INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Improving Point of Entry Database through Data Science: Deep Learning-based Identification of Unmapped Roads Using Remote Sensing Images

Matilde Soares Saraiva

Master Degree in Data Science

Supervisor:
PhD Tomás Gomes da Silva Serpa Brandão, Assistant Professor,
ISCTE - University Institute of Lisbon

Co-Supervisor:
PhD Diana E. Aldea Mendes, Associate Professor,
ISCTE - University Institute of Lisbon

June, 2024

BUSINESS SCHOOL

TECNOLOGIAS E ARQUITETURA

Improving Point of Entry Database through Data Science: Deep Learning-based Identification of Unmapped Roads Using Remote Sensing Images

Matilde Soares Saraiva

Master Degree in Data Science

Supervisor:
PhD Tomás Gomes da Silva Serpa Brandão, Assistant Professor,
ISCTE - University Institute of Lisbon

Co-Supervisor:
PhD Diana E. Aldea Mendes, Associate Professor,
ISCTE - University Institute of Lisbon

June, 2024

Para os meus pais e avós.

Acknowledgment

Um especial agradecimento é devido aos meus orientadores, os Professores Tomás Brandão e Diana Mendes, pelo seu valioso auxílio e orientação. Extendo este agradecimento ao Nuno Nunes, por me ter ajudado a definir o tema da tese, e à Carmen, por me ter apresentado ao Nuno numa altura menos clara desta etapa. Carmen, obrigada por estares sempre ao meu lado.

Agradeço também aos meus pais, Ana e Pedro, assim como aos meus avós, Carlos, Natália e Pulquéria, por serem os meus maiores exemplos de resiliência e fonte de apoio. À minha irmã, Madalena, agradeço pela infinita paciência e companheirismo. Vocês sempre me encorajaram e apoiaram com o necessário para que eu pudesse perseguir todos os objetivos a que me pruposesse. Sou verdadeiramente sortuda por ter uma família com uma presença tão forte e positiva.

Estendo ainda os meus agradecimentos a todos os meus amigos, pelo constante encorajamento, mencionando em especial o Pedro, o Zé e o Diogo. Agradeço-vos pela amizade, ajuda e divertimento nestes últimos anos.

Por fim, um agradecimento muito especial é reservado para o Ricardo. Obrigada por me lembrares de que a vida é sempre mais simples do que eu tendo a imaginar. És excecional.

Resumo

Recorrendo a imagens de satélite e técnicas de aprendizagem profunda, esta Dissertação visa identificar interseções entre estradas e fronteiras terrestres, com vista à automatização do processo de atualização da base de dados de Pontos de Entrada, originalmente desenvolvida pelo programa "COVID-19 *Impact on Points of Entry*" da *International Organization for Migration*.

Utilizando Angola como área estudo, a Dissertação propõe uma abordagem baseada em classificação de imagens. Para isso, inicialmente extraíram-se imagens de satélite do ArcGIS Pro, em Angola, e criaram-se manualmente as legendas correspondentes. Posteriormente, foram criados conjuntos de dados de treino e teste, com as imagens divididas em pedaços de (64×64) pixels. O conjunto de teste contém exclusivamente imagens da zona fronteiriça de Angola, enquanto o conjunto de treino inclui imagens internas aos limites do país.

Criaram-se seis arquiteturas baseadas em Redes Neuronais Convolucionais (CNN) e utilizaram-se modelos pré-treinados com os dados da ImageNet (MobileNetV1 e ResNet50), com o propósito de investigar a melhor abordagem. Várias experiências foram desenvolvidas, recorrendo a cada arquitetura.

O modelo que atingiu o melhor desempenho é baseado numa CNN personalizada, composta por dois blocos com duas camadas convolucionais e uma camada de pooling. Este identifica corretamente 47 Pontos de Entrada, melhorando a base de dados de 7 para 47 pontos. A integração de métodos de ciência de dados com imagens de satélite, visa fornecer uma proposta mais automatizada para identificar novos Pontos de Entrada terrestres, relevante para a capacitação de organizações humanitárias e governamentais na monitorização, tomada de decisões e resposta a crises.

Palavras-Chave: Aprendizagem Profunda, Redes Neuronais Convolucionais, Imagens de Satélite, Classificação de Imagens.

Abstract

This Dissertation proposes an end-to-end framework for map creation using a data science approach to address the data gap identified in the International Organization for Migration's (IOM) COVID-19 Impact on Points of Entry program. Leveraging satellite imagery and deep learning techniques, the study aims to identify relevant nodes within complex networks of land roads and borders to augment the Points of Entry database, focusing on Angola as a proof-of-concept area.

Initial tasks involve data collection, namely satellite imagery extraction from ArcGIS Pro and the manual creation of corresponding ground truth data. Subsequently, train and test datasets are prepared, with images divided into (64×64) pixel pieces. The test dataset exclusively comprises data from Angola's border area, while the training dataset includes images from within the country's boundaries.

The Dissertation's training phase encompasses two main sections: Custom-Built and Pre-Trained models. Six custom-built Convolutional Neural Network (CNN) architectures are designed, alongside experiments using pre-trained models (MobileNetV1 and ResNet50), pre-trained with the ImageNet dataset, with fine-tuning applied.

The best-performing model is based on a CNN, consisting of two blocks with two convolutional and one pooling layers, correctly identifies 47 Points of Entry, enhancing the database from 7 to 47 Points of Entry. By integrating data science methods with satellite imagery, the study aims to provide automated mechanisms for identifying relevant nodes in complex networks, empowering humanitarian and governmental stakeholders to monitor, make informed decisions and respond efficiently to emerging challenges.

Keywords: Deep Learning, Convolutional Neural Networks (CNN), Satellite Images, Image Classification.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1. Background and Motivation

Remotely sensed data plays a key role in enhancing our understanding of the world. It serves as a powerful tool for capturing detailed information about the Earth's surface, including land cover, land use, environmental changes and infrastructure development. Using the knowledge available within this domain, researchers, non-governmental institutions and policymakers can make informed decisions and create sustainable development plans aimed at addressing global challenges.

Throughout the years, humanitarian organizations have increasingly turned to digital tools to assist and protect populations affected by conflict and crises. Recent advancements in computational capabilities, combined with an abundance of data, have greatly contributed to a broader adoption of digital technologies within the humanitarian field. This progression has transformed the approach of humanitarian action from reactive to preventive [3]. On that premise, artificial intelligence technologies offer the potential for expanding the toolkit available to use in humanitarian missions across three main dimensions: preparedness, response and recovery. Preparedness involves an ongoing effort to comprehend the potential risks at hand and suggest strategies to address them, ultimately enhancing the efficiency of humanitarian responses to crises and emergencies. Response primarily centres on providing aid to those requiring assistance, while recovery entails initiatives that extend beyond immediate relief efforts [3].

In the scope of this Dissertation, the applications associated with preparedness and response actions are primarily addressed. Data Preparedness (DP) consists of organizations' ability to effectively deploy and manage data collection, analysis tools, techniques and strategies in a specific operational context prior to a disaster [23]. By addressing challenges such as data disparity, distortion and damage, DP helps build trust between partners and affected communities [23].

Remarkably, recent progress in deep learning, natural language processing and image processing contributes to a more prompt and precise response to emergencies. An illustrative example involves leveraging artificial intelligence (AI) technologies for mapping disaster-stricken areas, with great effectiveness, as it is presented by initiatives like the OpenStreetMap (OSM) project [28]. This project uses AI systems to map disaster-affected regions by incorporating crowd-sourced social media data and satellite and drone imagery to provide reliable information, aiding in prioritizing response efforts [3]. Another example is the Rapid Mapping Service, a joint effort led by the United Nations Institute for Training and Research, the UN Operational Satellite Applications Program, and UN

Global Pulse. This initiative leverages AI to analyze satellite imagery, facilitating rapid mapping of flooded areas and assessing damage caused by conflicts or natural calamities such as earthquakes and landslides. Despite that, in some scenarios, such as armed conflicts, the applicability of these tools may be limited. On the one hand, this context presents challenges such as disinformation campaigns which impact data reliability. On the other hand, difficulties in accessing high-quality data during conflicts may obstruct the design and development of AI systems, compromising the suitability of their mapping tools [3].

While such technologies offer opportunities to enhance humanitarian relief responses, it is crucial to recognize that they do not present solutions for every scenario within the humanitarian domain. Consequently, it is crucial in informing humanitarian responses on-site [3]. The COVID-19 pandemic introduced unprecedented containment measures worldwide, aimed at curbing human mobility to stem the spread of the virus. Within the context of this global crisis, the International Organization for Migration (IOM) has developed a comprehensive global mobility database, which serves as a valuable tool for mapping, tracking and analyzing the pandemic's impact on Points of Entry (PoE) across the country's borders subject to restrictive measures [10].

