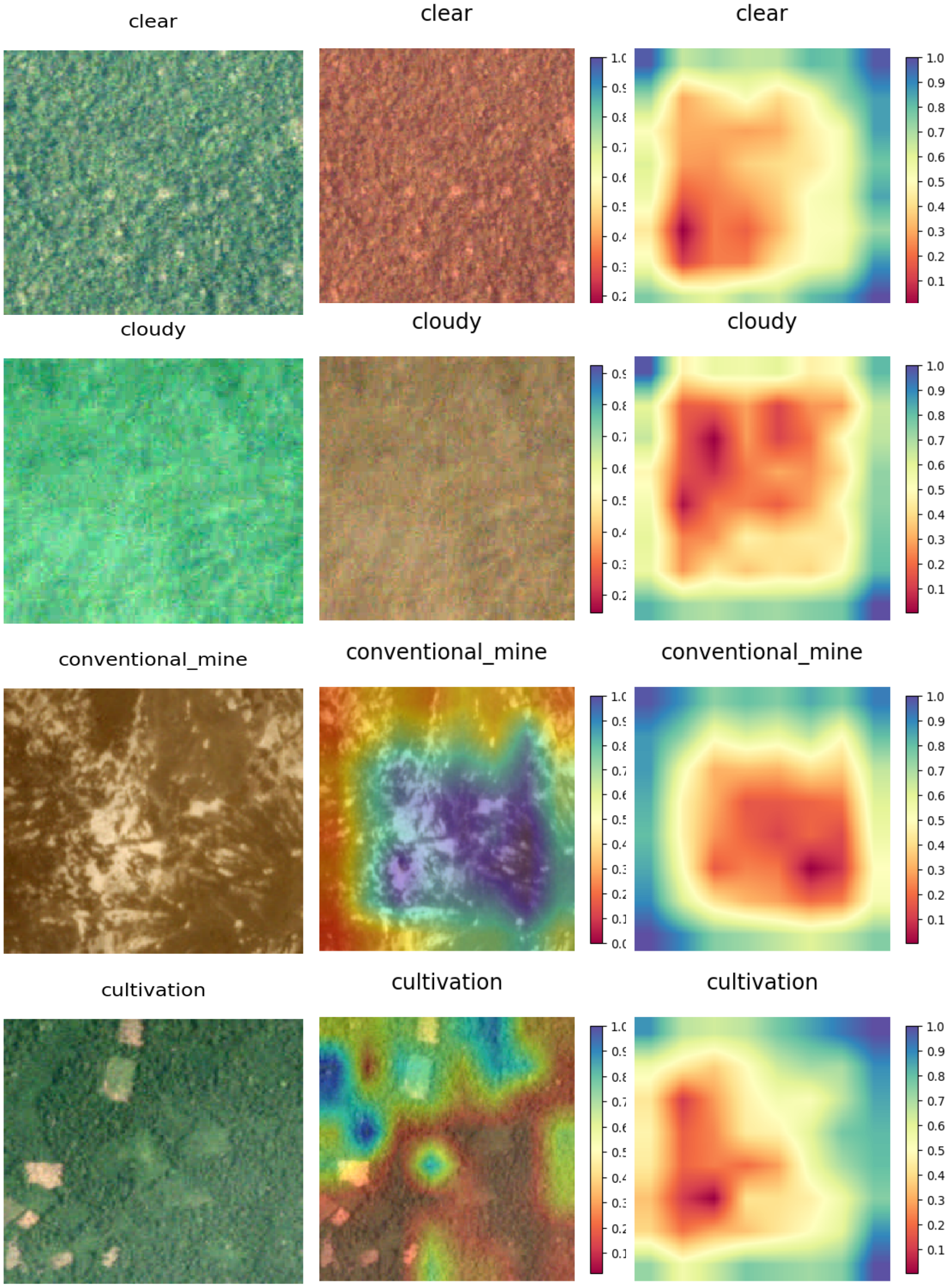
Moving on to the landscapes, a group that generally caused problems for all the models, primary seems to be treated correctly as a uniform label. Selective logging seems to have caused an all encompassing activation that surrounds the area of interest, which is also reflected in the last layer of the model. Furthermore, the cultivation seems to not be understood correctly due to the arbitrary activations it is subject to. The same can be said for the water label, albeit the image in question has got a shape and color not typical among water images. Lastly, while cultivation and selective logging did not perform well, the landscapes where in overall classified more accurately than they were from other models.

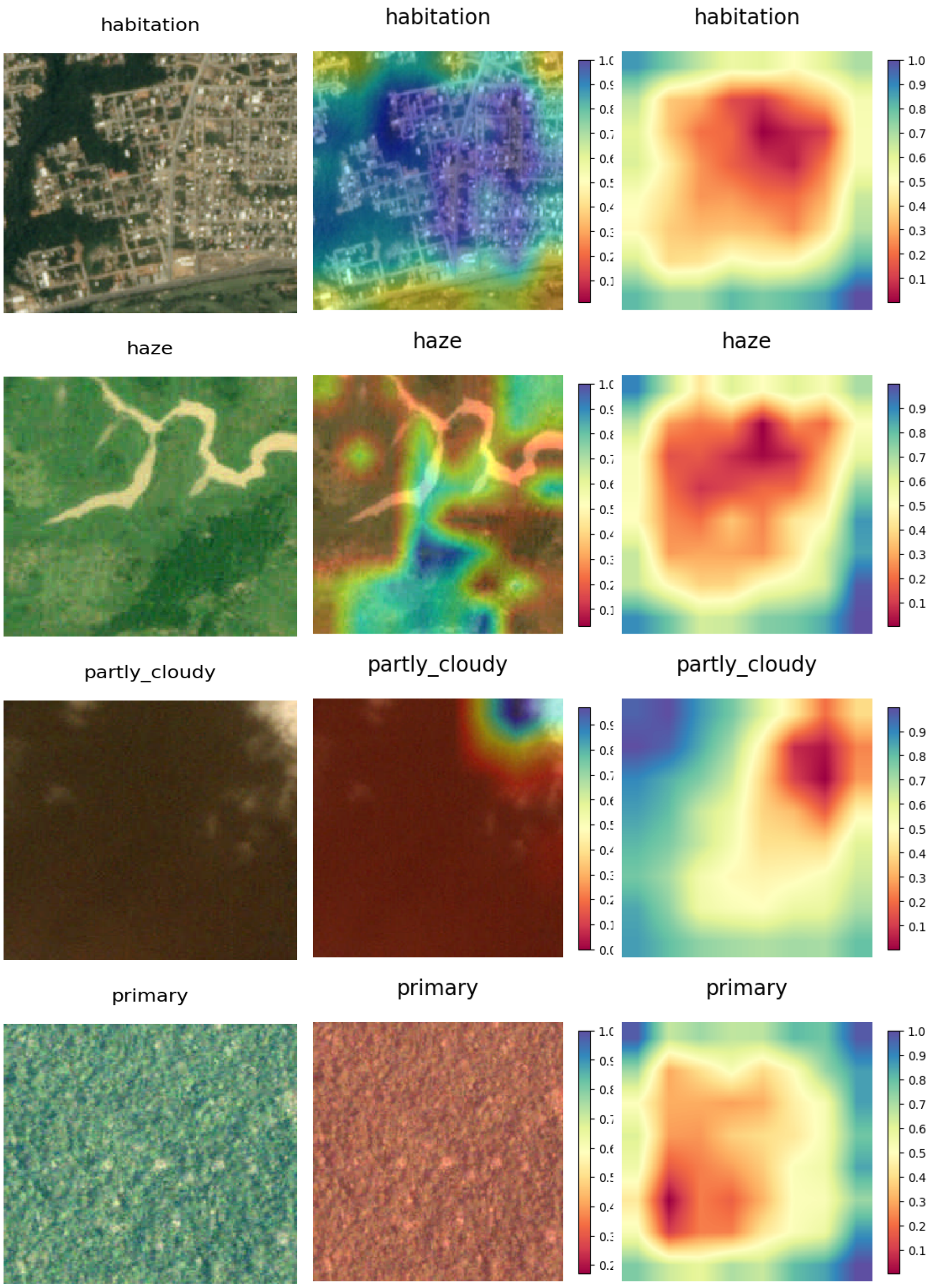
With regards to the infrastructure, the same image emerged as the most representative for both habitation and road. This seems logical as it is heavy in infrastructure and roads, probably activating the model from multiple perspectives. The same goes for the image in agriculture, where most of the image hosts components of that label causing it to activate heavily.

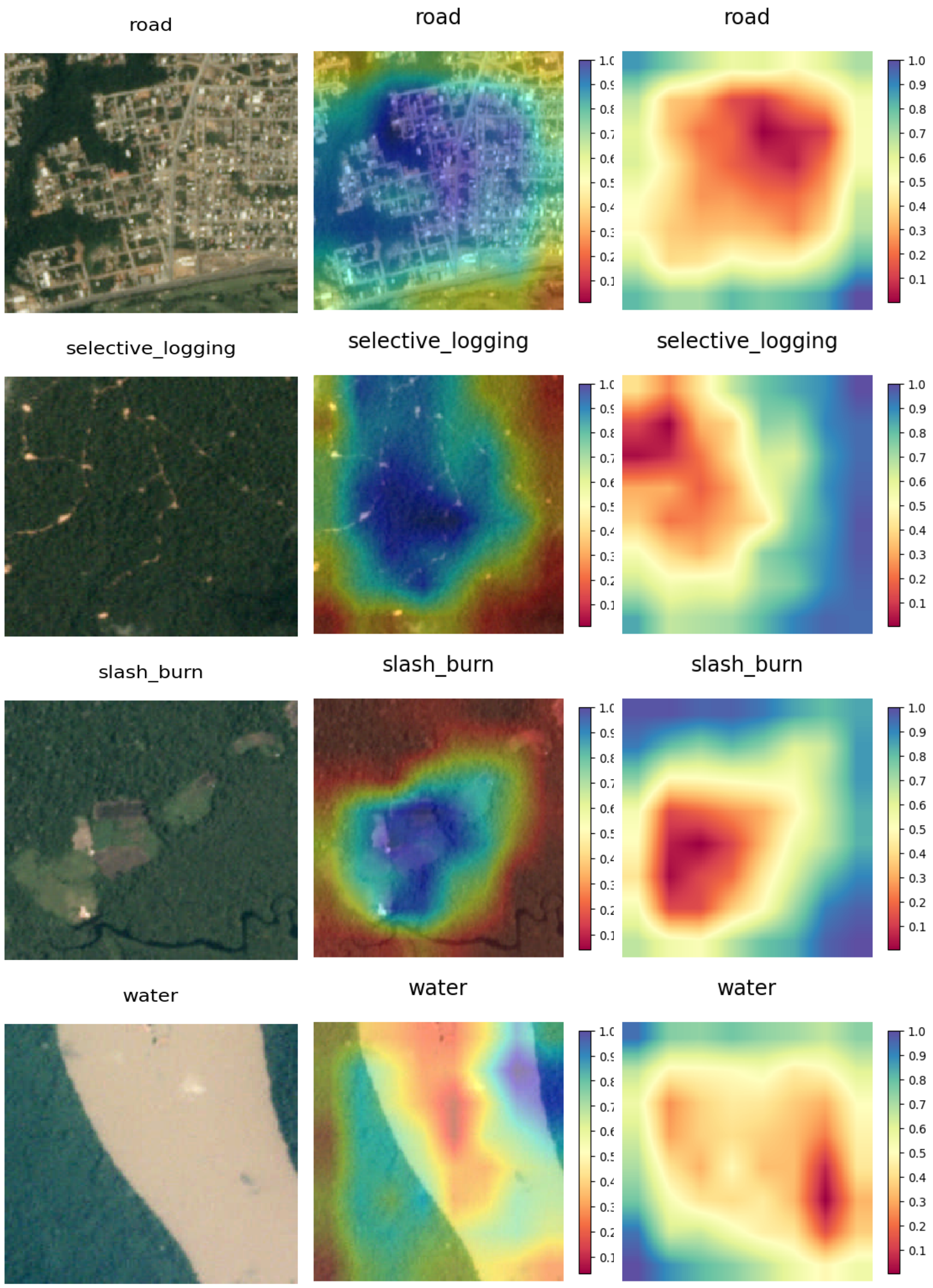
Lastly, a great differentiator of this model with regards to the other models is that it succeeded in classifying correctly some of the rare labels. Rare labels don't boost the performance by much and thus, they are mostly neglected by the models. This is not the case in this experiment. Artisinal and conventional mines were accurately spotted, which is also reflected in the contents of the last layer. Blow down seems to have caused precise activations instead of the required wide ones and blooming has the opposite problem. Nevertheless, the model still scored some without probably understanding their meaning in full. Slash burn was not detected at with the activations we see being around habitation.




artisinal_mine
artisinal_mine
artisinal_mine
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
bare_ground
bare_ground
bare_ground
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
blooming
blooming
blooming
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
blow_down
blow_down
blow_down
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


clear
clear
clear
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
cloudy
cloudy
cloudy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
conventional_mine
conventional_mine
conventional_mine
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
cultivation
cultivation
cultivation
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


habitation
habitation
habitation
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
haze
haze
haze
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
partly_cloudy
partly_cloudy
partly_cloudy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
primary
primary
primary
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


road
road
road
selective_logging
selective_logging
selective_logging
slash_burn
slash_burn
slash_burn
water
water
water
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

4.2.4 Vision Transformer

Experiment preparations

Table 4.3 lists the hyperparameters used for the multi-label experiment using the Vision Transformer. The resolution is 256x256 which is the maximum resolution available for the given images. The patch size is 16 which means that the original image will be split into 256 16x16 images. Those patches will pass through 4 Transformer layers. Each Transformer layer's attention layer will have 8 heads, which means that the context of each patch will be measured from 8 different angles which ideally should be different. The final embedding will have a dimension of 1024. Those specifications, albeit derived from manual tuning, seem relatively close to the original paper's recommended model parameters. The batch size of choice was the largest possible that fit the GPU which was 16. The dataset of 40k images was split into 50/20/30 training/validation/test sets.

Resolution	256x256
Patch size	16
Transformer layers	4
Number of heads	8
Batch size	16

Table 4.3: Hyperparameter setup for the multi-label experiment using the Vision Transofrmer.

Training process

In figure 4.24, we can observe the training process for the classification. Due to our limited resources early stopping was implemented with patience 4. It seems that convergence between the training and validation score is happening from the 28th epoch onwards. Also, we reduce the learning rate in each plateau with patience 2 in order to make sure that the model will rapidly train during its first stages and then it will slow down during its last stages in order to learn as many details as possible. The training phase of the model seems relatively stable till the end, with its validation performance somewhat better than the training set's, mainly due to the lack of transformations taking place in the validation set.

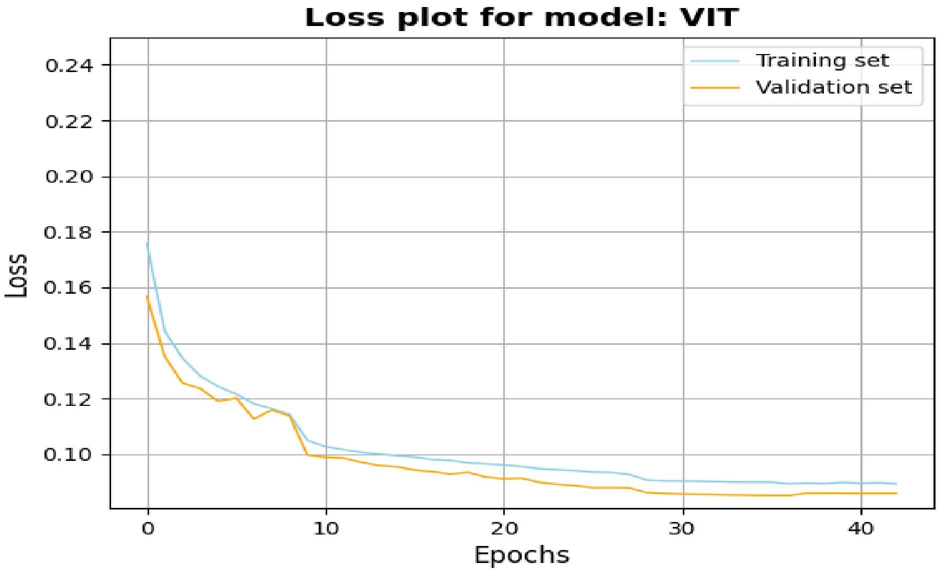


Figure 4.24: Training and validation loss during the model's training phase

Experiment results

Figures 4.25 and 4.26 show that the results are pretty identical to those of the EfficientNet. In some rare labels such as bare ground it performs even better.

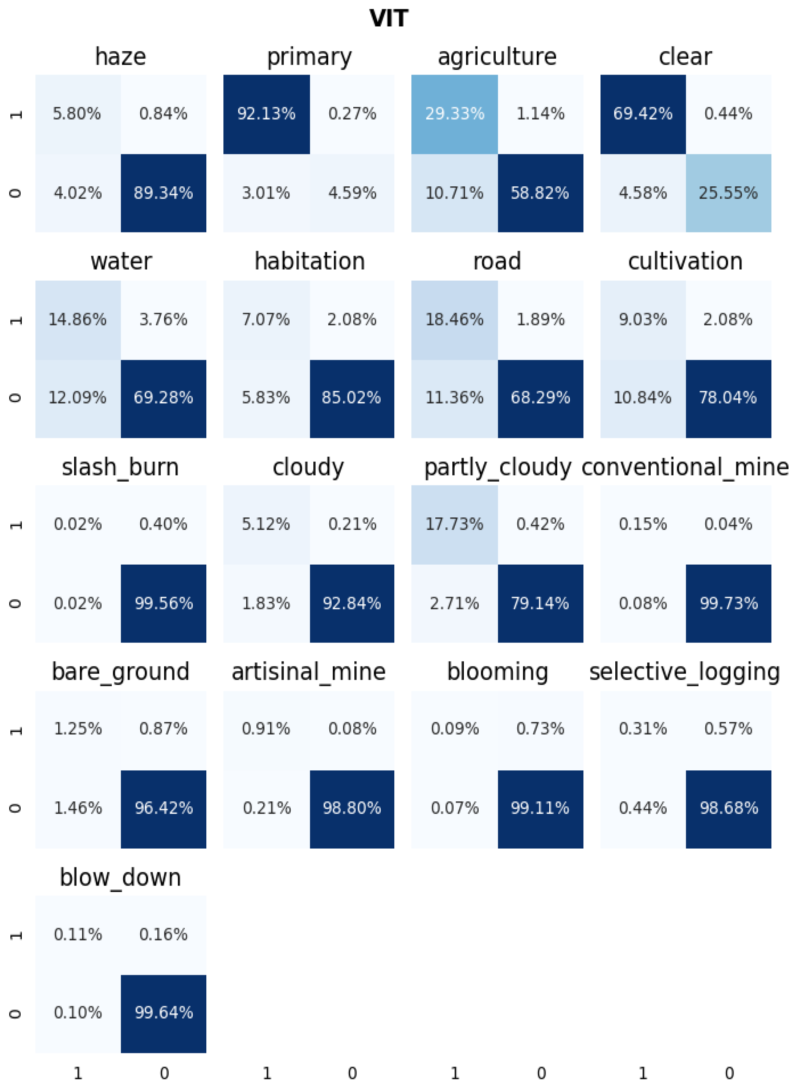


Figure 4.25: Relative performance for each individual label

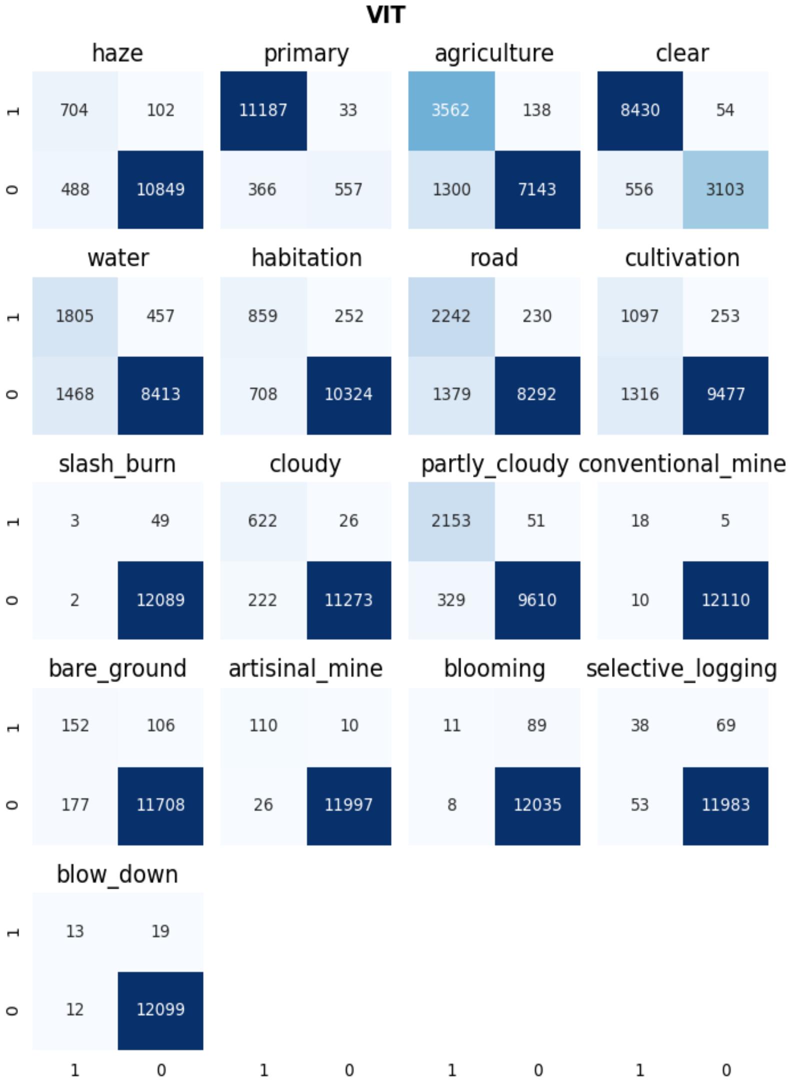


Figure 4.26: Absolute performance for each individual label

Possible explanations for the results results

Before start explaining the results, as we foretold the technique that was used to create the Grad-Cam visualizations and excitations slightly differed from the previous ones. Vision transformer makes heavy use of the Transformer layers instead of the convolutional ones. As mentioned in the methods section, Those layers accept the image in patches in a stacked form as the query and output the answer to the query in a new row of neurons. Those stacked patches could be seen as refined pieces of the original image. Due to this property, we reshape the stacked patches into 16x16 images under the assumption that each patch carries all the information of the original space of the image that it was clipped from.

Using the above technique, the resulting Grad-Cam images produced take extreme values in the upper right corner and/or uniform values that do not give us a lot of information. The reason is that Grad-Cam is not a technique compatible with this type of network. Nevertheless, sometimes the output makes sense and we can derive some insights from it. An interesting fact is that as we will showcase later, the last layer's output that we assembled from the patches, is a quite accurate representation of what is happening in each image.

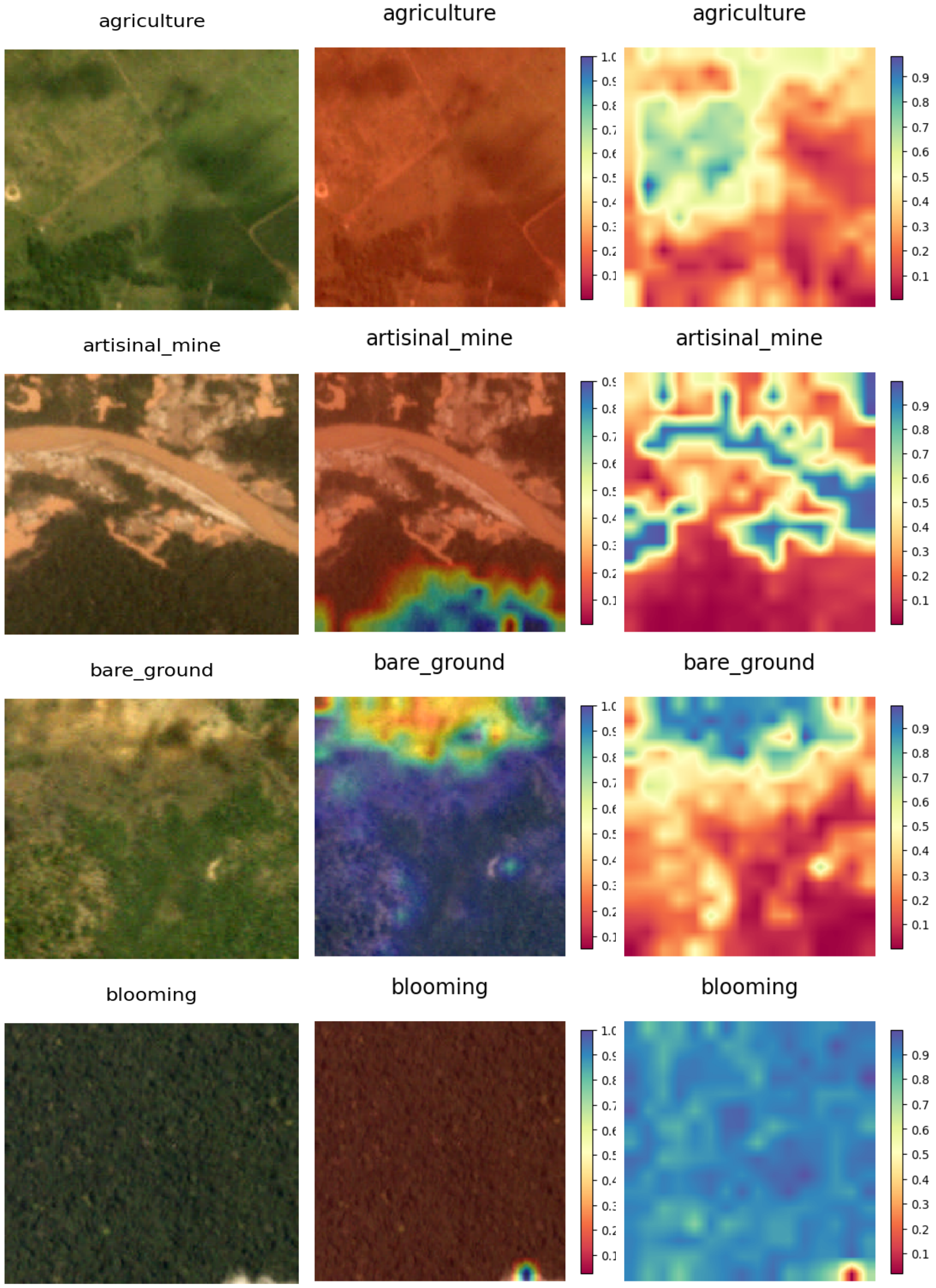
Starting with the weather labels, with the exception of partly cloudy, Grad-Cam produced uniform value maps. However, the information on the last layer was uniform red or blue for clear and haze which is what we would expect. Additionally, cloudy was detected in great detail, with the lower part of the image which is somewhat hazy marked with high values and the visible clouds marked in deep blue (highest values). The image for partly cloudy was also interesting, with Grad-Cam obtaining a more representative output than the last layer's slice. In overall, this led to high performance for the weather labels which is the norm across most of our experiments.

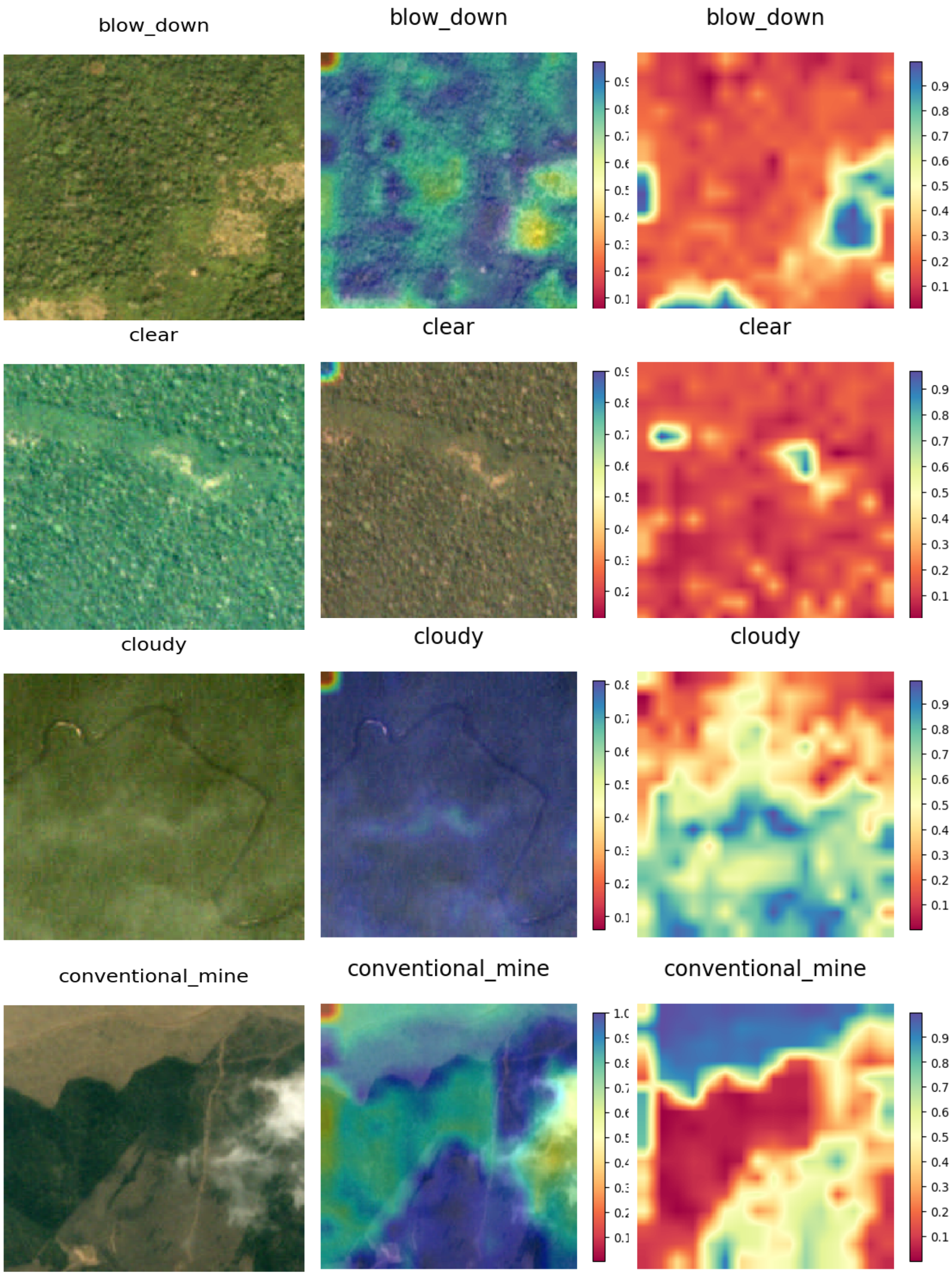
Moving on to the landscapes, Grad-Cam's output for water and cultivation was not useful. However, the last layer managed to have a very good representation of the image, with the areas of interest being marked in blue with astounding precision. This trend continues for bare ground and selective logging, where apart from the slices, Grad-Cam also produced detailed maps that segment the areas of interest from other labels.