The primary objective of this database is to provide valuable insights to IOM Member States, UN partner agencies, voluntary partner agencies and other stakeholders. By understanding the evolving situation, stakeholders can tailor their response strategies accordingly, particularly in addressing the specific needs of migrants and mobile populations who are disproportionately affected by mobility restrictions. Ultimately, a global mobility database can serve as an essential resource for civil society, including the media and the general population. It provides up-to-date information about mobility restrictions related to airports, land and blue borders, helping to keep communities informed about the measures in place.

As the COVID-19 pandemic transitioned into an endemic stage, the ongoing need to maintain an updated database of PoE remains highly relevant in the context of humanitarian efforts. In this regard, there can be identified room for further improvement. A significant concern lies in relying on OSM data to automatically identify PoE. It is recognized that the OSM database presents data gaps in less developed or sparsely populated regions, thus presenting challenges in accurately pinpointing PoE in these areas.

Further development of this area is essential to enhancing the effectiveness of future initiatives and ensuring a comprehensive representation of all regions, regardless of their level of development, within the database. By improving data collection methods and expanding data sources, organizations like the IOM and other humanitarian or governmental bodies can strengthen their ability to monitor, make informed decisions, and respond efficiently to new challenges. This collaborative approach enhances readiness and resilience against global crises, promoting a sustainable and secure future for everyone.

In response to the constraints identified in the creation of the DTM's Report - IOM's COVID-19 Impact on Points of Entry initiative [10], this Dissertation proposes a novel approach for extracting knowledge from remotely sensed data using deep learning tools. The objective is to identify intersections of roads and land borders by leveraging satellite imagery combined with Convolutional Neural Networks (CNNs). By developing an image classification analysis of land roads, this Dissertation aims to identify new Points of Entry (PoE), ultimately seeking to present an alternative methodology that supplements traditional sources such as OpenStreetMap (OSM).

1.2. Objectives and Research Questions

In an era marked by rapid technological advancements, leveraging the available tools is essential for the humanitarian aid sector to effectively address the challenges that are posed. This Dissertation introduces a Computer Vision methodology for map creation using remotely sensed data, addressing the gap in mapping less developed or sparsely populated regions identified by the IOM's COVID-19 Impact on PoE program.

By leveraging satellite imagery of Angola's territory as the basis for the proof of concept, with deep learning techniques, an approach based on image classification algorithms for road detection is used to identify roads intersecting the country's border lines. This approach is able to provide a general overview of the identified roads, providing a computationally efficient alternative to the widely used pixel-wise analysis techniques. Additionally, it is aimed to obtain comparable accuracy levels while using a relatively small amount of input data. Furthermore, this Dissertation compares the effectiveness of using pre-trained neural network models versus training these models from the ground up - presented in the Literature Review Chapter (Chapter 2).

Ultimately, this Dissertation seeks to demonstrate that the insights gained from a simplified approach, such as image classification, can significantly contribute to identifying road networks and enriching information within PoE databases. The Dissertation operates under the premise that it is preferable to quickly determine that within a given region exists a road, rather than risk overlooking its existence due to inadequate information. The idea is to offer an accessible and expedited solution to map creation.

In order to achieve the goals that have been previously described, a set of research questions has been set out to serve as guiding principles for the Dissertation:

RQ1: Can reliable ground truth data be extracted using fully-automated processes?

RQ2: Can Pre-Trained models present a better performance at extracting roads from satellite imagery than training a new model from scratch?

RQ3: Can an Image Classification-based approach identify relevant nodes in complex networks of land roads and borders with accuracy?

RQ4: Is it possible to correctly identify relevant nodes in complex networks of land roads and borders?

These research questions summarize the core objectives of the Dissertation, guiding the analysis and interpretation of the findings as the study progresses. In an effort to

contribute to the state-of-the-art in remote sensing and geospatial analysis, these questions offer a practical solution to real-world challenges.

1.3. Methodology and Organization of the Dissertation

This Dissertation draws inspiration from the Cross-Industry Standard Process for Data Mining (CRISP-DM) reference model [6], which provides a structured and systematic methodology for exploring and interpreting data. Aligned with the principles of the CRISP-DM methodology, this Dissertation follows an inferential approach, where various experiments are performed, and parameters are adjusted based on the results - illustrated in Figure 1.1.

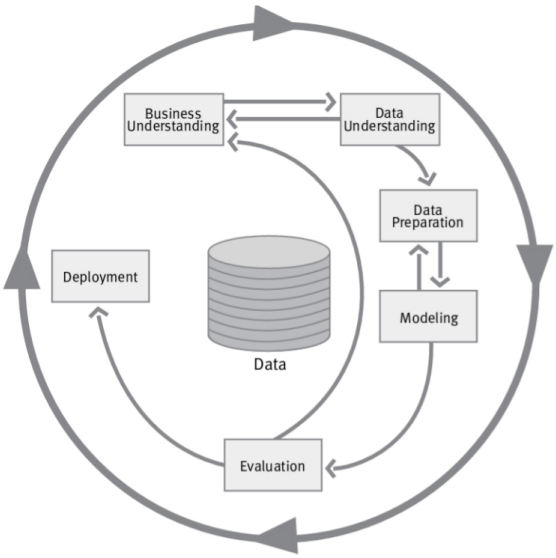


FIGURE 1.1. Phases of the CRISP-DM reference model. Figure reprinted from [6].

The Introduction and Literature Review Chapters (Chapters 1 and 2), focus on gaining a thorough understanding of the subject matter. These chapters aim to delineate the goals of the Dissertation, introduce key concepts relevant to the topic, clarify the potential contributions to the field of study and outline the planned approach to achieving these objectives. This section lays the groundwork for a comprehensive Business Understanding.

Following this, the Data and Tools Chapter (Chapter 3) provides an overview of the resources to be used throughout the Dissertation. It encompasses the data understanding stage (Section 3.1), an exposition of the proposed models' architectures (Section 3.2) and an introduction to the evaluation metrics to be employed in assessing the Dissertation's outcomes (Section 3.3).

Chapter 4 intends to promote informed decision-making in the subsequent phases of the Dissertation. The data preparation tasks that were performed are outlined, as well as a description of the additional experiments that enhance the Dissertation's scope. Through the developments of early experiments and acquisition of further methodological insights, the groundwork is laid for the analysis phase.

In Chapter 5, the processes and results of the training phase are presented. The training workflow details the data preparation and modeling tasks conducted throughout this phase. Subsequently, an in-depth analysis of the results is displayed, serving as a foundational step for the subsequent test phase.

The test phase is detailed in Chapter 6. Following a similar structure to Chapter 5, the methodological specifications of the test phase are presented, followed by a detailed analysis and discussion of the test's results. The analysis of the results serves the purpose of assessing the model's performance and effectiveness in meeting the Dissertation's objectives. In the concluding Chapter, the Dissertation's strengths and weaknesses are examined again, offering a comprehensive reflection which aims to provide insights for forthcoming studies.

CHAPTER 2

Literature Review

This Chapter presents a review of the fundamental concepts of data science in the field of satellite imagery analysis. First, an introduction to the key concepts related to remotely sensed data and deep learning methodologies is presented. Second, a comprehensive review of the research work in the Dissertation's domain is undertaken, providing a comparison of the diverse approaches and relevant contexts. Lastly, the conclusions drawn from the theoretical review will be presented.

2.1. Remotely Sensed Data

Observing and understanding Earth's dynamic systems is essential for addressing environmental challenges and managing resources through a knowledge-based approach. Researchers can leverage technologies that remotely collect data and analyze various phenomena worldwide. Such technologies can enable tracking temperature changes in oceans, weather forecasting, identifying erupting volcanoes and studying urban, agricultural or forested area changes. The wide range of applications highlights the critical role of remote sensing in enhancing our understanding of Earth's dynamics and informing evidence-based decision-making for the benefit of society and the environment [32].

The Copernicus initiative supports the EU's role as a global actor and contributes to solutions to common global challenges. Sentinel satellites, developed specifically for the Copernicus program, provide detailed observations of Earth's atmosphere, marine environments, land surfaces, climate change impacts, emergency response and security. By harnessing data from Sentinel missions such as radar imaging, optical imagery, ocean and land measurements, atmospheric composition monitoring and sea surface topography, Copernicus supports informed decision-making and scientific research across multiple disciplines [7].

On this premise, remote sensing consists on the acquisition and monitoring of the physical characteristics of an area by capturing and analyzing the reflected and emitted Electromagnetic Radiation (EMR) from a distance - typically through the use of satellites or aircraft [32]. The aircraft instruments or satellites can capture data over large geographic areas in a single observation. By analyzing the electromagnetic properties of Earth's surface, oceans and atmosphere, remote sensing contributes greatly to its identification and classification. EMR is characterized by energy propagation at the speed of light, following a regular wave pattern. This means that all waves internal to the electromagnetic spectrum are uniformly spaced and exhibit a repetitive nature over

time. The Electromagnetic Spectrum aggregates all EMR categories - visible light, radio waves, infrared and gamma rays - varying in wavelength and frequency through the Electromagnetic Spectrum [26].

Remote sensing relies on how EMR interacts with different matters like trees, water or atmospheric gases. When EMR encounters these materials, they can be absorbed, reflected, scattered, emitted by them, passed through or transmitted. The essence of remote sensing lies in the ability to detect and record the EMR that is reflected or emitted by objects or materials. Each matter has its own spectral signature, which is the unique properties that emit or reflect EMR. Remote sensors are designed to capture and analyze spectral data, identifying and differentiating various objects and materials [26].

Spectral information can be gathered through passive or active means. Passive sensors observe the energy naturally emitted or reflected by an object (Figure 2.1(a)), using devices such as radiometers and spectrometers. These sensors are commonly employed in remote sensing across various parts of the electromagnetic spectrum, including visible light, infrared, thermal infrared and microwave wavelengths [21]. The Landsat Mission is the longest running mission using passive sensors to collect spectral data, but Maxar and Planet Labs are also examples of commercial satellites that are used for the same purpose [11].

Active Sensors, on the other hand, emit a pulse of energy and then analyze the alterations in the returning signal (Figure 2.1(b)), generally functioning within the microwave range of the electromagnetic spectrum, enabling them to penetrate the atmosphere effectively under various conditions [20]. Examples of this type of satellite include the Canadian Space Agency's RADARSAT-1 and RADARSAT-2, Airbus Defense & Space TerraSAR-X Radar Satellite and LiDAR, which uses light emitted from aircraft or helicopters to measure bounce-back time to the sensor [11].

(a) Passive Sensor (b) Active Sensor

FIGURE 2.1. Diagram of a passive sensor versus an active sensor. Figure reprinted from NASA's Applied Sciences Remote Sensing Training Program, available in [22].

Geospatial characteristics are inherent in remotely sensed data, which means that the observed areas are tied to their geographic reference, using a given coordinate system. This allows for the mapping and analysis of the data in conjunction with other spatial information, such as road networks or population density maps, among other applications. For this reason, remote sensing data can fruitfully contribute as a data source for

geographic information systems (GIS), which encompass "organized collections of computer hardware, software, geographic data and personnel designed to efficiently capture, store, update, manipulate and analyze all forms of geographically referenced information" (Jensen, 2005; ESRI, 2001, as cited in [26]).

The choice of sensor data in research is influenced by resolution, which varies based on factors like the satellite's orbit and sensor design. There are four main types of resolution to consider when examining any remotely sensed data: radiometric, spatial, spectral and temporal. Radiometric resolution refers to the level of detail in each pixel, represented by the number of bits capturing energy. Spatial resolution relates to the size of pixels in a digital image and the corresponding Earth surface area. Spectral resolution involves the sensor's ability to distinguish between different wavelengths, with multi-spectral sensors typically having 3-10 bands and hyper-spectral sensors possessing hundreds or thousands. A narrower wavelength range per band indicates finer spectral resolution. Temporal resolution measures the time it takes for a satellite to orbit and revisit the same observation area [22].

Advancements in satellite sensors with sub-meter resolution present new opportunities for detailed urban land cover mapping at the object level. Such advancements, particularly with finer-scale data, introduce increased complexity as it is crucial to enhance the efficiency of high-resolution classification methods in order to effectively handle the challenges associated with classifying high-resolution imagery [35]. As a consequence, choosing appropriate sensor data constitutes the initial crucial step for achieving a successful classification task, tailored to a certain purpose [17]. Ultimately, these processes allow researchers to gain insight about the Earth, providing a broader view of the Earth's surface, compared to ground-level observation. Remotely sensed imagery has emerged as a real-time and cost-effective method for mapping land cover [35].

When working with remote sensing data, one other crucial aspect is concerning the acquisition of ground truth data to support the analysis. Various techniques are available, namely manual, fully-automated or semi-automated. Manual labeling of reference data involves a time-consuming task of human-driven annotation. Although this approach has proven to yield better results in terms of model performance, it poses challenges due to its resource-intensive nature, particularly when dealing with large-scale imagery datasets. Fully-automated data labeling involves the automatic extraction of reference data. One commonly used tool for fully-automatic labeling is OSM crowd-sourced data, which, as it was stated in the introductory Chapter, has been used in the context of the IOM's COVID-19 Impact on Points of Entry initiative. OSM stands out for its abundance of labeled features, global availability and contributions from diverse entities, including individuals, Non-governmental organizations and corporations. However, several authors note that OSM still exhibits an under representation of data in less developed regions. This poses challenges in terms of ensuring the quality and correctness of available data,

addressing temporal and spatial alignment issues between imagery and OSM data and managing the high precision but low recall of OSM's tagged features [5].

2.2. Deep Learning Fundamentals

In recent years, the AI field has experienced great advancements, particularly in the deep learning (DL) field. Within the machine learning (ML) domain, DL emerges as a sophisticated statistical approach primarily used for pattern recognition, relying on neural networks with multiple layers. It represents a powerful tool for automated pattern recognition and classification, with implications across diverse fields where complex data analysis is required [19].

In the following Sub-Sections, the main theoretical aspects concerning Deep Learning will be described, in the form of a review of the underlying concepts and their practical applications in a variety of domains.

2.2.1. Basic Neural Networks

Known for their ability to learn complex patterns and data representations, neural networks (NN) have enabled machines to perform tasks with human-like accuracy and efficiency, due to its biologically-inspired learning mechanisms in the form of computational models [1]. In NN, data flows from input layer to output neurons, adjusting connection weights along the way. This adjustment, driven by training data consisting of input-output pairs, facilitates learning. The goal is to refine the weights in a mathematically justified manner to minimize prediction error for each example, improving prediction accuracy in subsequent iterations. Through the iterative refinement of weights across multiple input-output pairs, NN enhance their learning capacity over time, resulting in more accurate predictions and eventual model generalization — which is the ability to process unseen instances after training on a finite set of input-output pairs.

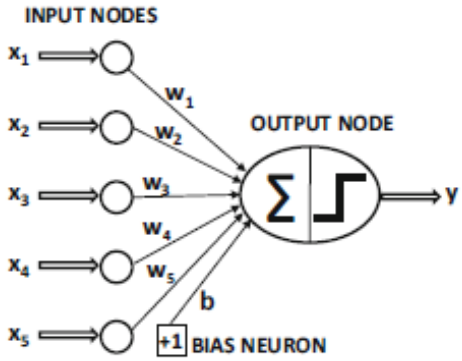


FIGURE 2.2. The basic architecture of the Single-Layer Perceptron with bias. Figure adapted from [1].

As a baseline, a NN with a single computational layer is commonly referred to as the Single-Layer Perceptron, which consists of an architecture where inputs are directly mapped to outputs through a generalized version of a linear function [1]. As illustrated in Figure 2.2, the input layer contains d nodes, each corresponding to one of the d features

in the input vector ($X = [x_1, \ldots, x_d]$). The nodes transmit the input features to the output node via edges with weights ($W = [w_1, \ldots, w_d]$), with which the features are multiplied and added at the output node. Subsequently, the sign function is applied in order to convert the aggregated value into a class label, i.e. to predict the dependent variable associated with X. It is then necessary to incorporate an additional bias variable b to account for the invariant component of the prediction. The predicted output $\hat{y}$ is determined by taking the sign of the computed value, represented as [1]:

$$\hat{y} = \text{sign}(W \cdot X + b) = \text{sign}\left(\sum_{j=1}^{d} w_j x_j + b\right) \tag{2.1}$$

Furthermore, the addition of one or more hidden layers between the input and output layers can contribute with several critical capabilities, compared to the Single-Layer Perceptron. The Multi-Layer Perceptron (MLP) can learn more complex, non-linear decision boundaries, namely hierarchical features, through multiple layers - as presented in Figure 2.3. Each hidden layer can learn increasingly abstract representations of the input data. This behaviour increases the model's capacity and tendency to generalize better to unseen data. Therefore, hidden layers are an essential component of the MLP, consisting of neurons that perform intermediate computations before passing the results to the next layer. These layers are termed "hidden" because their values are not directly observed in the input or output; they are internal to the network. Each neuron within a given hidden layer receives inputs from the previous layer, processes them through a weighted sum, adds a bias and applies an activation function to produce an output that serves as the input for the next layer.

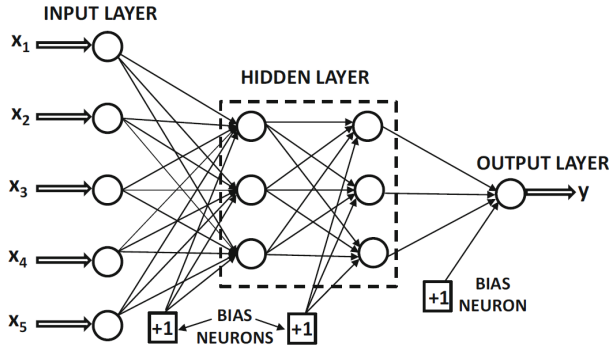


FIGURE 2.3. The basic architecture of the Multi-Layer Perceptron with bias. Figure adapted from [1].