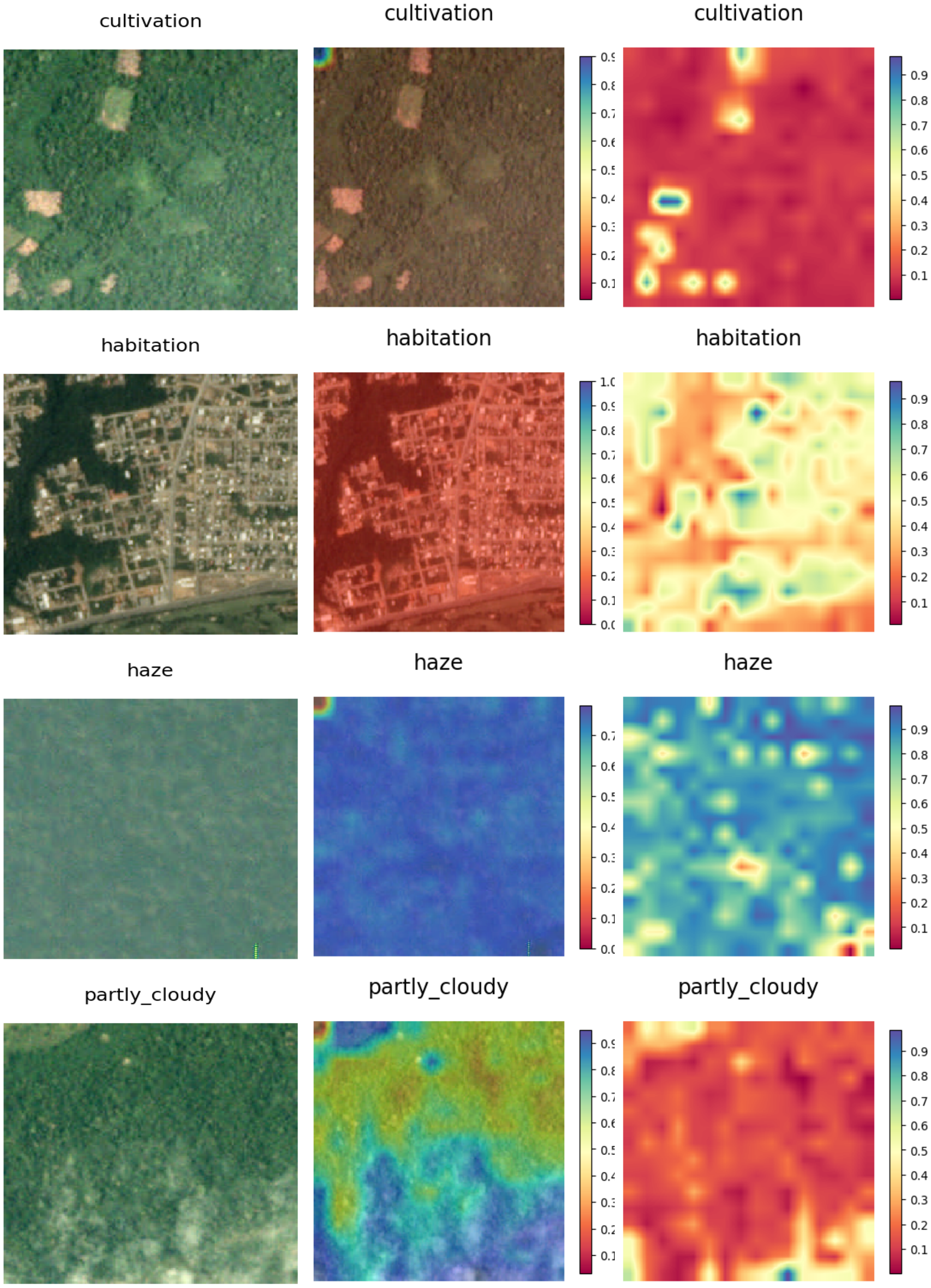
With regards to infrastructure, road and habitation which are dominant in their respective images seem to be well understood by the network by the time they reach the last layer, judging by their patterns. The same goes for agriculture, although the pattern looks less representative than the other two.

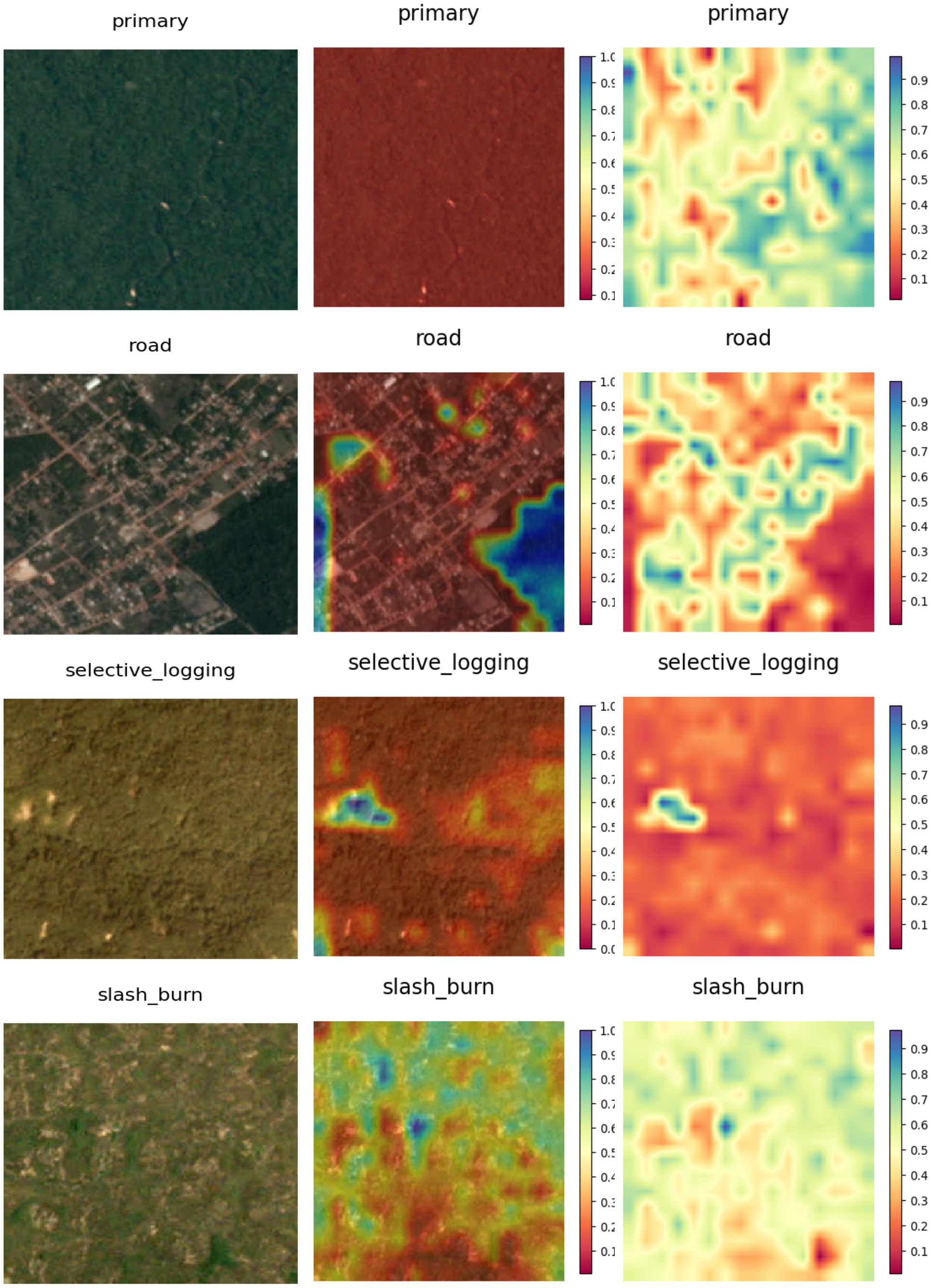
When it comes to the rare labels, the last layers' output seems to be representative of the underlying images with the same amount of precision as the common labels. Artisinal mine was correctly discerned from the road. Furthermore, the only instance of blooming was correctly detected and slash burn's pattern appears logical. Lastly, the conventional mine's image is not clear enough for us to properly analyze the pattern.

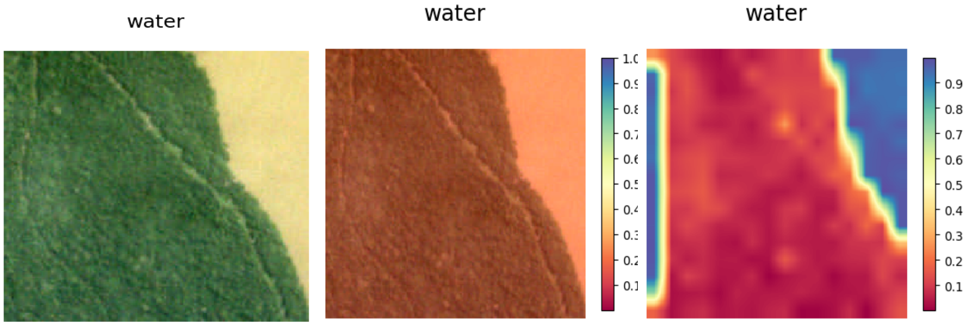
Despite the great precision of representation for each label's most representative image, Vision Transformer's score does not exceed that of EfficientNet's. A possible explanation is that it was easier for VIT to classify patterns found in the test set, which are highly similar to patterns found in the training set with strong confidence. However, more exotic patterns may cause more confusion for the network (form of overfitting), whereas EfficientNet tried a more general approach to the problem.


agriculture
agriculture
agriculture
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
artisinal_mine
artisinal_mine
artisinal_mine
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
bare_ground
bare_ground
bare_ground
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
blooming
blooming
blooming
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


blow_down
blow_down
blow_down
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
clear
clear
clear
cloudy
cloudy
cloudy
conventional_mine
conventional_mine
conventional_mine
1.0


cultivation
cultivation
cultivation
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
habitation
habitation
habitation
0.0
1.0
haze
haze
haze
partly_cloudy
partly_cloudy
partly_cloudy


primary
primary
primary
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
road
road
road
selective_logging
selective_logging
selective_logging
0.0
slash_burn
slash_burn
slash_burn


water
water
water
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

4.2.5 ResNet50, DensetNet and MobileNet

Experiment preparations

For the final part of the experiment, we tried some out of the box models to measure how they fare in comparison to the more fine-tuned and manually adjusted models showcased above. This section briefly describes the experiments and the results in a more general manner that encompasses all those models simultaneously. The is justified as throughout the section the methodology and results are quite similar across the models. The order of the resulting tables and images will always be ResNet50, then DenseNet and lastly MobileNet. Table 4.4 showcases the hyperparameters for the models (they are the same for each one).

Resolution	256x256
Batch size	128

Table 4.4: Hyperparameter setup for the multi-label experiment using the ResNet50, DensetNet and MobileNet.

Training process

In figures 4.27, 4.28 and 4.29 we can observe the training process for the classification. Due to our limited resources early stopping was implemented with patience 4. All models seem to generalize well, with the validation loss being better than the training loss. This trend seems to be present throughout all the models as the validation set is smaller than the training set and hence there might be disproportionately less edge cases. Additionally, most models started with an unstable validation loss curve, but as plateaus are reached and the learning rate drops the curve becomes smoother and stabilizes. Of all the models, ResNet was the best performer with the largest gap between training and validation loss.

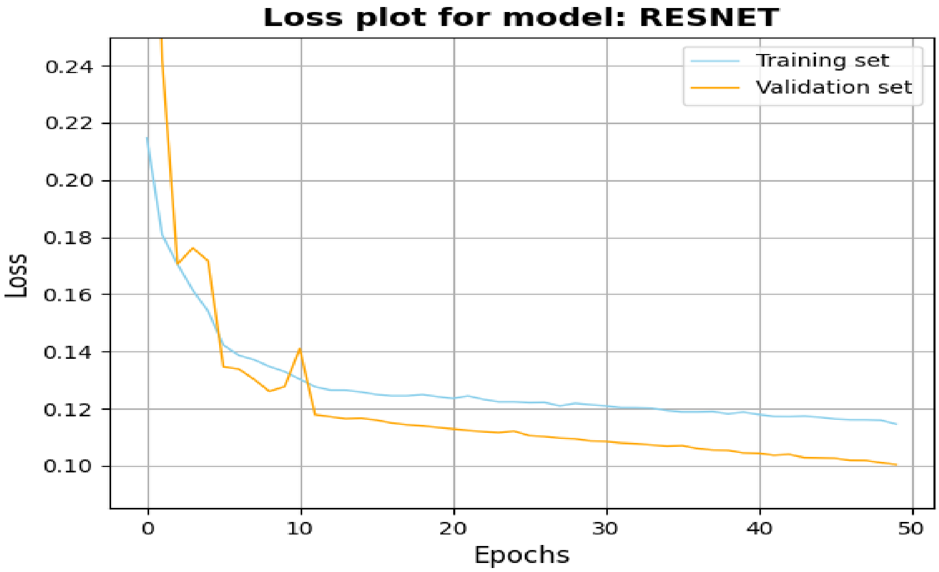


Figure 4.27: Training and validation loss during the ResNet's training phase

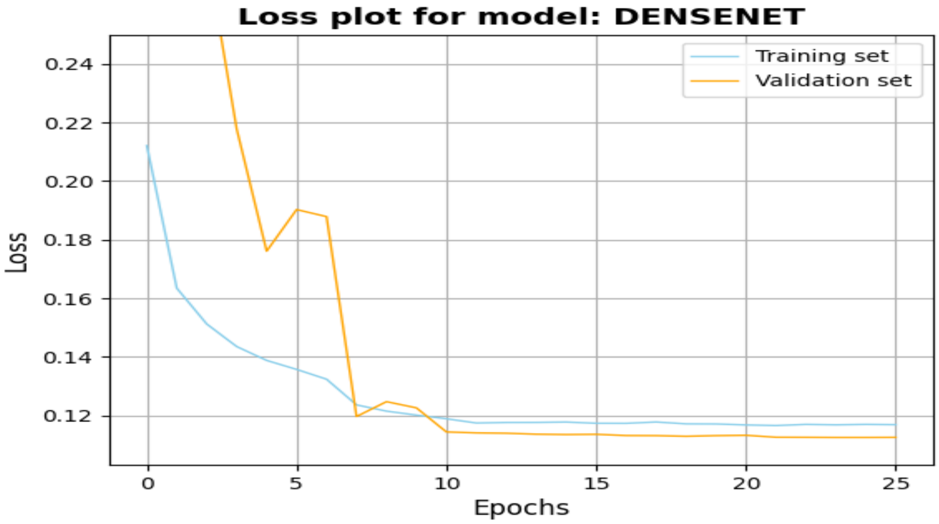


Figure 4.28: Training and validation loss during the DenseNet's training phase

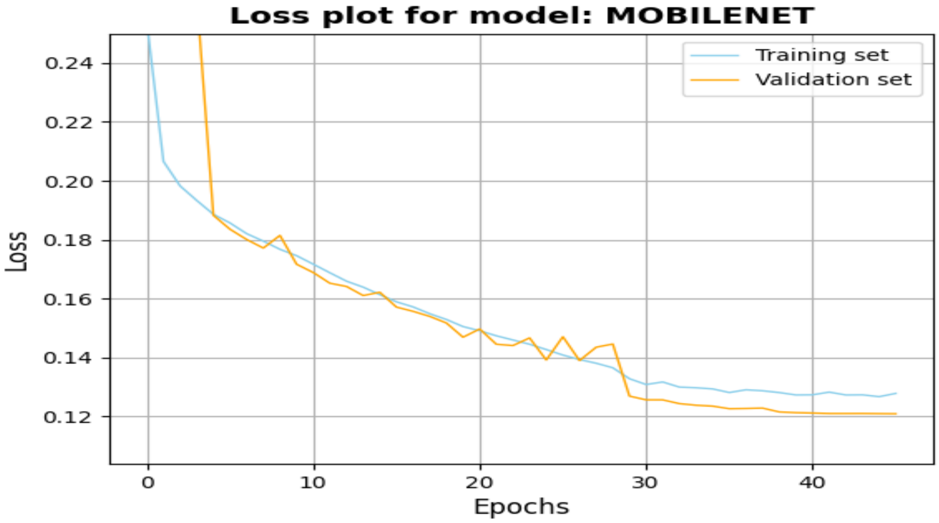


Figure 4.29: Training and validation loss during the MobileNet's training phase

Experiment results and explanation

The confusion matrices for the three models Can be found in Appendix A and the output from the Explainable AI techniques in Appendix B in order to make this section more concise. In overall, all those models follow the same patterns. Their confusion matrices show that they have more success in common labels, sacrificing scarce label performance in the process.

Grad-Cam's output often results to mappings, where the gradients do not exist, or they highlight a general area around the label. This can also be reflected in the very general patterns that are the last layer's output. This behaviour possibly explains the reason that the models were very easy to generalize as showcased by their loss plots, but also found it difficult to detect scarce or local labels.

As we show throughout the experiments, the heaviest models choose to create maps in the last layer that segment the original inputs with the greatest detail possible, leading to some success in the rare labels. The fact that those models chose to adopt generalized patterns for those labels, can possibly mean that in those types of experiments, if the model is not large enough to approximate the correct pattern in detail, it will likely choose to fit in the most common labels that affect the loss the most.

4.3 Results

Cross-examination and comparison

Due to the variety and depth of the experiments, it might be beneficial to perform a cross examination into how those different kinds of models approached the problem and comparison between them in order to find the best approach.