Additionally, activation functions act as the mechanism that introduces non-linearity to a NN, enabling the network to address complex learning tasks [25]. The Sign activation function suits scenarios where output values need to be either -1 or $+1$, while the Sigmoid activation function presents suitability for transforming values into probability estimations - ranging from 0 to $+1$, it is adequate for binary classification tasks. Similarly, the Tanh activation function ranges from -1 to 1 and is preferable to use, rather than the Sigmoid,

when the outputs of the computations are desired to be both positive and negative. For regression tasks involving the prediction of real values, the Identity activation function is often suitable because it preserves the input values without introducing any non-linearity, aligning well with the continuous nature of the prediction [1].

The Rectified Linear Unit (ReLU) activation function is particularly popular for its ability to allow large values to pass through, resulting in sparsity and keeping certain neurons inactive. It maps negative inputs to 0 while leaving positive inputs unchanged. This intermittent firing accelerates the training process, as its gradient is computationally efficient, resulting in lower computational costs. These attributes guarantee that the ReLU activation function is effective in solving a wide range of problems, namely the Vanishing Gradient problem. The Vanishing Gradient can occur during the training of a NN, when the gradients become very small, the weight updates during training are tiny, causing the training process to be very slow or even to stall completely. This means the network is unable to learn effectively, especially in the earlier layers [1].

The selection of an appropriate loss function also depends on the nature of the output and the specific requirements of the task. Loss functions are essential in training ML models as they quantify the disparity between predicted and actual values, thereby guiding the optimization process to enhance the model's performance. In regression problems where the goal is to predict continuous values, Squared Loss is frequently used. This is because Squared Loss penalizes larger errors more heavily, which is often desirable in tasks where precise numerical predictions are required [1].

In binary classification with outputs ranging from 0 to 1, Binary Cross-Entropy Loss, also known as Logistic Loss, is often chosen due to its suitability for handling probabilities and predicting class likelihoods. In multi-class classification scenarios where the output denotes a probability distribution across multiple classes, Cross-entropy Loss is typically utilized [1].

2.2.2. Convolutional Neural Networks

Renowned for their specialized architecture, Convolutional Neural Networks (CNN) were designed for processing structured grid data such as images. Due to its ability to capture spatial hierarchies in data, CNN consist on a specialized type of NN. It incorporates parameters like weights and biases, that are combined with inputs and passed, amongst intermediate layers, through non-linear activation functions to generate outputs [25].

In CNN architectures, each layer is three-dimensional, with depth corresponding to the quantity of feature maps. It is essential to differentiate the concept of "depth" within a single layer of a CNN, from the depth concerning the number of layers. In the input layer, input features correspond to color channels, typically RGB (red, green and blue). For instance, if the input is grayscale, the input layer will have a depth of 1, but subsequent layers may still have a depth of 3 due to the architecture's design. While hidden layers are responsible for transforming the input data into feature maps, which are the intermediate outputs that capture the presence of specific features detected by the filters

in the convolutional layers. These feature maps, through successive layers, enable the network to learn and represent complex patterns in the data, facilitating tasks like image recognition and classification.

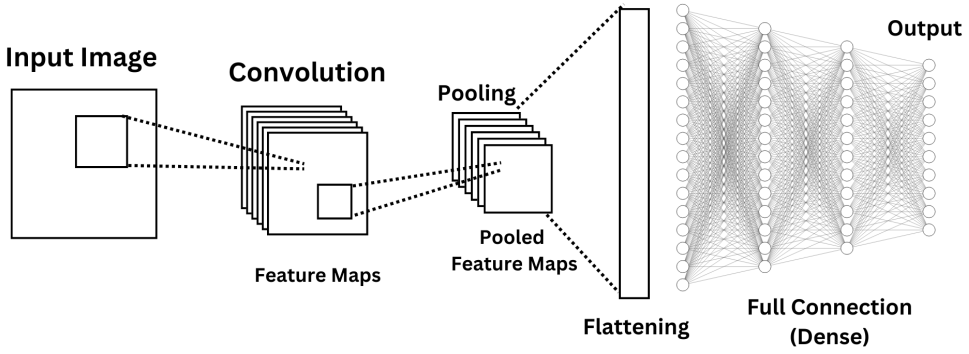


FIGURE 2.4. The basic architecture of a Convolutional Neural Network. Figure adapted from [27].

In CNN architectures, there are two main types of hidden layers: Convolutional and Pooling [1], as illustrated in Figure 2.4. Within the convolutional layers, a convolution process occurs. This involves applying a filter, also referred to as kernel, to alter activations from one layer to the next, maintaining spatial relationships but with decreased spatial dimensions. This process involves computing the dot product between the filter's weights and various spatial regions internal to a layer, determining the hidden state value in the subsequent layer. At each possible position inside a layer, the interaction between the filter and spatial regions occurs, ensuring that the subsequent layer retains spatial connections from the previous layer. This process defines the activations of the next layer, with sparse connections between layers, as each activation in a given layer is influenced by only a small region of the previous layer. Except for the final layers, all other layers maintain their spatial arrangement, allowing for visual analysis of how different areas of an image affect specific portions of activations inside a layer. Initial layers capture basic shapes such as lines, while deeper layers extract more complex features [1].

Subsampling layers, alternatively referred to as Pooling layers, are positioned between convolution layers to diminish the image size across layers via sampling. This sampling involves selecting either the maximum or average value within a specified window. Pooling serves as a regularization technique to prevent overfitting, by reducing spatial dimensions. It operates on all feature channels and can employ various steps [25].

2.2.3. Transfer Learning

Pre-trained CNN architectures, often sourced from publicly available repositories such as ImageNet, expedite development by providing a foundation for fine-tuning specific tasks. These models have been trained on large datasets, utilizing significant computational resources and time to learn how to extract valuable features or representations from the data. Leveraging pre-trained networks saves time and computational resources, which

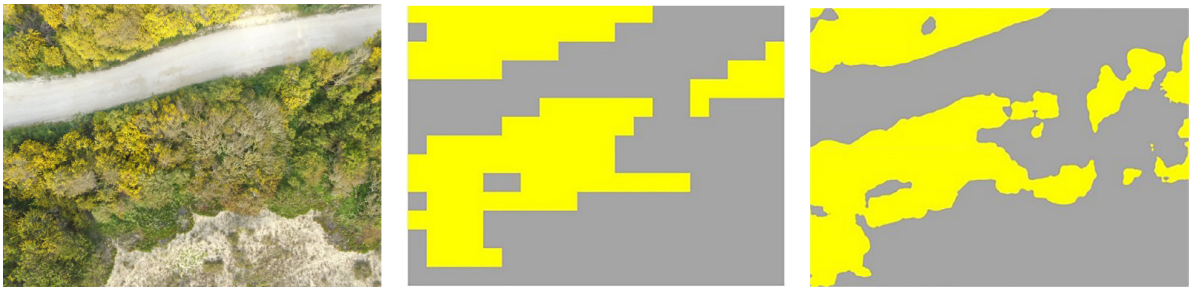
is especially useful when working with limited data while enabling a diverse range of applications.

Transfer learning is widely used in NN, where knowledge from one task or dataset is applied to another related one. It involves using a pre-trained model, originally trained on a large dataset for a specific task like image classification or natural language processing, and adapting it for a different task or dataset. Fine-tuning is a specific type of transfer learning where the pre-trained model is further trained (or fine-tuned), adapting the model's parameters to better suit the new task. This method improves efficiency and performance, especially when the new task is similar to the original task the model was trained on [1][25].

The ImageNet database is an extensive collection of over 14 million images spanning 1000 different categories. It covers a wide range of visual concepts, making it comprehensive enough to represent most types of images encountered in everyday life. The ImageNet dataset's size and diversity make it highly representative of key visual concepts, making it a popular choice for training CNN. Pre-trained CNN models trained on ImageNet can effectively capture and represent various visual features present in images. These pre-trained models can then be used to extract features from unseen images, enabling transfer learning across different applications and datasets. This approach creates multidimensional representations of image data, suitable for use with traditional Machine Learning methods, effectively transferring the knowledge and visual concepts learned from ImageNet to other tasks [1].

2.3. Related Work

Deep learning techniques have demonstrated significant efficacy in addressing various challenges in computer vision tasks, making substantial contributions to the field of remote sensing [13]. They enhance classification accuracy while optimizing computational efficiency, aligning with the growing need for precise outcomes and resource efficiency in remote sensing applications [13]. This Section focuses on the analysis of related research work, organized into the three main processes central to the Dissertation's objectives: Classification, Segmentation and Labeling.



(a) Original Image (b) Image Classification (c) Image Segmentation

FIGURE 2.5. Deep Learning approaches to image analysis. Figure adapted from [12].

Figure 2.5 presents an example illustrating the presence or absence of an invasive flora species. The original image is shown in Figure 2.5(a). Image classification involves assigning a label to an entire image (or block), indicating the presence of an object along with a degree of confidence, portrayed as an output block's collage in Figure 2.5(b). In contrast, segmentation entails classifying individual pixels to provide a detailed separation of objects within an image, as shown in Figure 2.5(c) [25].