ResNet, DenseNet and mobileNet were the least fine-tuned models and gave us results that can be directly compared to VGG16 (the baseline). All those models had in common that they were architectures with complexities of the same scale. Throughout their training phase, they began with validation loses that were highly unstable that became smoother as the learning rate dropped in each plateau. All of them seem to have found a way to perform better in the validation set than they did in the training set. This might suggest that the training set contained in average more difficult cases than the validation set did. Another common feature among those models is that the decided to overlook the rare and local labels performing poorly on them in the process. This behaviour can also be evidenced by the general patterns found in their last layers which are more suited for detecting wide and common labels. VGG16 was subject to more tuning, and this might have lead to its Explainable AI gradients and patterns being a little bit more specific.

Larger and very fine-tuned models like EfficientNet and Vision Transformer, performed better in comparison to the aforementioned models. Starting with their loss plots, we can observe a closer relationship between the training and validation curves, leading to a better loss overall. What sets them apart besides their slightly better performance in the common labels, is their relative success to the rare labels. The Explainable AI techniques allow us to understand that this phenomenon can be attributed to the networks' attention to detail which all other networks lacked. It is possible that the number of parameters and the state of the art techniques that those models possess, allowed them to allocate some activation patterns to even the most local of labels, increasing their performance. The fact that the EfficientNet with slightly more generalized gradients, performed slightly better than the Vision Transformer whose gradients were more detailed, could indicate that there is a trade-off between a model's ability to produce detailed or generalized activation patterns with regards to the performance.

The comparison standard

Before showing the results between the trained models' predictions on the test set. It is worthwhile to briefly talk about which metric is the most reasonable when comparing models in the task of multilabel classification. As it is well known, accuracy, albeit a popular metric, does not take into account class imbalances. That said, our dataset is imbalanced. It is also very punishing towards the model's predictions, as the true label array and the predicted label array should exactly match in order for accuracy to increase. In our case, we would prefer a metric which can grade how good an approximate predicted description of the image is, relative to the true one. So, for instance, if out of the 17 labels, the 15 are predicted correctly, we should expect that the model produced a good enough description for this specific image and give it a positive score.

For this reason we will use the f-beta score in order to evaluate the model's performance on the test set. F-beta score computes the weighted harmonic mean between the precision and the recall. Beta is set to 0.5 by default which means that precision and recall and weighted equally. For each image, f-beta score will provide a value between 0 and 1 that determines how many predicted labels match the true labels. This is less punishing, rewarding better predicted descriptions for the images while also taking into account the imbalances between the appearances of the labels in the dataset.

Results table

Table 4.5 showcases the final results for each model. Both EfficientNet and VIT outperformed VGG16 which was the baseline. EfficientNet performed slightly better than VIT as it required less tuning and thus, we could focus on other aspects of the training phase. VIT on the other hand was very computationally expensive and on top of that it required a lot of fine tuning, as seemingly small tweaks in the hyperparameters could derail its performance. ResNet, DenseNet and MobileNet were the least tuned models but performed similarly to the baseline. More precisely, ResNet surpassed it, DenseNet and the most lightweight of them, MobileNet underperformed the baseline.

Model	**F-beta**
VGG16	90.39
ResNet	91.20
DenseNet	89.78
MobileNet	88.95
EfficientNet	92.75
Vision Transformer	92.22

Table 4.5: F-beta score comparison for the models' performance on the test set

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis we proposed a system that employs state of the art techniques in order to achieve the best performance possible in a multi-label classification task. The labels were probable causes of deforestation for the Amazon rain-forest. In order to reach our end goal which is the best possible performance for the given dataset, we compare 2 vastly different architectures with a baseline model which is the VGG16. VGG16 proved to be a good fit for the baseline model as it provided high performance, while also being relatively inexpensive with regards to the resources.

The first model, EfficientNet, proved to be high performing even in its less complex architectures. It is also important to note that its tuning phase was very simple due to its philosophy of "quantizing" its architecture into discrete levels. This feature allowed us to produce the best results out of all our other models, in a time frame that was non-prohibitive for the limited resources available.

The second model, Vision Transformer, provided a very interesting alternative to the already established convolutional network approaches. This experimental architecture imported many ideas from the field of natural language processing successfully. Its final performance was a bit worse than that of the EfficientNet. However, it has got a lot of room for improvement, as it offers a vast array of hyperparameters that can be tweaked but were not, due to its sheer computational cost.

For the purposes of tackling the problem with a variety of networks, we trained ResNet, DenseNet and MobileNet on the dataset for the last experiments. Comparing those established architectures to the state of the art, gave us a better understanding of the differences in perception between them. However, their scores were more comparable to that of the baseline.

In the context of explainable AI, we proceeded to use the Grad-Cam and the last convolutional layer's tensor, in order to explain why the architectures performed well or struggled in certain labels. By peeking inside the network we found trends in the models' depiction of the data which acted as the differentiating factor between the good models and the best models.

Ultimately, our best performing model achieved an f-beta score of 92.75, surpassing the score of 0.89 published by Aaron Loh and Kenneth Soo [12], which performed similar experiments on the same dataset and published them in their paper. However, it is important to note that their paper's results are derived from the respective kaggle submission, while in our thesis we split the training set that we downloaded from kaggle in 60/20/20 and test the final 20% slice for our final results.

5.2 Future work

5.2.1 Noisy Student

Noisy student [26] is a semi-supervised learning approach. It extends the concept of the student-teacher model. It starts by training a teacher model on the dataset. The teacher then has the responsibility on creating new images based on the dataset that it has been exposed and/or mislabeling some of the images. Then, a student model is trained on the dataset produced by the teacher. The student gets evaluated on the prediction performance in the original dataset while having been trained exclusively on the fabricated dataset. The idea behind this technique is that the student will eventually learn how to distill valid information from noisy data. It will only rely on a very small dataset that is given to the teacher, from which a very large dataset will be produced. A very popular core (architecture) for the noisy student is the already trained EfficientNet, which can perform both the roles of a student and a teacher. In practise, it can increase our performance scores in multilabel classification but it is prohibitively computationally expensive for the purposes of this thesis.

5.2.2 EfficientNet B7 and transfer learning

During the tuning phase of the EfficientNet model, we could not effectively test architectures that were more complex than the b4 level. By increasing the complexity, the batch size would decrease to something lower than 16, which significantly dropped the performance of the model. It would be interesting to observe whether the most complex architectures would outperform the simpler ones given enough GPU memory to maintain a batch size of 16.

Additionally, for each of those architectures, there exists a respective model pretrained in the imagenet dataset. By experimenting on this, it was found that simply performing predictions using those models did not result in good f-beta scores. Nevertheless, by using transfer learning in the most heavy architectures while also unfreezing a portion of those layers, we could actually boost the performance.

5.2.3 TResNet

As of the time this thesis is written, TResNet [17] is a brand new variation of the well known ResNet and its variations [24]. While the core architecture does not change, a lot of successful design tricks that were developed over the years are employed in order to greatly improve its performance. Namely those are, SpaceToDepth stem, Anti-Alias downsampling, In-Place Activated BatchNorm, Blocks selection and squeeze-and-excitation layers. Its performance looks especially promising for multilabel classification problems as it became state of the art for the MS-COCO dataset which is used for multilabel classification benchmarks.

5.2.4 Tuning

As stated before, the project was ran in limited resources and as a result certain compromises were made mostly in terms of performance. First of all, there was not a possibility for proper the tuning of the Vision Transformer model. While in general the experiments showed an increase in performance for small size architectures when compared to the medium sized ones, the heavier settings for the architectures were very difficult to test and could potentially lead to an increase in the overall performance.

Additionally, there is still an increase in performance to be found by tuning the image generator using techniques such as AutoAugment. Generally, due to the nature of the dataset, non-invasive augmentation were proffered such as flips. Even a small alteration to a group of pixels could lead to the complete loss of information that hints to specific label in an image. Consequently, one can doubt the effect that AutoAugment can have on the dataset but it might still be worth trying.

Bibliography

[1] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115.

[2] N. Bakalos et al. "Protecting Water Infrastructure From Cyber and Physical Threats: Using Multimodal Data Fusion and Adaptive Deep Learning to Monitor Critical Systems". In: *IEEE Signal Processing Magazine* 36.2 (2019), pp. 36–48. DOI: `10.1109/MSP.2018.2885359`.

[3] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *arXiv preprint arXiv:2005.12872* (2020).

[4] Rodrigo Caye Daudt et al. "Urban change detection for multispectral earth observation using convolutional neural networks". In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 2115–2118.

[5] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[6] A. Doulamis et al. "Combined Convolutional Neural Networks and Fuzzy Spectral Clustering for Real Time Crack Detection in Tunnels". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 4153–4157. DOI: `10.1109/ICIP.2018.8451758`.

[7] Dillon Hicks et al. "Mangrove Ecosystem Detection using Mixed-Resolution Imagery with a Hybrid-Convolutional Neural Network". In: ().

[8] Jeremy Irvin et al. "ForestNet: Classifying Drivers of Deforestation in Indonesia using Deep Learning on Satellite Imagery". In: *arXiv preprint arXiv:2011.05479* (2020).

[9] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[10] Xiaomeng Li et al. "H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes". In: *IEEE transactions on medical imaging* 37.12 (2018), pp. 2663–2674.

[11] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[12] Aaron Loh and Kenneth Soo. "Amazing Amazon: Detecting Deforestation in our Largest Rainforest". In: ().

[13] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[14] Thiago Nunes Kehl et al. "Amazon rainforest deforestation daily detection tool using artificial neural networks and satellite images". In: *Sustainability* 4.10 (2012), pp. 2566–2573.

[15] MX Ortega et al. "EVALUATION OF DEEP LEARNING TECHNIQUES FOR DEFORESTATION DETECTION IN THE AMAZON FOREST." In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2019).

[16] Pranoy Panda, Martin Barczyk, and Jen Beverly. "Estimating Forest Ground Vegetation Cover From Nadir Photographs Using Deep Convolutional Neural Networks". In: ().

[17] Tal Ridnik et al. "TResNet: High Performance GPU-Dedicated Architecture". In: *arXiv preprint arXiv:2003.13630* (2020).

[18] Rafael AS Rosa et al. "Deforestation detection in Amazon rainforest with multitemporal X-band and p-band sar images using cross-coherences and superpixels". In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2017, pp. 3015–3018.

[19] Pedro Savarese. "On the Convergence of AdaBound and its Connection to SGD". In: *arXiv preprint arXiv:1908.04457* (2019).

[20] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[21] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013).

[22] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[23] Yongjia Song and Yuhang Wang. "Global Wildfire Outlook Forecast with Neural Networks". In: *Remote Sensing* 12.14 (2020), p. 2246.

[24] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *arXiv preprint arXiv:1602.07261* (2016).

[25] Mingxing Tan and Quoc V Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *arXiv preprint arXiv:1905.11946* (2019).

[26] Qizhe Xie et al. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10687–10698.

[27] Salih Can Yurtkulu, Yusuf Hüseyin Şahin, and Gozde Unal. "Semantic Segmentation with Extended DeepLabv3 Architecture". In: *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2019, pp. 1–4.

[28] Yunhua Zhang et al. "Structured siamese network for real-time visual tracking". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 351–366.