2.3.1. Studies Based on Image Classification

As previously introduced, image classification consists in assigning a label to an input image, based on its visual content. The workflow typically begins with an input image, which is then processed by a classification model. The model initially extracts features from an image, which can consist of representations that are learned during the model's training phase - such as shapes, textures and patterns. Once the features are extracted, the model then uses them to predict the label of the image. The final output of the image classification process is a predicted label or class for the input image, which indicates what the model believes to be the most appropriate category for the image content [1]. The image classification approach demonstrates its advantage by consuming less time while providing an overview of the information that the image contains [12].

In an attempt to classify zebra crossings, Berriel et al. (2017) resorted to an image classification approach by addressing the challenges associated with the scarcity of data regarding crosswalk locations globally [4]. The study's key findings state that the model based on the VGG architecture achieved the best results among different evaluated architectures (AlexNet, VGG, GoogLeNet). The research demonstrated consistency across different levels of locality (city, country or continent) and achieved an average accuracy of 96.9% in intra-based experiments, emphasizing the VGG model's capability to generalize across different regions.

Bonafilia et al. (2019) addressed a building detection task by employing binary image classification on block images to identify the presence or absence of buildings. The authors used a seed dataset (D) and a combined dataset (D') that integrated weakly-supervised and semi-supervised training techniques with OpenStreetMap (OSM) data to locate buildings in high-resolution satellite imagery. The models were trained using ResNets with 18, 34 and 50 layers. They observed overfitting in the 34 and 50-layer ResNet models when using the D dataset. However, the 50-layer ResNet architecture performed best with the D' dataset. The study concludes that training with OSM ground truth data alone yields high-quality results [5]. Furthermore, they suggest that even stronger results can be achieved by pre-training models with global OSM data and fine-tuning with a small amount of manually labeled data for specific regions of interest.

The sliding window technique is also frequently used to assist in the creation of datasets that support the identification of objects within a subset of pixels. Such technique has been used to identify palm trees in high-resolution satellite imagery [15] through the use of a LeNet CNN architecture and a dataset created by a 17x17 pixel sliding window with

a step of 3 pixels, where each class is determined by the existence of a palm tree in the center of the window. Achieving accuracy values ranging from 94.00% to 98.77% across different regions, the study proves to outperform traditional computer vision methods. A similar approach is employed to identify another flora species - *Acacia Longifolia* - by sampling the original images (4000 x 3000 pixel, RBG) into 200 x 200 image patches, for binary and multi-classification approaches [12]. When employing the sliding window method, it is important to carefully consider the size of the step used. A step that is too small might decelerate the task, while a step that is too large could result in missing relevant data.

2.3.2. Studies Based on Image Segmentation

In efforts to attain detailed information, image segmentation contributes with pixel-wise approaches. In addition to the study that aims to identify the existence of *Acacia Longifolia* [12], the authors also performed a segmentation-based approach. Both approaches (classification and segmentation) achieve satisfactory results in conformity with the task goals, However, the segmentation approach provides detailed information but lacks sensitivity to small changes in the image.

As the use of a Fully Convolutional Neural Network (FCN) has been proven to reduce the number of trainable parameters while maintaining the model's generalisation ability, a binary, pixel-wise task as been experimented by Maggiori et al. (2017) [18], in an attempt to boost the models performance when using low-quality ground truth data. The authors conclude that the use of an FCN architecture outperforms the reviewed CNN approaches, showing improvements in accuracy. In an attempt to solve a similar challenge, one other approach was experimented with by Demir et al. (2018), using a ResNet18 backbone and Focal Loss to solve a binary road extraction challenge.

Aiming to create a more exhaustive and up-to-date global road network dataset, Keijzer et al. (2022) presented an automated road extraction approach using Sentinel-1 Synthetic Aperture Radar data. Using an architecture designed for image segmentation, the U-Net architecture demonstrates similar accuracy results, across various study areas and environmental differences. Additionally, upon comparing the outcomes of the research's model results [14] with the GRIP dataset - a multi-class dataset created by combining existing national and supranational road maps manually, it has been noted that the U-Net model has detected more roads, especially local roads.

The D-DenseNet architecture was introduced by He et al. (2022), aiming to enhance the precision of segmentation while simultaneously reducing the computational power required for the model. This method [13] involves two main stages: 1) altering the dilated convolutions to capture global context information throughout the entire network; and 2) the rearrangement of the stem block as the initial block to boost the network's ability to acquire broader context information. Another attempt to solve this problem has also been introduced [16], through the use of a four-stage approach where: 1) baseline U-Net model with seven pooling layers (without pre-train); 2) LinkNet34 with a pre-trained encoder

but with no dilated convolution in the center part; 3) an ensemble of the two previous approaches; 4) D-LinkNet with a pre-trained encoder. The D-LinkNet uses a ResNet34 pre-trained on ImageNet's dataset as its encoder, designed to receive, as input, 1024 x 1024 images with several pooling and dilated convolutional layers. This approach achieved the best model's performance on the validation set. Even though D-LinkNet's model shows promise in addressing certain road properties, such as narrowness, connectivity and complexity, it still faces challenges relating to the recognition and road connectivity [16].

Contrarily, in another study, He et al. (2022), opted to replace the original backbone of D-LinkNet with a DenseNet, instead of ResNet, attempting to expand the receptive field and incorporate more feature information into the model. The effectiveness of D-DenseNet presents the ability to strike a balance between model size and the segmentation's precision, revealing that the adaptations contribute significantly to the improved performance of the model [13].

The DenseUNet architecture is introduced in an effort to identify roads in high-resolution remote sensing images [33]. The DenseUNet architecture incorporates dense connections inside dense units. The dense connections enable feature reuse and help transfer information across different network layers. A suitable weighted loss function that assigns different weights to different types of pixels, with a focus on foreground pixels, is introduced to address challenges like occlusion by trees and shadows. The term "fractal extensions" is used in the context of simple connection rules, which involves the integration of deep supervision, identity mappings and diversified depth attributes. The DenseUNet achieves higher accuracy, F1-score, and kappa metrics than the classical segmentation methods (U-Net, SegNet, FRRN-B), particularly effective in scenarios with dense roads and shadows.

Finally, in a recent study, an approach for road detection from high-resolution satellite images employing the VGG19 architecture is introduced [9]. The proposed method is composed of a two-step process: 1) image segmentation to remove small objects based on semantic division, and 2) a combination of image segmentation with edge detection to enhance road detection. The VGG19 architecture is chosen for its good performance in the accuracy evaluation metric but also for its simplicity and requirement for a few parameters. Presenting an IoU value above 80%, outperforming the compared methods, the VGG19 architecture proves its effectiveness in extracting roads with proper accuracy.

2.3.3. Labeling Processes

The labeling process can be categorized into manual, fully-automated or semi-automated methods. The manual method revolves around the creation of ground truth data corresponding to the raster data in use, resorting to human-driven annotation. Despite being a time and resource-intensive task, manual labeling has proven effective in achieving superior results, compared to other methods. Various techniques for ground truth collection have been observed in the examined research work, where in studies with smaller

datasets, authors often opt for manually labeled data [4][12][14][15]. However, fully or semi-automated approaches are more commonly utilized.

Fully-automated label extraction from sources like OSM and the Google Static Maps API have been employed in the creation of larger and more diverse datasets. Comparing fully-automated and manual labelling approaches, used separately, authors have concluded that manual labeling leads to slight improvements in accuracy, compared to automatic labels [4]. Nonetheless, the fully-automated process has demonstrated the capability to acquire, annotate and classify satellite imagery on a global scale, showing promise for various applications.

In efforts to enhance model performance, a two-step approach has also been introduced [5][18], where models are initially trained using low-quality ground truth data extracted from OSM, then fine-tuned with manually labeled data. These semi-automated labeling approaches have been proven to be effective in capturing dataset generalities and improving precision [18]. It yields high-quality results, suitable for development and relief efforts, providing a pathway to extend models from well-performing regions to others [5].

2.4. Dissertation's Framework

The previous sections examined the theoretical foundations of the main topics related to the Dissertation. The key concepts of remote sensing data were introduced, providing further understanding of the characteristics inherent to this type of data. This includes technical insights, methods for data collection, types of image resolution and reference data. Subsequently, an introduction to Convolutional Neural Networks and their characteristics was conducted. This analysis includes an introduction to the parameters, advantages and limitations of CNN technology, providing a solid foundation for the Dissertation.

Additionally, a review of previous studies related to this topic was presented. It is possible to observe that image segmentation approaches are more commonly employed in similar research work than classification tasks. This preference may arise from the availability of big datasets or the need to establish more refined analyses through pixel-based approaches to similar problems. It was also observed that, even though fully-automated labeling tools have been proven advantageous for large-scale tasks, they yield constraints, particularly in accurately labeling areas with lower development levels or smaller population densities.

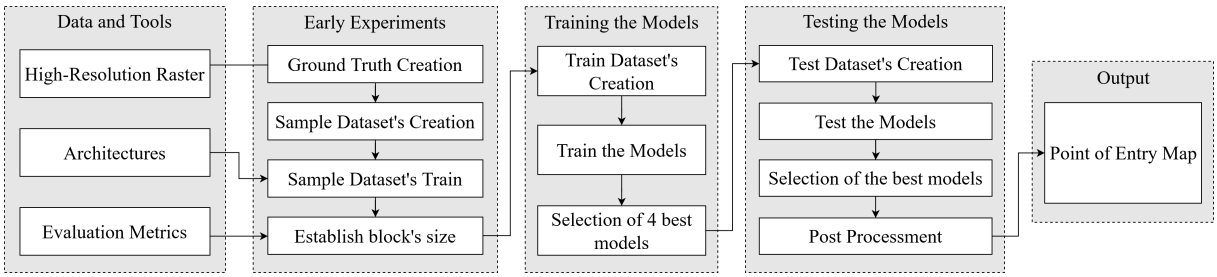


FIGURE 2.6. Dissertation's general workflow.

Building upon the insights gained in the previous sections, and aligned with the research questions presented in section 1.2, this Dissertation focuses on the binary classification of image blocks extracted from satellite imagery. The rationale behind this decision is that this Dissertation aims to offer a broad overview of roads, presenting a computationally efficient alternative to the commonly used pixel-wise analysis techniques. As presented in Figure 2.6, the details of the architectures and evaluation metrics used throughout this Dissertation are provided in Chapter 3. After that, early experiments are carried out to iterate the Dissertation approach (Chapter 4). Following this, the training and testing processes are implemented, as described in Chapters 5 and 6, contemplating the Dissertation's approaches and results.

CHAPTER 3

Data and Tools

This chapter comprehensively examines the materials used throughout the Dissertation process. It includes descriptions of the original data, model architectures explored (categorized into Custom-Built and Pre-Trained) and evaluation metrics supporting the Dissertation's analysis.

3.1. Data Understanding

This section offers an initial overview of the primary data utilized, setting the foundation for the Dissertation. Understanding the characteristics of the data is crucial to support the decisions taken throughout the study. Two distinct types of data are required for this Dissertation: raster and ground truth data. This Section solely focuses on presenting the Dissertation's raster data, while ground truth data is detailed in Section 4.1.

The raster data used in the Dissertation consists of satellite imagery extracted from Maxar's Vivid base map, which resulted from a collaboration between Esri and Maxar with the aim of enhancing Esri's Living Atlas. This agreement substantially improved data's spatial resolution, with half of the global landmass within the Living Atlas experiencing an upgrade from Esri's original 1.2-meter specification to a 60-centimetre spatial resolution [29]. This enhancement has notably expanded global access to high-resolution data.

Maxar's Vivid data was accessed through ArcGIS Pro's base map, from which two sets of data were manually extracted for distinct purposes. The first set of imagery does not include Angola's border coordinates and is meant for training purposes. In contrast, the second set of imagery specifically includes Angola's border coordinates and is designated for testing purposes.

In an attempt to contribute to a robust generalization of the models across the training and test phases, meticulous attention was paid to ensure consistent similarity among both sets of images. This deliberate approach aims to promote dataset consistency, enhancing the model's effectiveness across both scenarios. Both sets of data contain images with the characteristics described in Table 3.1. Each image was extracted at a scale of 1:5000, with a spatial resolution of 0.5m and 3 bands (RGB). The extracted files are georeferenced, with dimensions of 5956x3134 pixels per width and height, respectively and are saved in the .tiff file format.

Characteristics	Original Raster
Scale	1:5000
Spatial Resolution	0.5m
Accuracy	5m
Source Info	Vivid
Source	Maxar
File type	.tiff
Image Compression	None
Image Size (Number of Pixels)	5956 x 3134
Image Size (Meters)	2978 x 1567
Write Geotiff tags	Yes
Color Depth	24-bit True Color
Number of bands	3
CRS	ESPG:3857

TABLE 3.1. Original Raster image's metadata.

The set of images that are intended to support the creation of the training dataset consists of 150 satellite images, covering an approximate area of 700 square kilometres within Angola's territory - which roughly represents 0.06% of Angola's total territory. The spatial coverage provided by these images aims to achieve a diverse representation of Angola's geographical features and land use patterns. For the test dataset, 29 satellite images were extracted, covering approximately 73 kilometres of Angola's land border. According to the Embassy of Angola in the United States of America [24], Angola's total land border measures approximately 4837 kilometres. Therefore, it can be inferred that the test set covers roughly 1.5% of Angola's entire land border.

3.2. Model Architectures

This Section provides a description of the model architectures used in the Dissertation. The Dissertation aims to contribute to the field of road identification through an image classification approach, using small-sized image blocks as input data. Three simple Convolutional Neural Networks (CNNs) were designed, consisting of convolutional and pooling blocks — called Custom-Built architectures. To evaluate and compare the performance of these less complex architectures, Pre-Trained architectures using ImageNet data were also tested. Therefore, this Section is organized into two main Sub-Sections: Custom-Built and Pre-Trained architectures.

3.2.1. Custom-Built Architectures

Two categories of Custom-Built architectures were created: Baseline and Dense. Starting with the Baseline architectures, the CPCP Architecture was designed as illustrated in Figure 3.1(a), consisting of two convolutional and pooling blocks. The first block, consists of a convolutional layer with 32 filters, each having a size of 3x3 pixels and ReLU activation function, followed by a batch normalization layer to normalize the output and a MaxPooling layer with a window size of 2x2. The MaxPooling layer is added to reduce

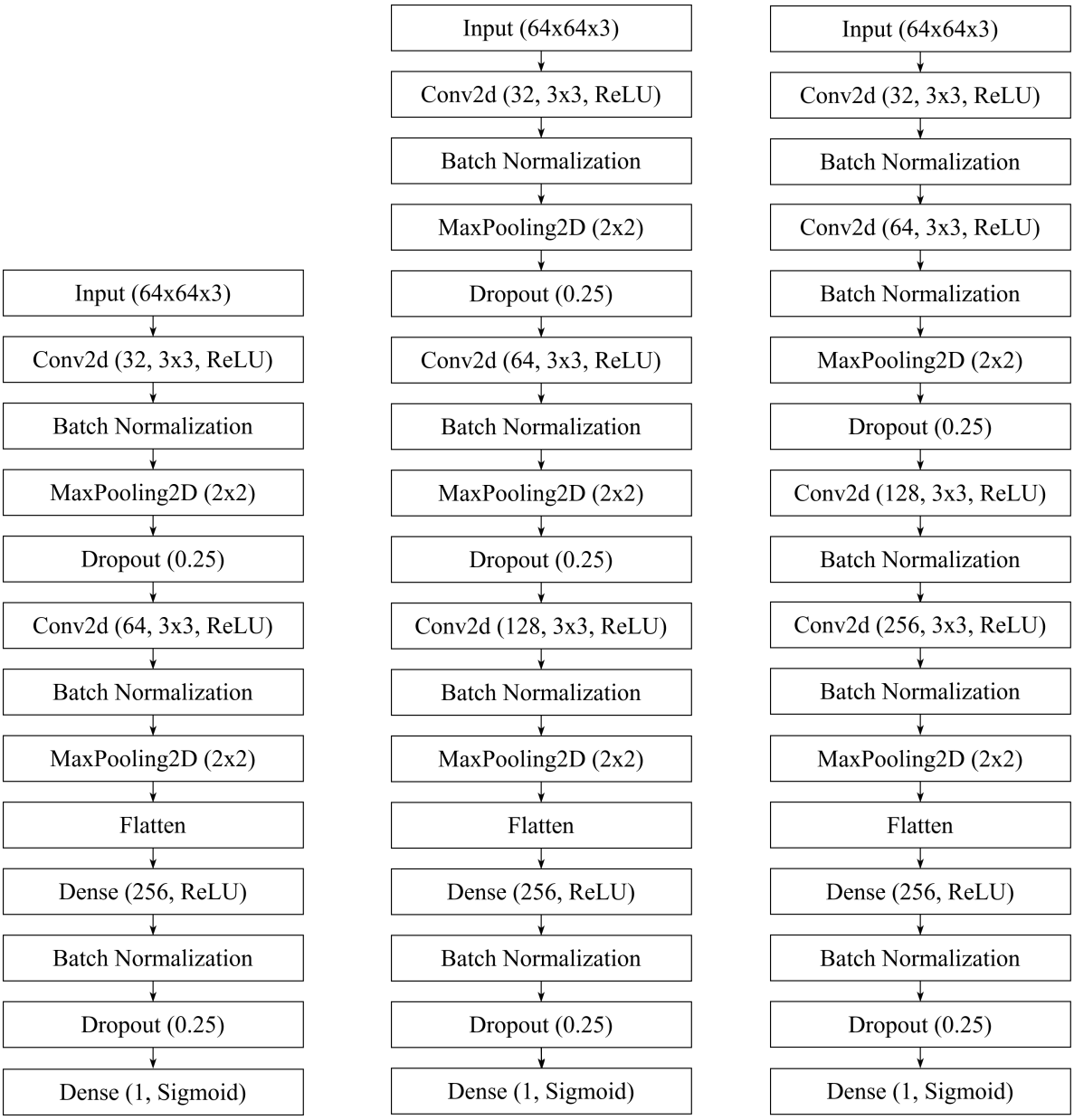
the feature map's size by half. Lastly, in order to mitigate overfitting, a Dropout layer with a dropout ratio of 25% is introduced to the block. The second block replicates the structure of the first one, except for the number of filters in the convolutional layer, which is now set to 64. After the second block, a Flatten layer is added to convert the feature map that it received from the previous max-pooling layer into a one-dimensional array. This array is then passed through a Dense Layer with 256 units and ReLU activation function, followed by another Dropout layer with a ratio of 25% to serve as a regularization technique. Lastly, the output layer consists of a Dense layer with the number of classes and a Sigmoid activation function to generate the class's probabilities.