Appendices

Appendix A

Confusion matrices for ResNet, DenseNet and MobileNet

Confusion matrices (absolute and percentage) for ResNet, DenseNet and MobileNet

A.0.1 ResNet

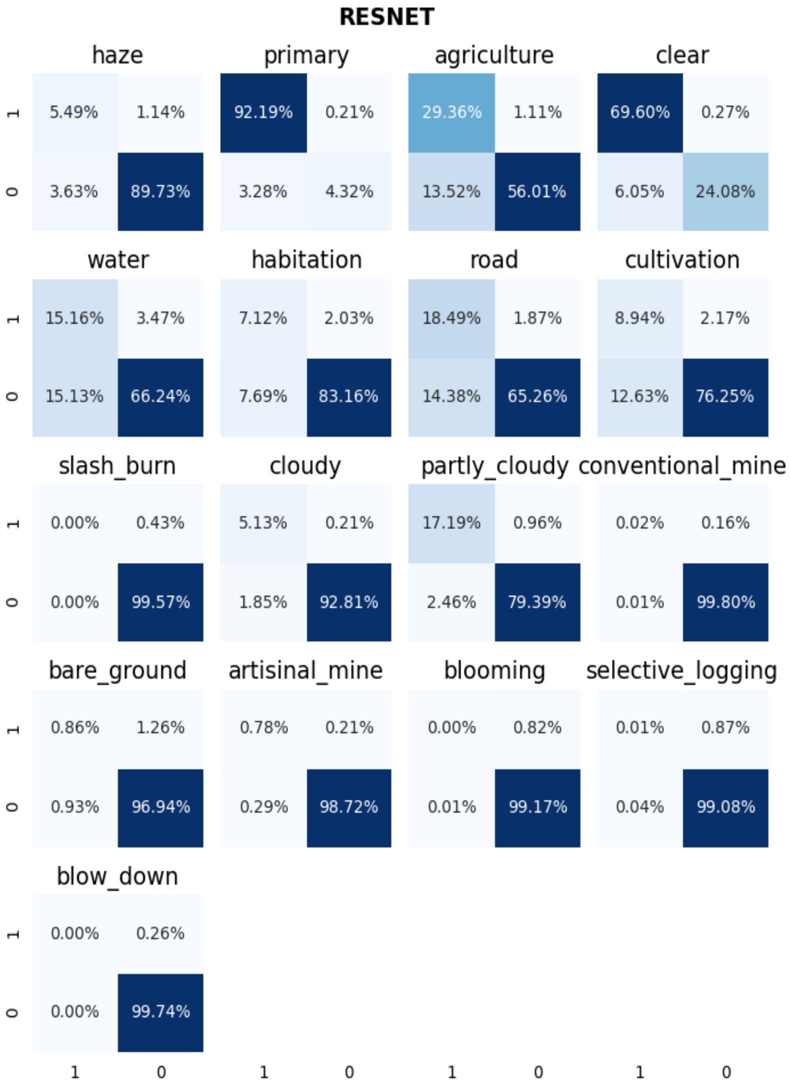


Figure A.1: Relative performance for each individual label

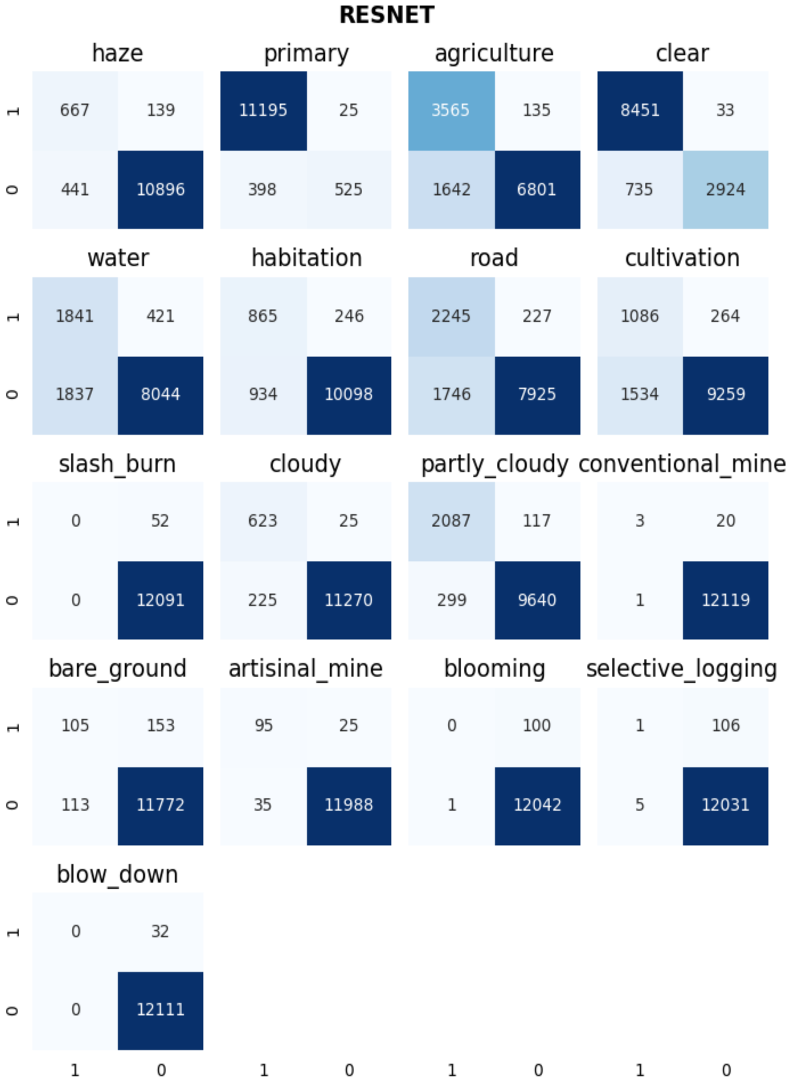


Figure A.2: Absolute performance for each individual label

A.0.2 DenseNet

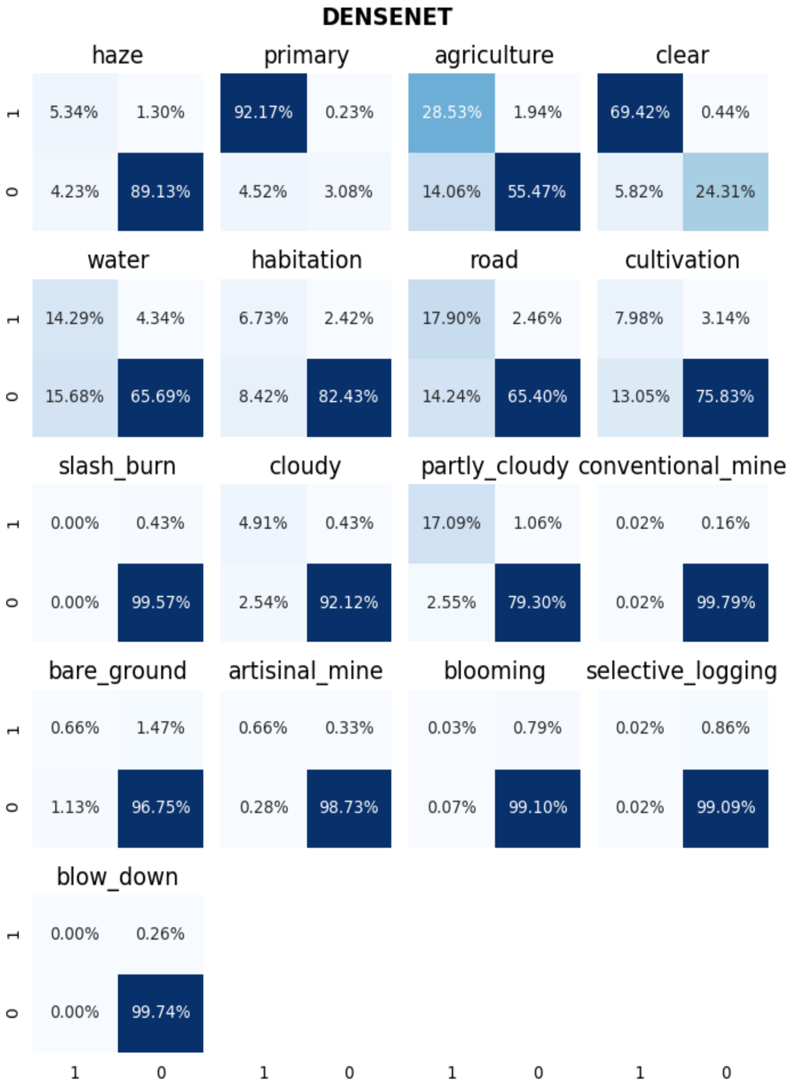


Figure A.3: Relative performance for each individual label

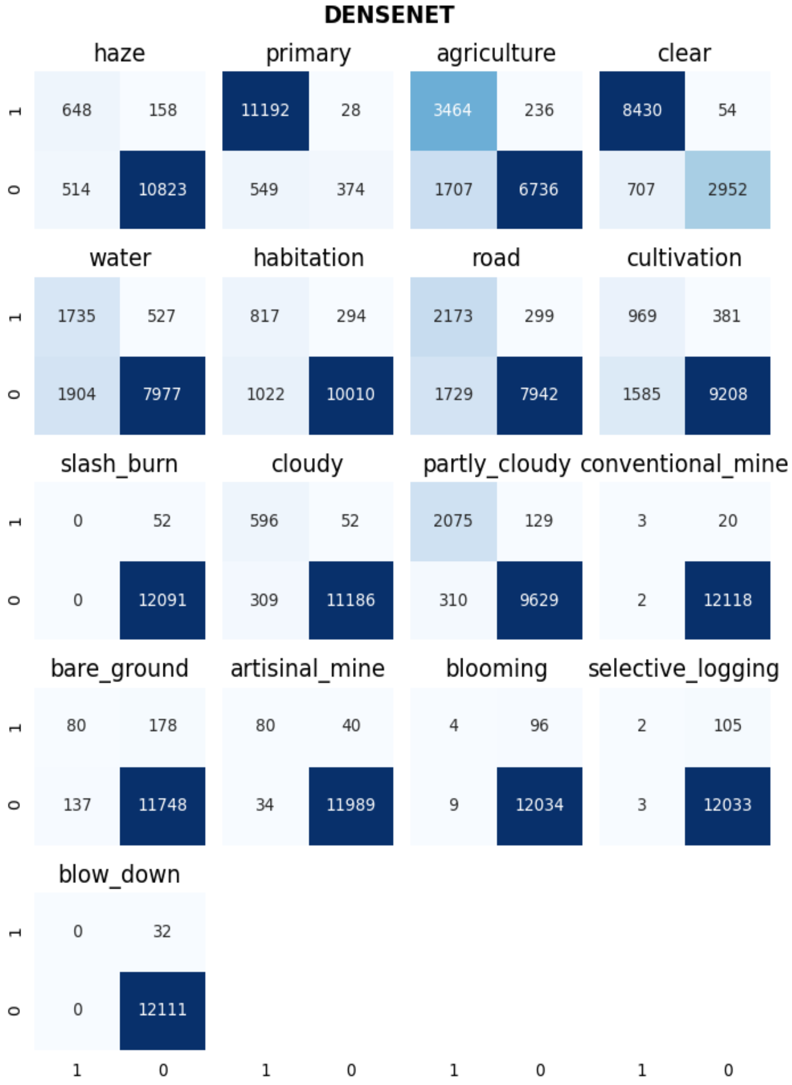


Figure A.4: Absolute performance for each individual label

A.0.3 MobileNet

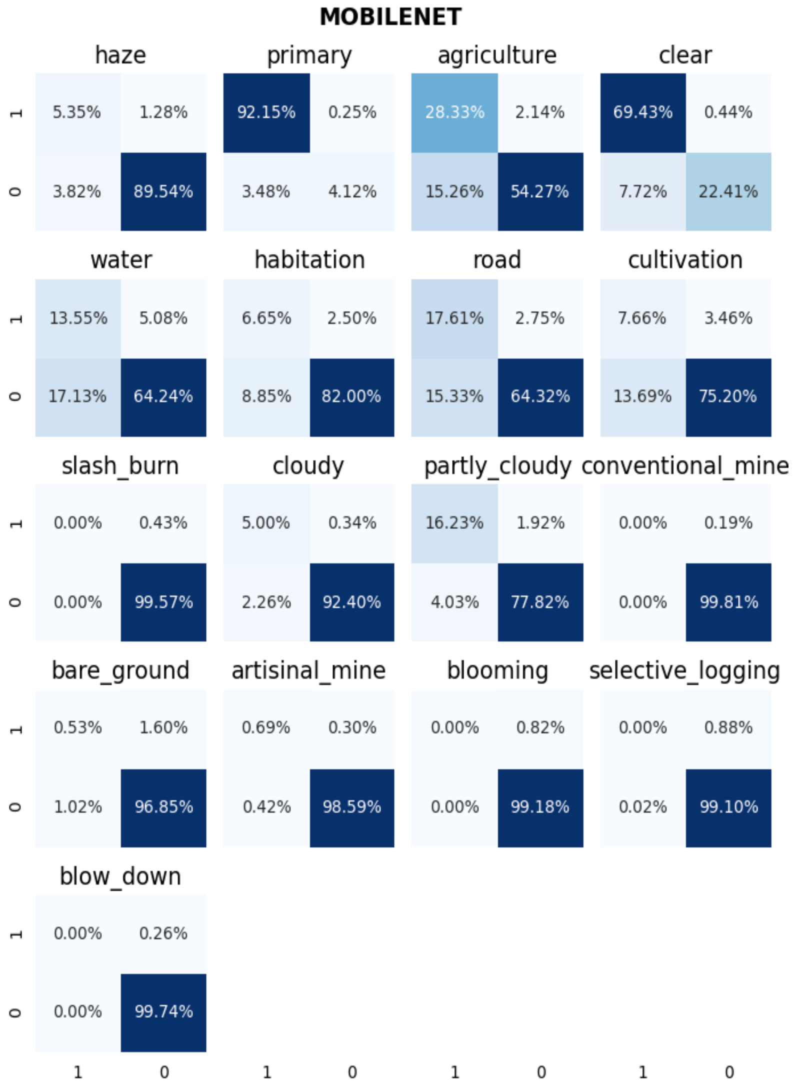


Figure A.5: Relative performance for each individual label

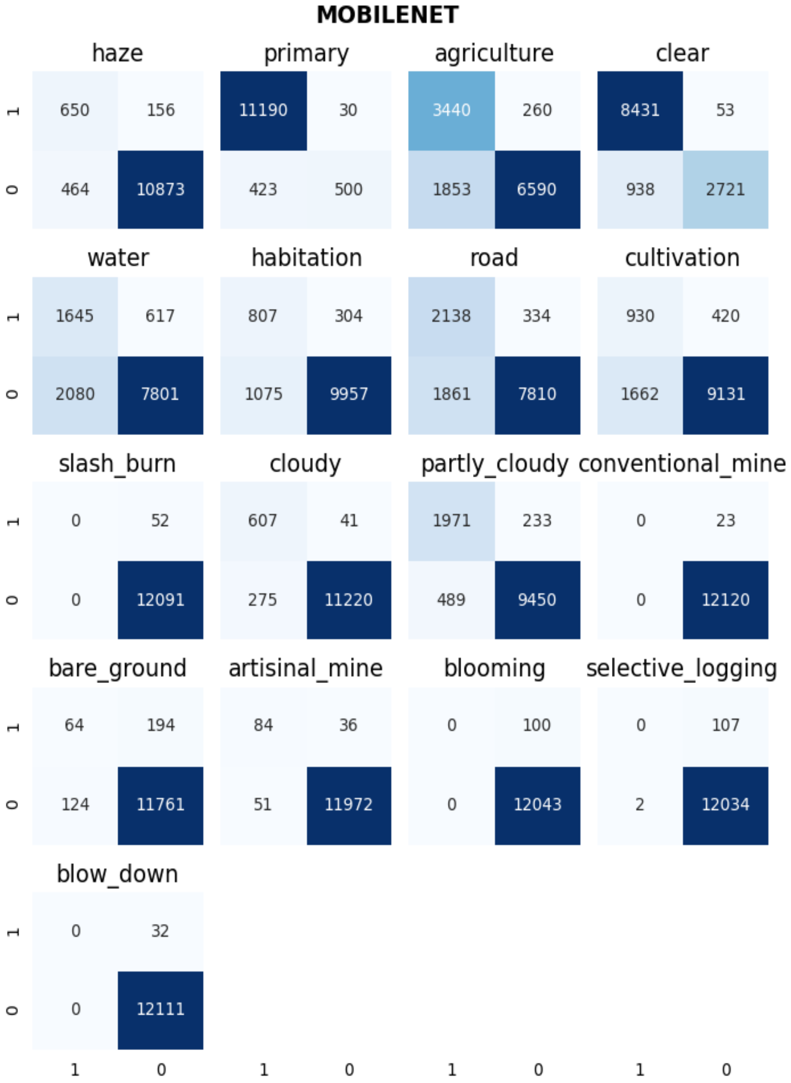


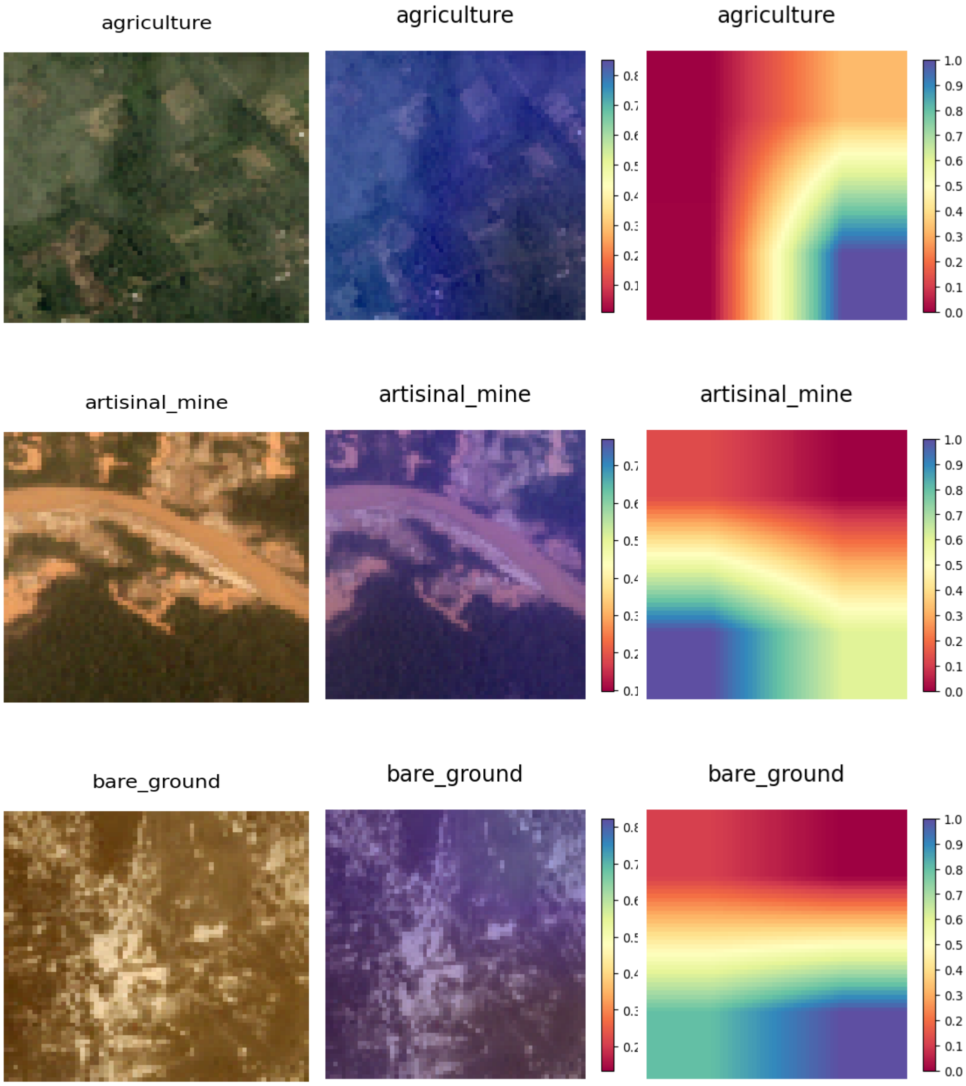
Figure A.6: Absolute performance for each individual label