Additionally, as shown in Figure 3.1(b), the CPCPCP Architecture consists of an extension of the CPCP Architecture by adding one extra convolutional and pooling block. The number of filters in each block increases progressively, from 32 in the first block to 64 in the second block, finishing with 128 filters in the third block. The remaining elements of the architecture remain unchanged, as detailed in the CPCP Architecture description.

Completing the baseline Custom-Built architectures experiments, the CCPCCP Architecture was created. As shown in 3.1(c), the CCPCCP Architecture consists of a variation of the CPCP Architecture, with an additional convolutional layer in each convolutional and pooling block. In the initial block, the first convolutional layer has 32 filters, and the second one has 64 filters. In the subsequent block, the first convolutional layer has 128 filters, and the second one has 256 filters. The remaining elements of the architecture remain consistent with the ones described for the CPCP Architecture.

Furthermore, in addition to the previously described architectures, a corresponding dense architecture was developed for each of them. In this iteration, a second dense block is appended to the end of the architecture to augment the architecture's complexity. By doing so, the model's capacity to capture complex patterns in the data is increased, as it has more parameters and, therefore, a higher capacity to learn features and relationships within the data. An additional dense block can also provide additional regularization benefits by introducing more dropout and batch normalization opportunities. This aids in preventing overfitting and enhancing the model's ability to generalize to unseen data.

The Dense block involves the repetition of the Dense layer (256 units, ReLU activation), Batch Normalization and Dropout (25%) combination before the final Dense layer of the model in each architecture, as it is illustrated in Figure 3.2.



(a) CPCP Architecture (b) CPCPCP Architecture (c) CCPCCP Architecture

FIGURE 3.1. Custom-Built baseline Architectures.

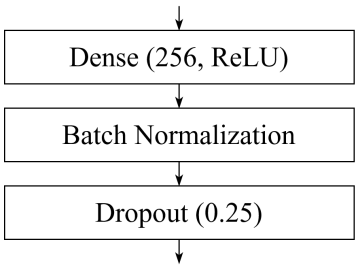


FIGURE 3.2. Custom-Built Dense Layers.

3.2.2. Pre-Trained Architectures

In the context of experimental architectures, pre-trained models were also used to explore the discrepancies between custom-built and pre-trained models. Two pre-trained architectures were selected for experiments: MobileNetV1 and ResNet50. Both models were trained using ImageNet's dataset.

The MobileNetV1 Architecture, illustrated in Figure 3.3, maintains the pre-trained layers and incorporates a GlobalAveragePooling2D layer. This addition aims to preserve semantic information by considering the entire feature map, contrasting with MaxPooling2D, which retains only the most prominent features. For this reason, it is better suited for transferring knowledge from pre-trained models to new tasks or datasets, an essential characteristic of this task. This is particularly relevant as the input files are considerably smaller than the recommended 224x224 input size. Another benefit of using GlobalAveragePooling2D is its computational efficiency compared to MaxPooling2D. Following the GlobalAveragePooling2D layer, a final Dense layer is appended, with the number of classes and a Sigmoid activation function, facilitating the generation of class probabilities.

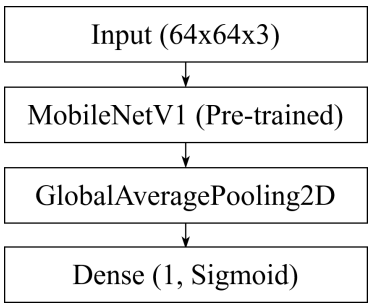


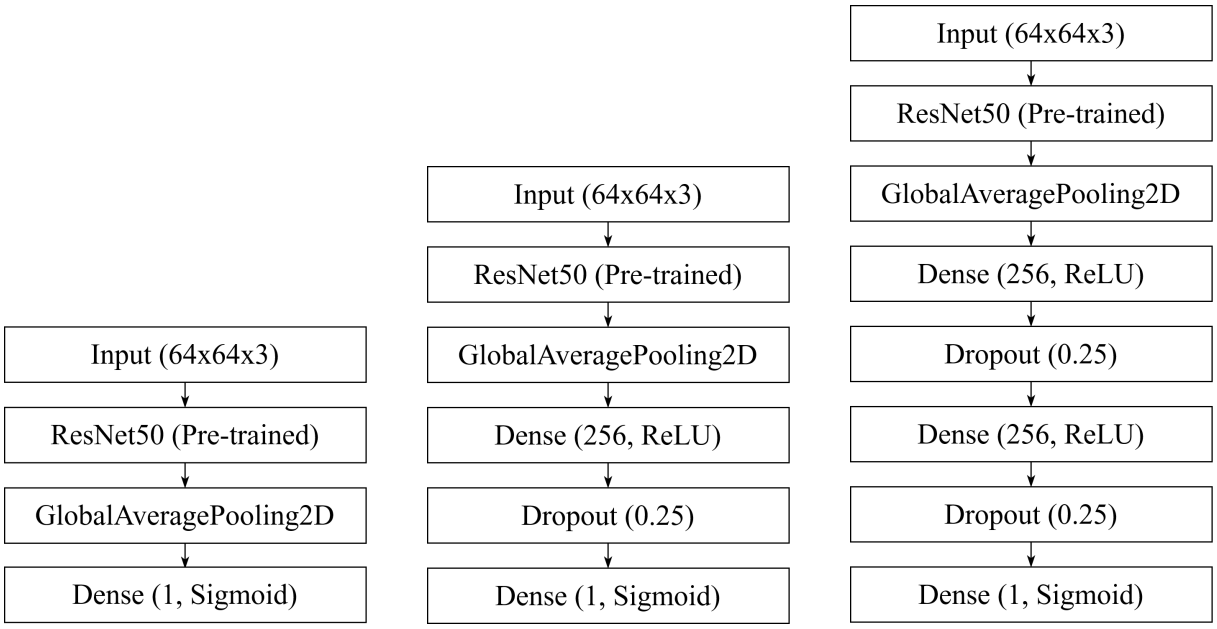
FIGURE 3.3. MobileNetV1 Architecture

The ResNet50 Architectures, illustrated in Figure 3.4, suffered similar adjustments to the ones mentioned above. As a baseline model, the pre-trained layers remained frozen, being added a GlobalAveragePooling2D layer, followed by one final dense layer with the number of classes and Sigmoid activation function - observable in Figure 3.4(a).

Following this, two additional designs were created to test more complex architectures. The ResNet50+1D Architecture, illustrated in Figure 3.4(b), consists of the ResNet50 Architecture, with the addition of a dense block composed of one dense layer with 256 units and ReLU activation function and a Dropout layer with a ratio of 25%, before the final dense layer. Subsequently, ResNet50+2D Architecture was created. As shown in Figure 3.4(c), the ResNet50+2D consists of the ResNet50+1D with two dense blocks before the final dense layer.

3.3. Evaluation Metrics

In deep learning, evaluation metrics play a crucial role in quantifying the performance of the models and determining their effectiveness in solving a given task. These metrics provide insights into various aspects of a model's performance, such as its predictive accuracy, ability to generalize to unseen data and the capacity to balance between precision and



(a) ResNet50 Architecture (b) ResNet50+1D Architecture (c) ResNet50+2D Architecture

FIGURE 3.4. ResNet50 Architectures.

recall. The loss function selected to use in this Dissertation is the Binary Cross-Entropy Loss Function that, for each data point, calculates the loss as the negative log likelihood of the true class, given the model's predicted probability. The overall loss for the dataset is often computed as the average of the losses for all individual data points (Loss Binary Cross-Entropy), expressed as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\right) \tag{3.1}$$

The accuracy measures the proportion of correctly predicted instances out of the total instances in the dataset. It serves different purposes across the training phase. In the training dataset, accuracy reflects the model's performance, indicating how well it learns from the training data. On the other hand, in the validation set, it assesses the model's ability to generalize to new, unseen data by evaluating its performance on the validation dataset. Accuracy is calculated by dividing the number of correctly predicted instances (both true positives and negatives) by the total number of instances in the dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

The Recall evaluation metric, on the other hand, measures the ability of the model to correctly identify positive instances out of all actual positive instances. It is calculated as the ratio of true positive predictions to the sum of true positives and false negatives expressed as:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

Precision quantifies the proportion of correctly predicted positive instances out of all instances predicted as positive. It is calculated as the ratio of true positive predictions to the sum of true positives and false positives, expressed as:

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

The F1-score is the harmonic mean of recall and precision, providing a balanced measure of a model's performance in terms of false positives and false negatives. It considers both false positives and false negatives, making it useful for imbalanced datasets. F1-score is expressed as follows:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.5}$$

Overall, these evaluation metrics provide a comprehensive understanding of a model's performance across different aspects, enabling researchers and practitioners to make informed decisions regarding model selection, optimization and deployment.