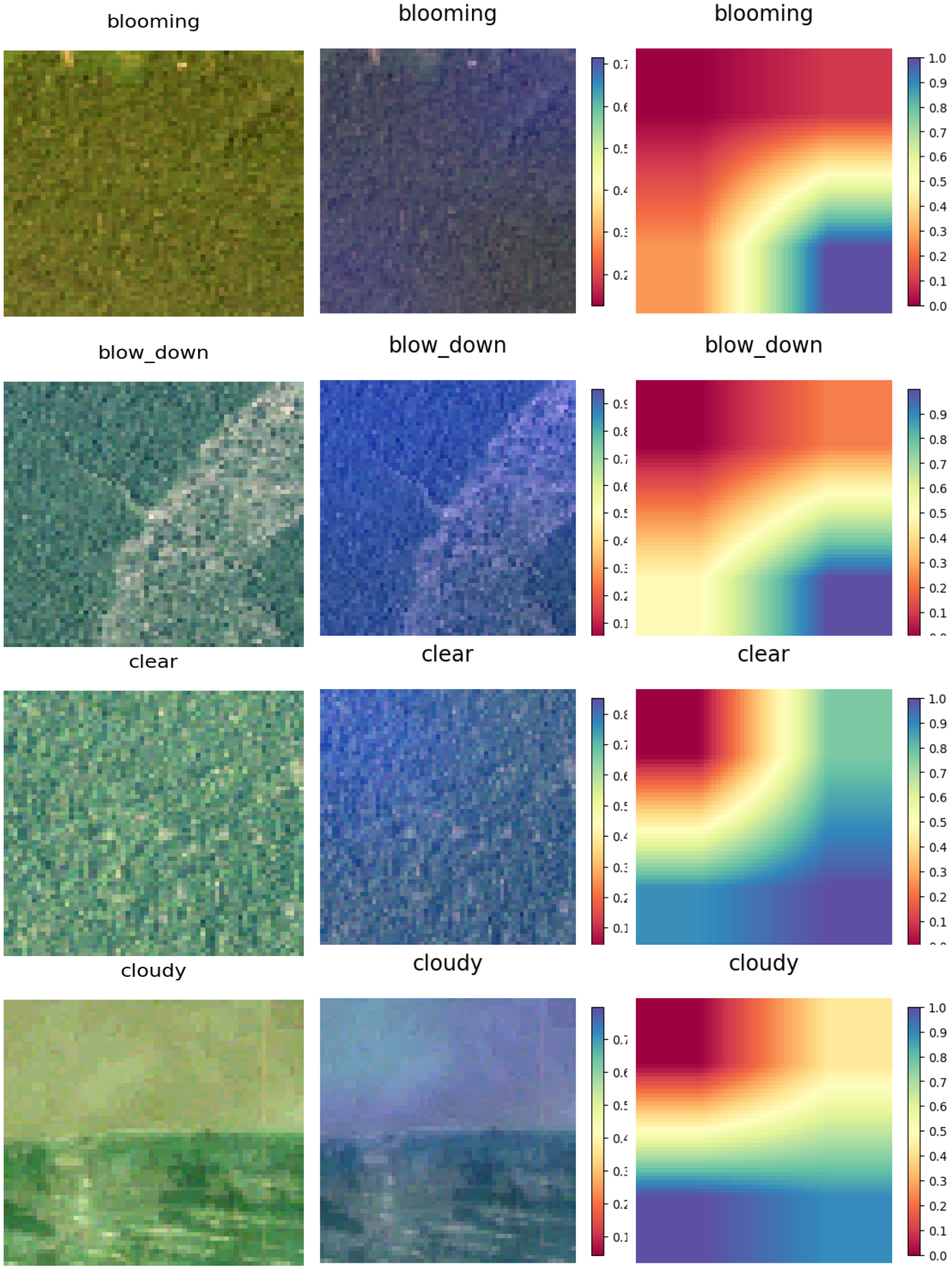
Appendix B

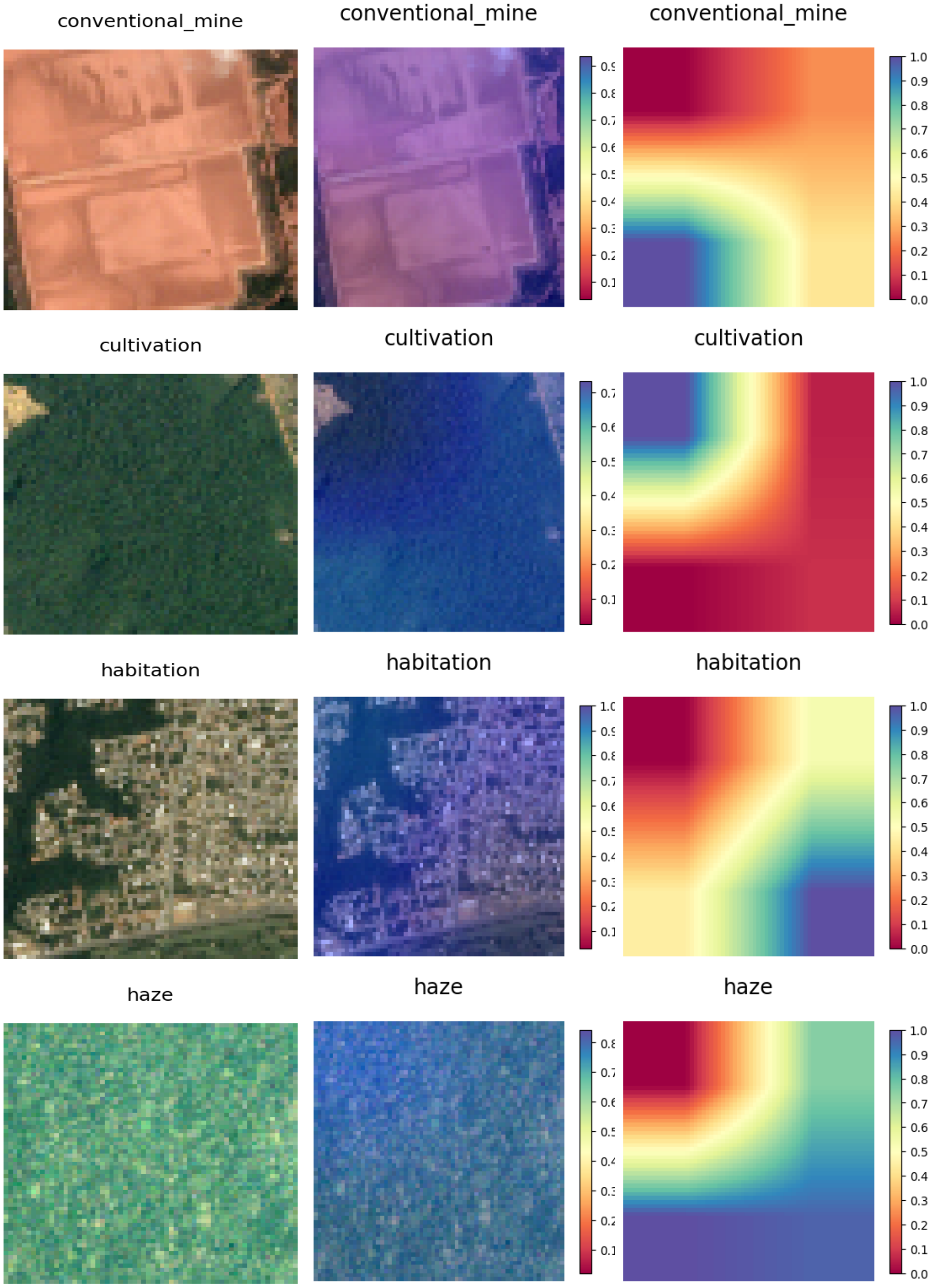
Explainable AI for ResNet, DenseNet and MobileNet

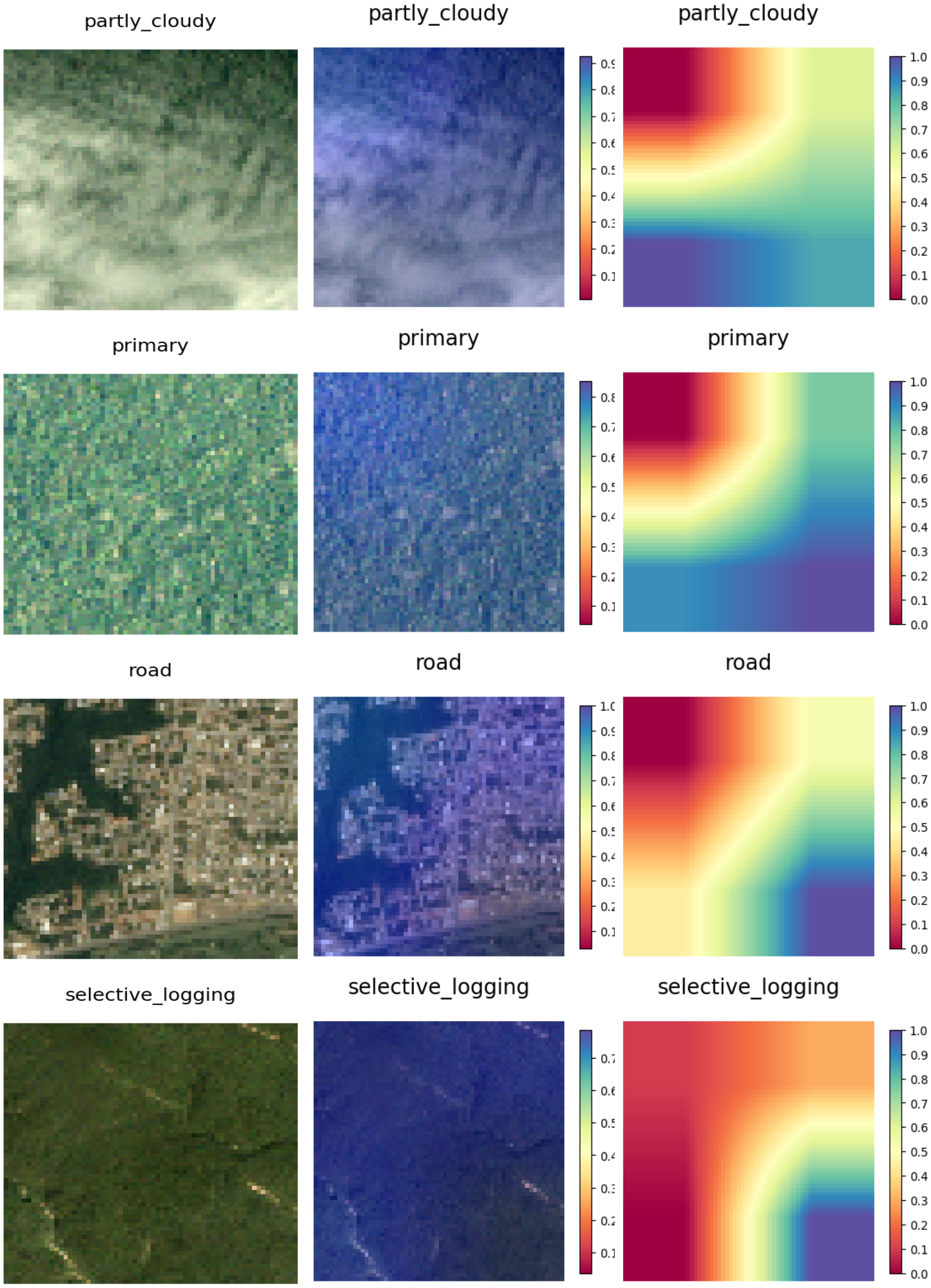
In the context of explainable AI, the most representative images (left) for each label for ResNet, DenseNet and MobileNet where processed using the Grad-Cam technique (middle). The last convolutional layer's output was also extracted in order for us to form a more comprehensive explanation.

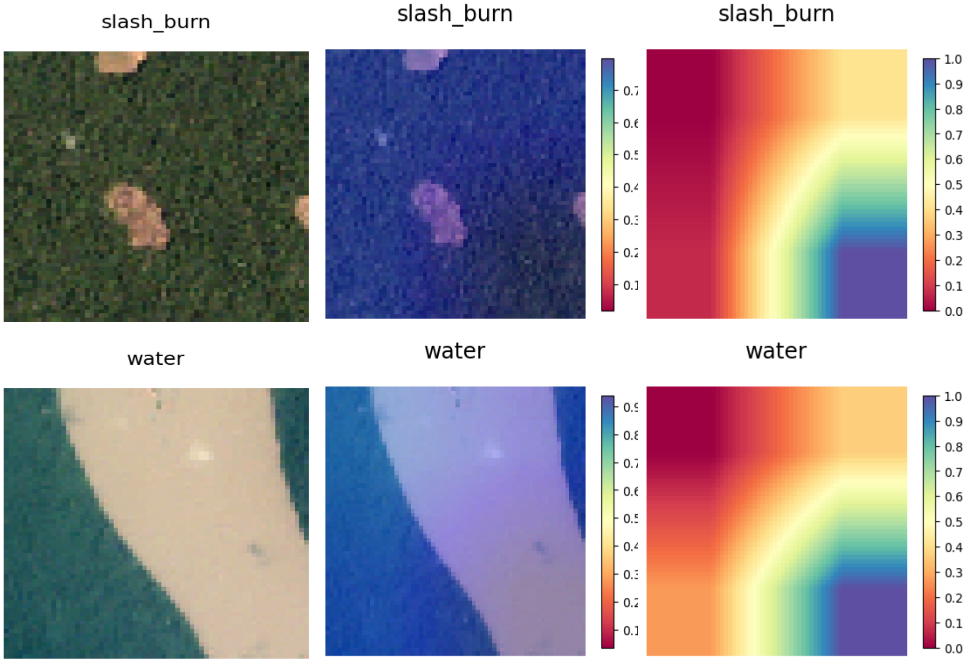
B.0.1 ResNet


agriculture
agriculture
agriculture
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
artisinal_mine
artisinal_mine
artisinal_mine
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
bare_ground
bare_ground
bare_ground
0.8
0.7
0.6
0.5
0.4
0.3
0.2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

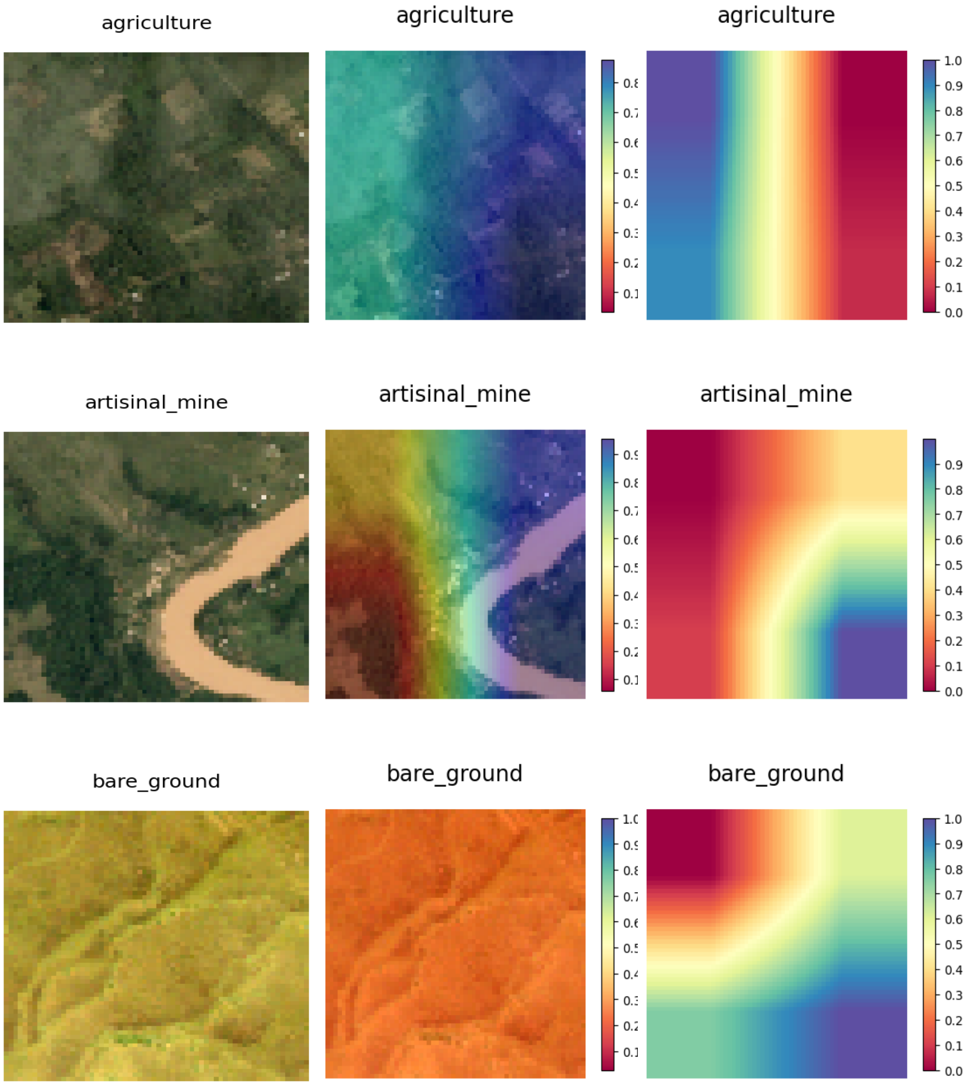

blooming
blooming
blooming
0.7
0.6
0.5
0.4
0.3
0.2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
blow_down
blow_down
blow_down
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
clear
clear
clear
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
cloudy
cloudy
cloudy
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

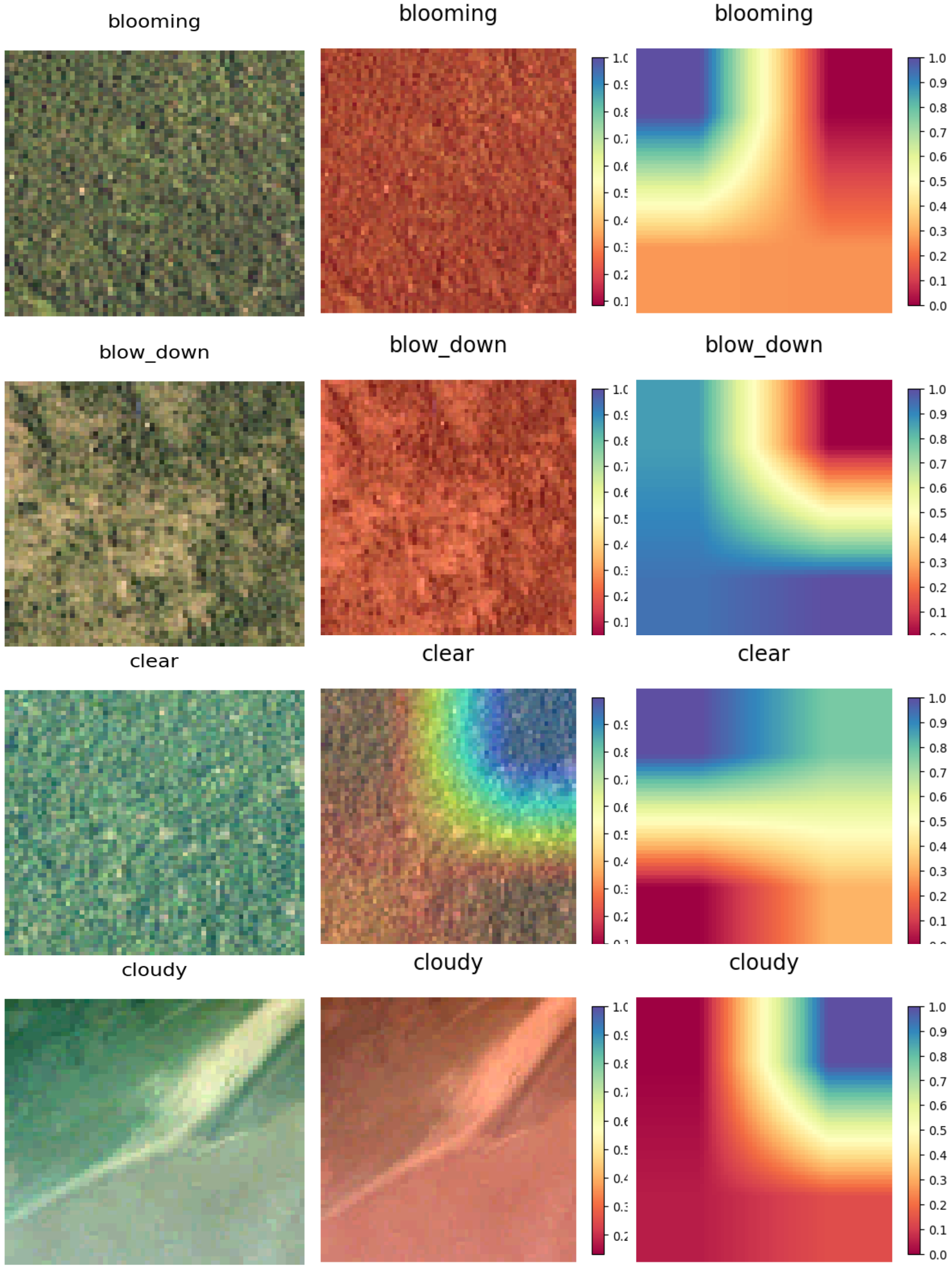

conventional_mine
conventional_mine
conventional_mine
cultivation
cultivation
cultivation
habitation
habitation
habitation
haze
haze
haze
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

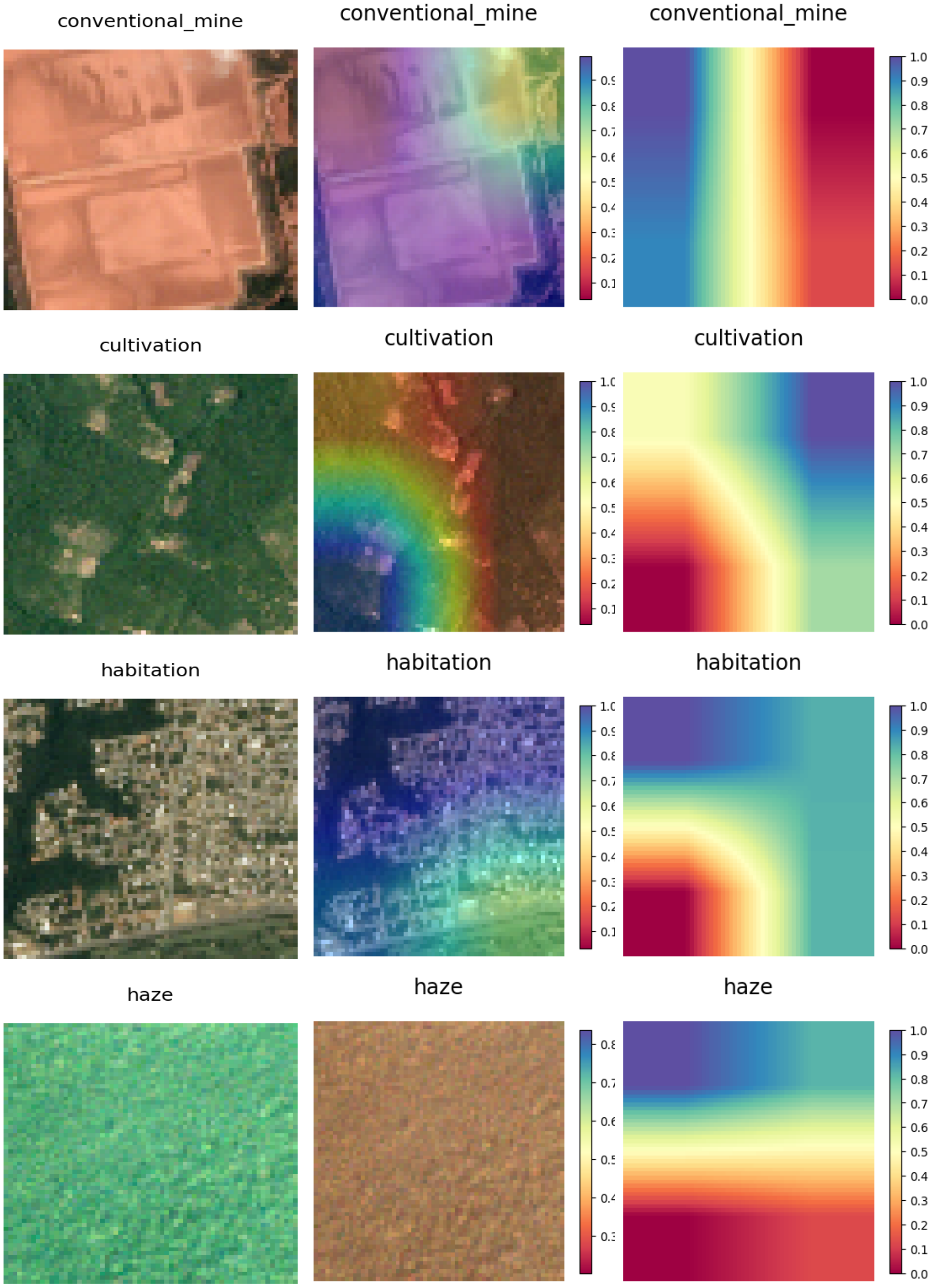

partly_cloudy
partly_cloudy
partly_cloudy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
primary
primary
primary
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
road
road
road
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
selective_logging
selective_logging
selective_logging
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

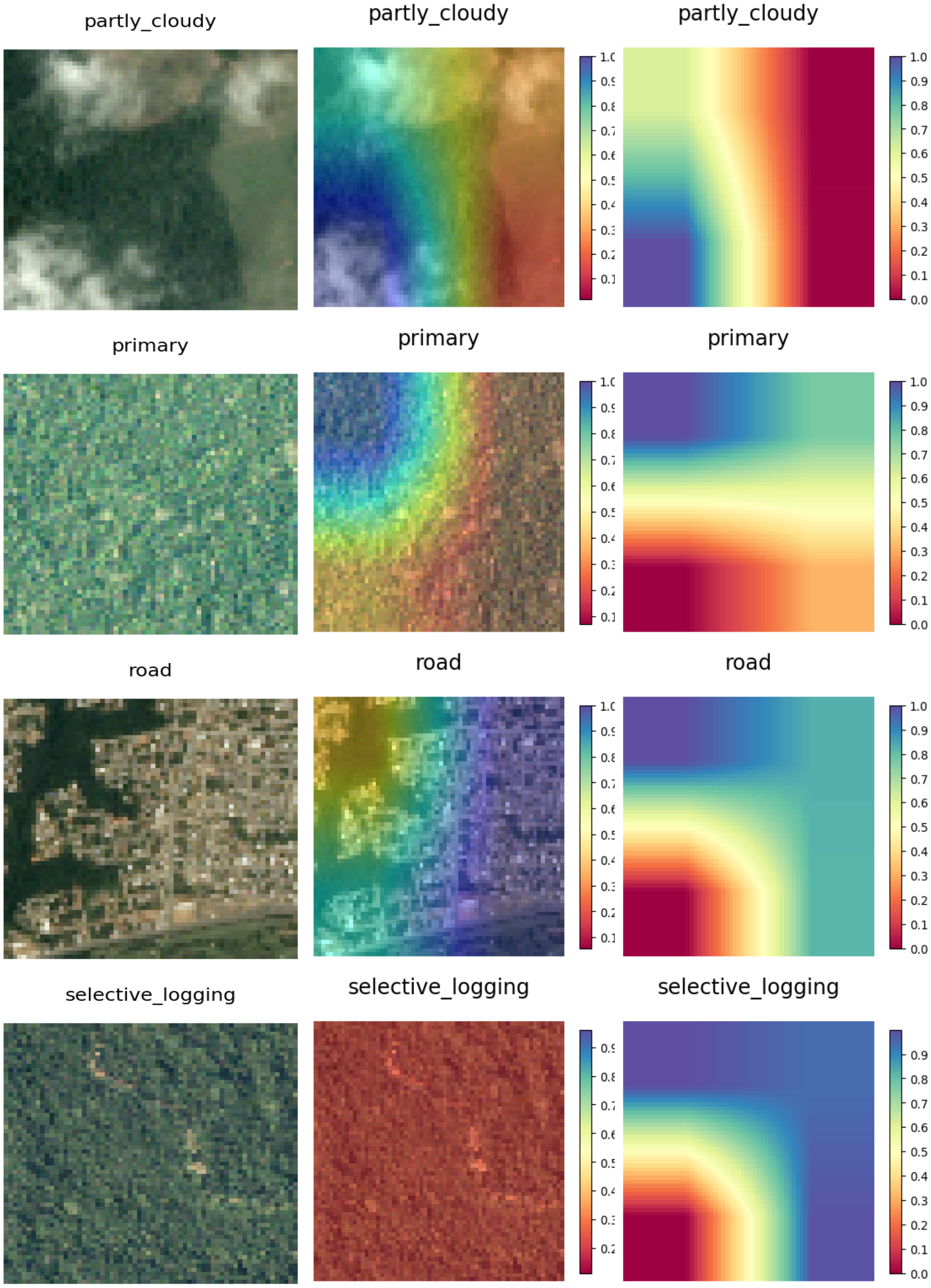

slash_burn
slash_burn
0.1
0.2
0.3
0.4
0.5
0.6
0.7
slash_burn
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
water
water
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
water
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

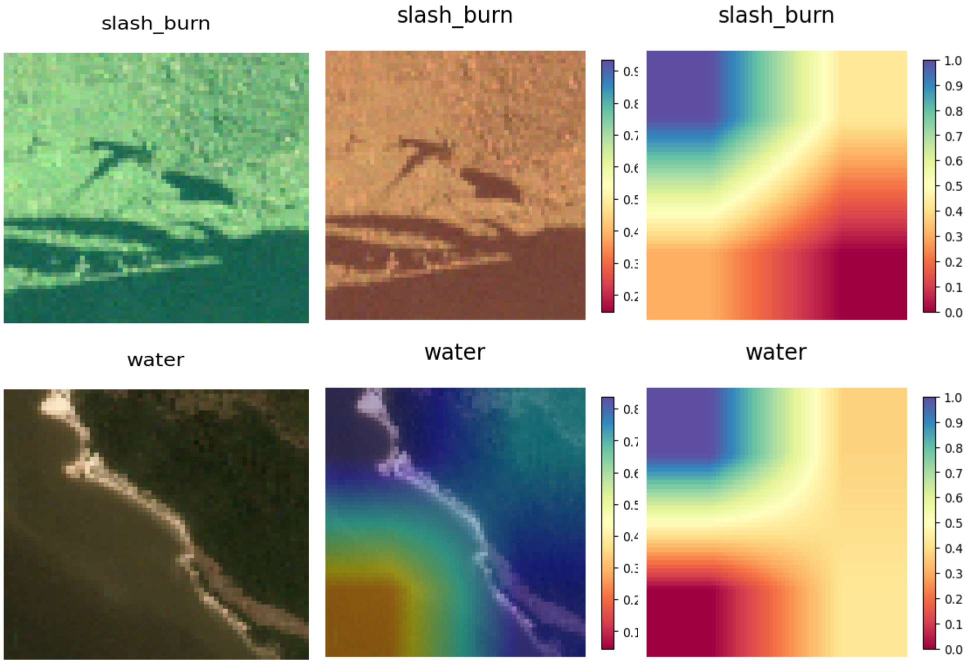
B.0.2 DenseNet



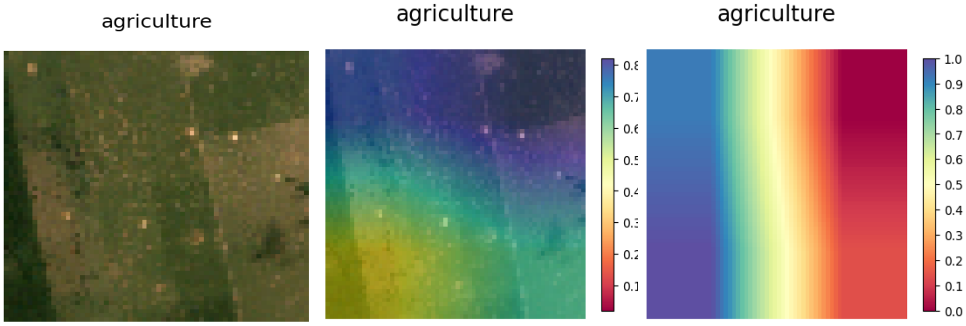

blooming
blooming
blooming
blow_down
blow_down
blow_down
clear
clear
clear
cloudy
cloudy
cloudy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

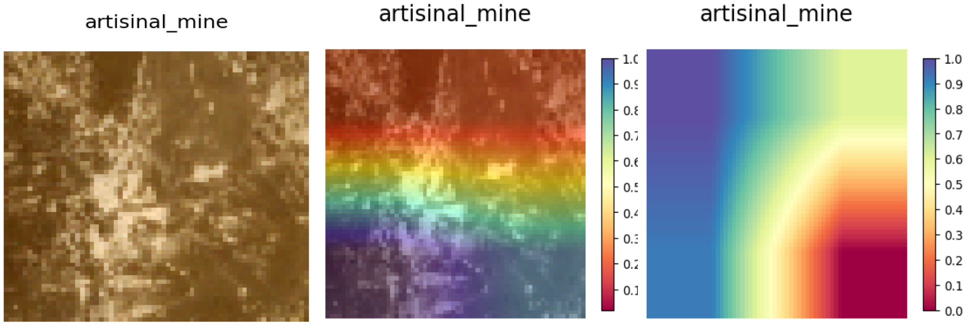

conventional_mine
conventional_mine
conventional_mine
cultivation
cultivation
cultivation
habitation
habitation
habitation
haze
haze
haze
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

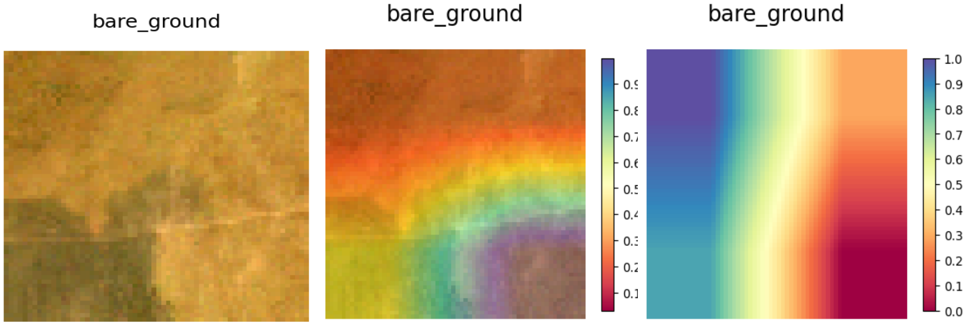

partly_cloudy
partly_cloudy
partly_cloudy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
primary
primary
primary
road
road
road
selective_logging
selective_logging
selective_logging

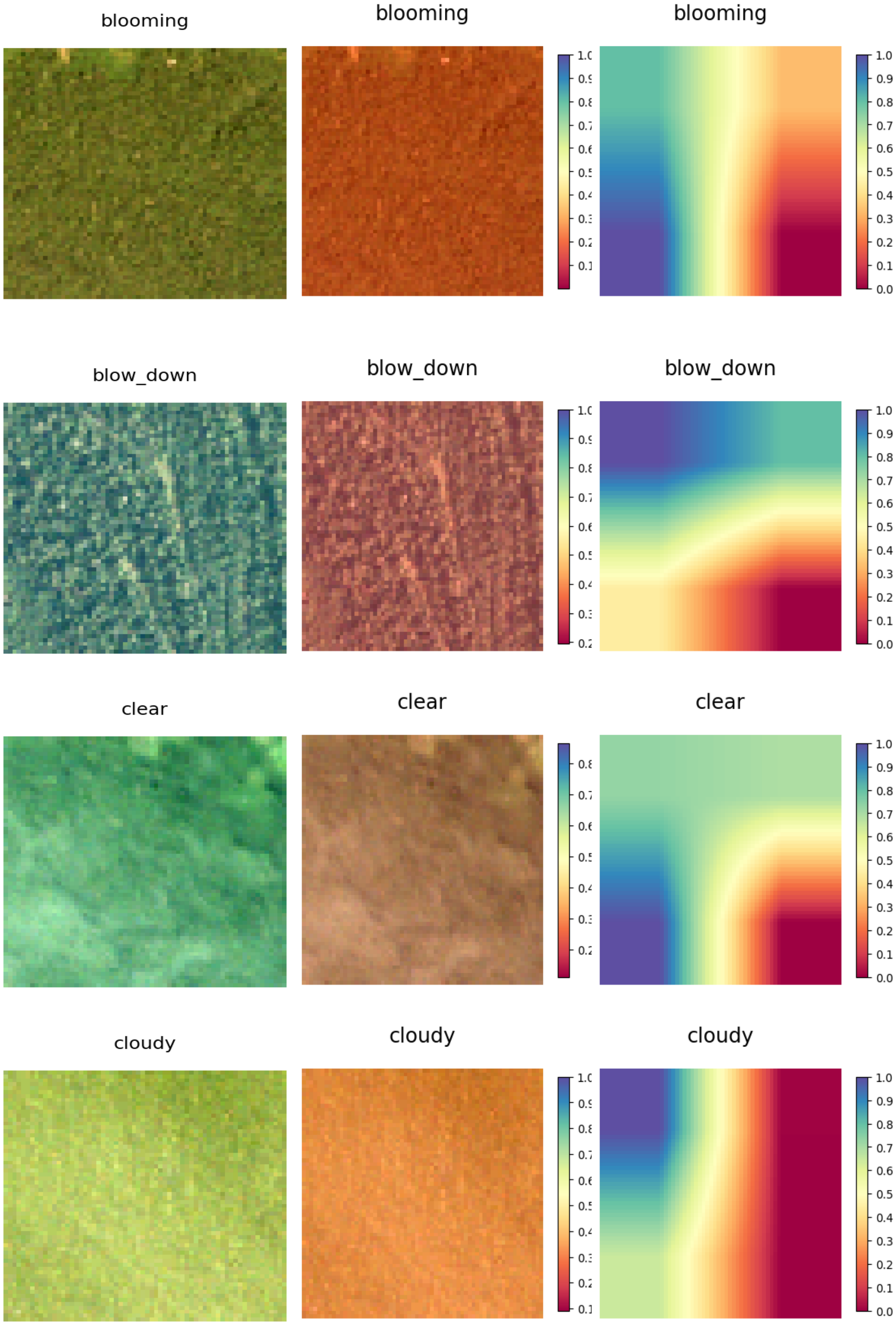

slash_burn
slash_burn
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
slash_burn
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
water
water
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
water
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

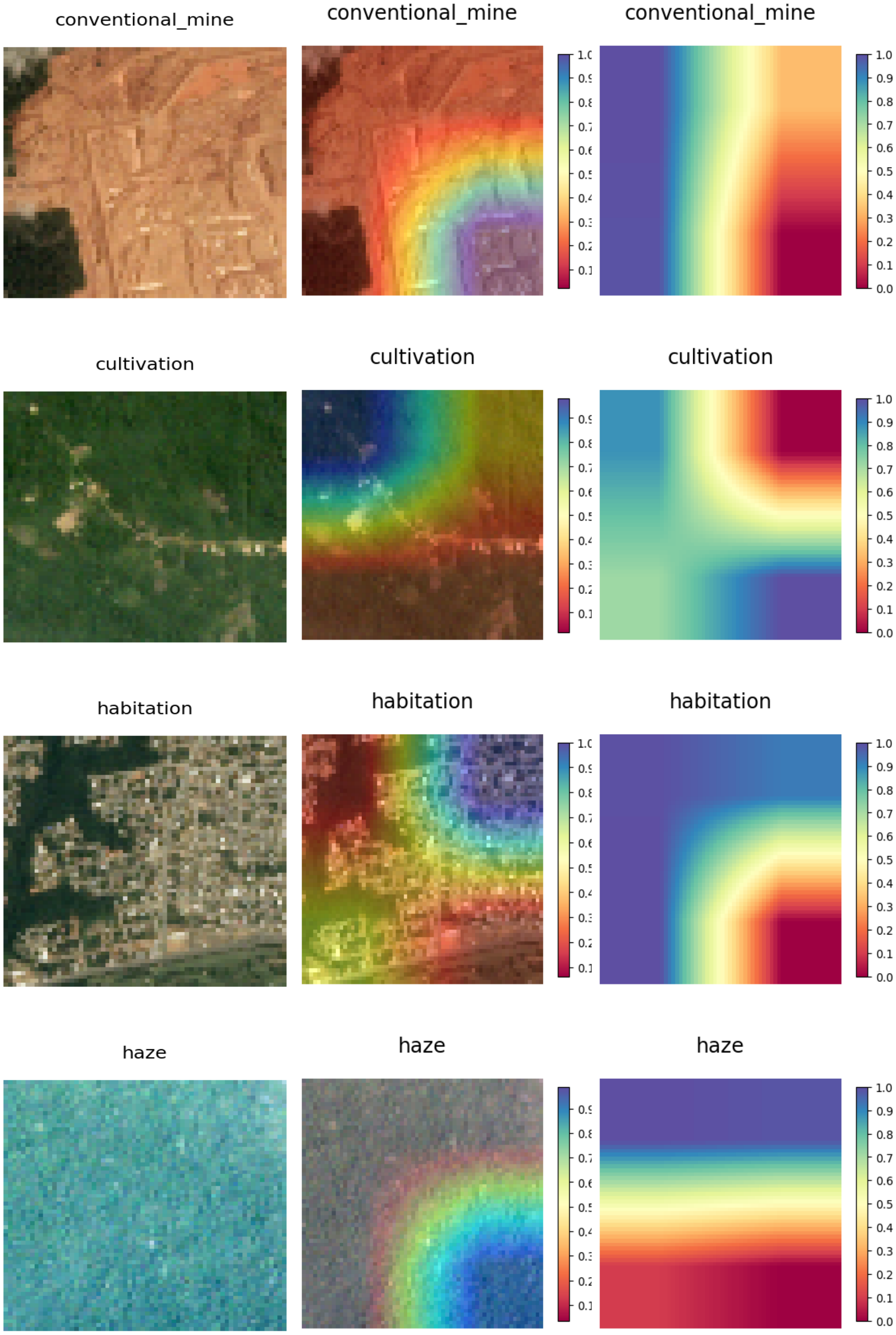
B.0.3 MobileNet

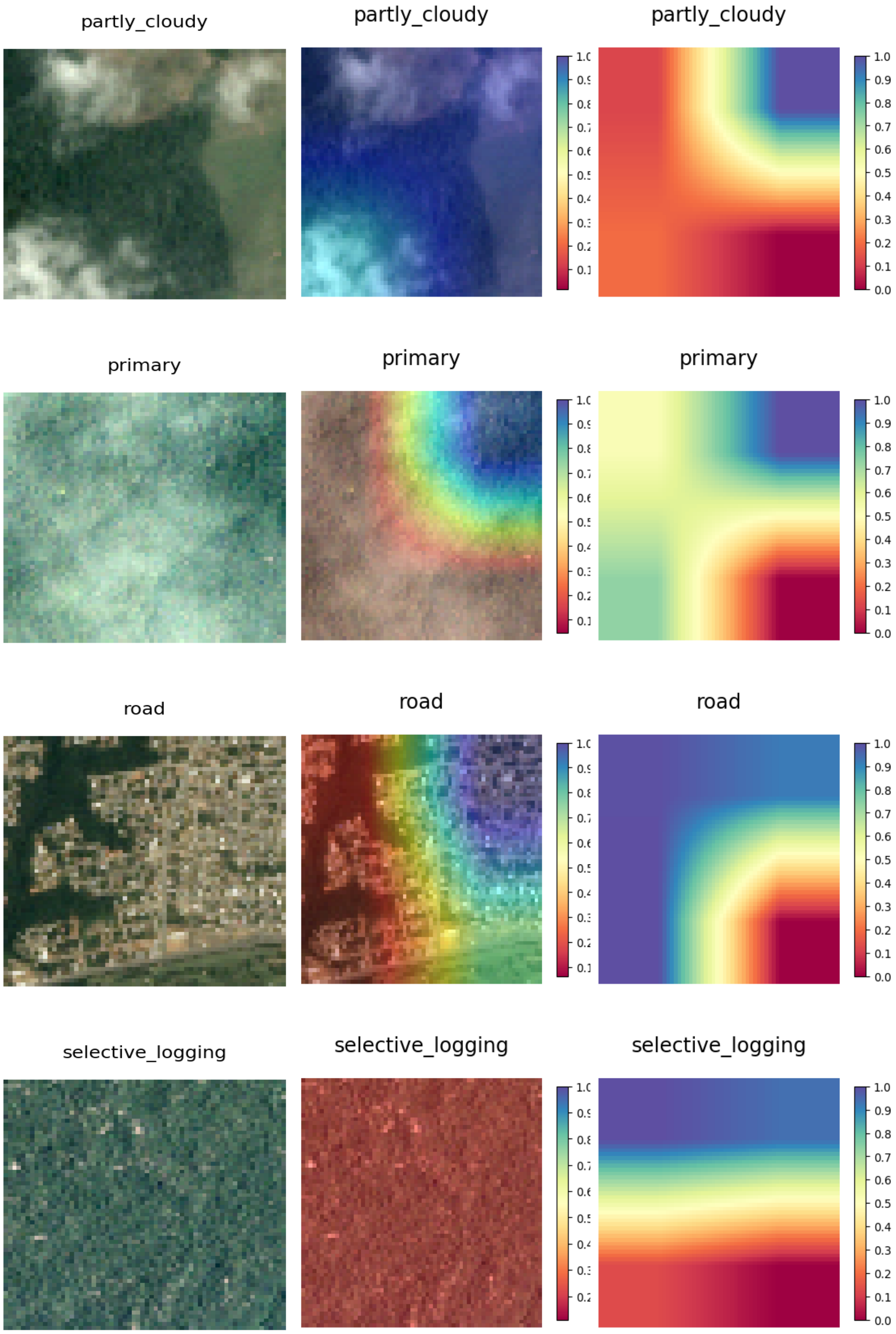


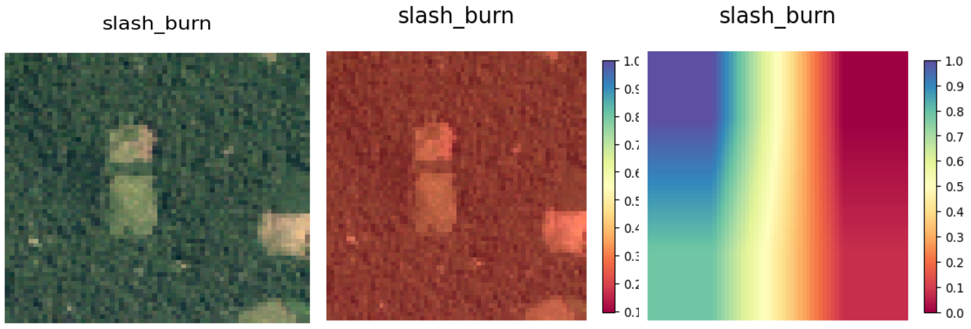


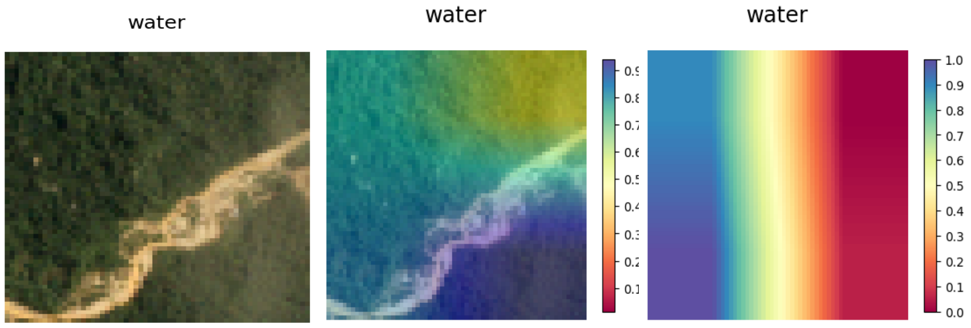



blooming
blooming
blooming
blow_down
blow_down
blow_down
clear
clear
clear
cloudy
cloudy
cloudy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


conventional_mine
conventional_mine
conventional_mine
cultivation
cultivation
cultivation
habitation
habitation
habitation
haze
haze
haze
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


partly_cloudy
partly_cloudy
partly_cloudy
primary
primary
primary
road
road
road
selective_logging
selective_logging
selective_logging
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


slash_burn
slash_burn
slash_burn
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


water
water
water
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Appendix C

Index of Acronyms and Abbreviations

AI: Artificial Intelligence (Τεχνητή Νοημοσύνη)

CFSR: Climate Forecast System Reanalysis (Συστήματος Ανάλυσης Κλιματικής Αλλαγής)

CNN: Convolutional Neural Network (Συνελικτικό Νευρωνικό Δίκτυο)

FLOPs: Floating Point Operations (Πράξεις Κινητών Υποδιαστολών)

GLM: General Linear Model (Γενικό Γραμμικό Μοντέλο)

GPU: Graphics Processing Unit (Μονάδα Επεξεργασίας Γραφικών)

LR: Learning Rate (Ρυθμός Μάθησης)

MLP: Multi-Layer Perceptron (Πολυεπίπεδος Perceptron)

MODIS: Moderate Resolution Imaging Spectroradiometer (Φασματοσκοπιόμετρο Απεικόνισης Μεσαίας Ανάλυσης)

NDVI: Normalized Difference Vegetation Index (Δείκτης Κονονικοποιημένης Διαφοράς Βλάστησης)

NDWI: Normalized Difference Water Index (Δείκτης Κονονικοποιημένης Διαφοράς Νερού)

NIR: Near Infrared (Κοντά στις Υπέρυθρες)

PCA: Principal Component Analysis (Ανάλυση Κυρίων Συνηστοσών)

RGB: Red Green Blue (Κόκκινο Πράσινο Μπλε)

SAR: Synthetic Aperture Radar (Κεραία Συνθετικού Διαφράγματος)

SVM: Support Vector Machine (Μηχανισμός Υποστηρικτικού Διανύσματος)

TPU: Tensor Processing Unit (Μονάδα επεξεργασίας Τανυστών)

UAV: Unmanned Aerial Vehicles (Μη-επανδρωμένες Εναέρια Οχήματα)

ViT: Vision Transformer (Μετασχηματιστής Όρασης)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Δασκαλόπουλος Ιωάννης** του **Μιλτιάδη**, με αριθμό μητρώου **18390278** φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής **Μηχανικών** του **Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών**, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

ΙΩΑΝΝΗΣ ΔΑΣΚΑΛΟΠΟΥΛΟΣ