CHAPTER 4

Early Experiments and Methodological Insights

This Chapter presents the tasks undertaken at the beginning of the Dissertation in order to determine the optimal conditions for the Dissertation's workflow. As it is illustrated in Figure 4.1, various data preparation techniques were initially tested, particularly the establishment of the image block size. Subsequently, additional experiments were conducted to ensure that the decisions made were well-founded.

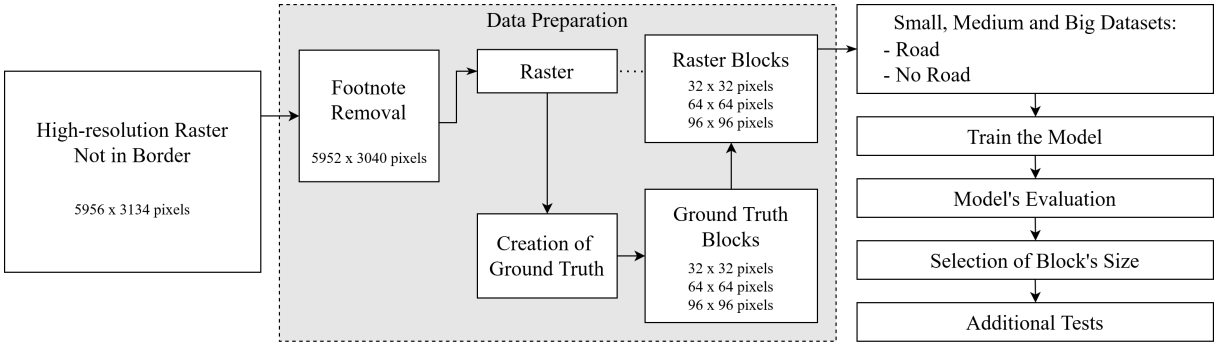


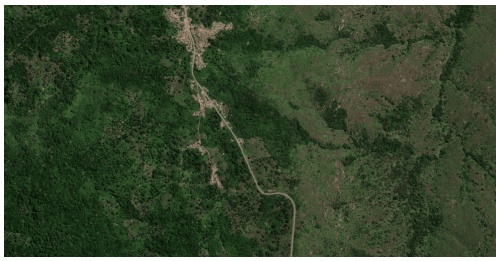
FIGURE 4.1. Early Experiments' workflow.

4.1. Data Preparation

This Section describes the techniques applied to prepare the data for the Dissertation. At first, the footnotes of the original images were removed from each image, adjusting their size to 5952x3040 pixels, while retaining all other characteristics such as georeferencing, number of bands and spatial resolution - illustrated in Figure 4.2(a), detailed in Appendix A(a).

Furthermore, acquiring ground truth data corresponding to each image is crucial. At first, experiments were attempted using fully-automated extracted ground truth data from OSM - OSMnx library. However, it soon became evident that the OSM database lacked ground truth data of adequate quality for this analysis. This inadequacy is primarily attributed to the limited amount of data available for this Dissertation, i.e., the proportion of data available in OSM is significantly smaller than that of the actual roads observable by the human eye. Therefore, the decision was made to generate ground truth data manually. As a consequence, this situation leads to the acknowledgement that research question number 1 is not truthful. In the current scenario, a fully-automated approach cannot extract reliable ground truth data to support the Dissertation.

The ground truth data was manually created for both sets of images, using an iPad, an Apple Pencil and Procreate 5.3.7 software by Savage Interactive Pty Ltd. For each Raster



(a) Example of a Raster file.

(b) Example of a Ground Truth file.

FIGURE 4.2. Example of the Prepared Data.

file, a ground truth file of the same dimensions (5952x3040 pixels) was created, featuring a transparent background with red lines delineating visible roads. Each ground truth image was then saved in .png file format with a matching filename to the correspondent raster file - illustrated in Figure 4.2(b). Afterwards, the ground truth files underwent binarization using the OpenCV library. With these steps completed, the prerequisites for the dataset creation were fulfilled.

4.2. Experiments Supporting the Dataset's Creation

4.2.1. Block's Size Establishment

As the main goal is to identify the presence of road on small image blocks, it is imperative to organize the dataset into a collection of small images, belonging to two classes - "road" or "no road", which will be subject to binary classification. Bearing that in mind, it is necessary to assess the size of the image blocks that serve as input data for the Dissertation. Considering that, three different sizes were experimented with, namely 96x96, 64x64 and 32x32 pixels - Figure 4.3.

(a) 96x96

(b) 64x64

(c) 32x32

FIGURE 4.3. Examples of the image blocks for each size.

The 96x96 pixel blocks dataset was initially created with balanced distribution between classes ("road" or "no road") - i.e. 50% of the sample belongs to each class. The criteria for labeling the blocks relates to its portion of pixels labeled as "road". If the block contains 20% of its total pixels labeled as "road", or more, then it is labeled "road" block. If it does not contain any pixel labeled as "road", then it is shuffled and only the same number of blocks labeled as "road" are stored - to assure balance in the distribution of classes. The remaining blocks without "road" pixels or the ones containing less than 20% of the total amount of pixels labeled as "road" were discarded. Afterwards, each of the selected blocks was saved in .png file format.

It is worth mentioning that the 20% threshold that was established, is opted to use as it has been considered neither a high nor too low proportion of pixels per block. Having completed the creation of the 96x96 pixel dataset - Big Sample dataset -, it comprises 5432 blocks, each class containing 2716 blocks. Afterwards, Medium Sample and Small Sample datasets were created by following the conditions detailed for the creation of Big Sample dataset, except for the number of files attributed to each class.

The three datasets were then used as input data for the mentioned Custom-Built Baseline Architectures, in order to compare the results achieved by each image size. Each experiment was trained using the Adam Optimizer as the adaptive learning rate optimization algorithm, with a fixed learning rate of 0.001. The Adam Optimizer's ability to adjust the learning rate for each parameter, individually, enables it to handle noisy or sparse gradients commonly encountered in deep learning tasks, making it a popular choice for various applications.

Binary Cross-Entropy Loss, or Log Loss, was employed as the loss function in every combination. This loss function is widely used for binary classification tasks, as it effectively measures the difference between the predicted probability distribution and the true distribution of the labels. It penalizes large deviations between the predicted and true labels, which is particularly beneficial in binary classification scenarios where the model's confidence in its predictions is crucial.

Data augmentation techniques were methodically applied to every combination, aiming to enable the learning progress of the model's comprehension of the diverse aspects of the training data. These techniques serve to enhance the models' capacity to generalize and manage variations in input data effectively. While various augmentation methods were explored, horizontal and vertical flips with a 50% probability emerged as the most effective, exhibiting the most promising outcomes. Furthermore, hyperparameter optimization was conducted by adjusting the batch size for each experiment. Different batch sizes (32, 64 and 128) were evaluated for each sample dataset and architecture, allowing for an exploration of the optimal batch size for each configuration.

Through the analysis of Table 4.1, which presents a summary of the results of the best performing experiment for each block size dataset, it is possible to conclude that medium-sized data (64x64) presents the best performing block size, selected by the F1-Score results. This indicates that the amount of information that each block yields, promotes a higher capacity for the model to learn, even when using a small portion of data.

Dataset	Model Batch Size	Loss	Accuracy	Recall	Precision	F1-Score
Big Sample	CCPCCP_64	0.2864	0.8757	0.8569	0.8941	0.8751
Medium Sample	CCPCCP_32	0.2620	0.8996	0.9167	0.8893	0.9028
Small Sample	CPCPCP_64	0.2809	0.8794	0.9094	0.8611	0.8846

TABLE 4.1. Results of the best performing Baseline models, selected by the F1-Score, for each block size.

In Appendix B.1, it is also observable that medium-sized data presents a general tendency to achieve higher values of Recall with slightly lower precision values. Looking into the loss and accuracy values, medium-sized data tends to present lower loss values and higher accuracy values than other block sizes.

Addressing the detailed values of the experiments, presented in Appendix B.1, additional conclusions can be drawn. In general, medium-sized and small-sized data tend to present higher F1-Score values as the complexity of the architecture increases. Big-sized data, tends to present higher precision values, even in less complex models, whereas medium-sized and small-sized data present a tendency for lower precision values, especially in less complex models. Medium and small-sized data also present higher recall values, even when experimented on less complex models, whereas, big-sized data tends to present lower recall values in general.

Since the F1-Score is established as the evaluation metric that supports decision-making during the investigation, the block size that is opted to pursue with the analysis is 64x64 pixels—medium-sized data.

4.2.2. Additional Dataset Creation Settings

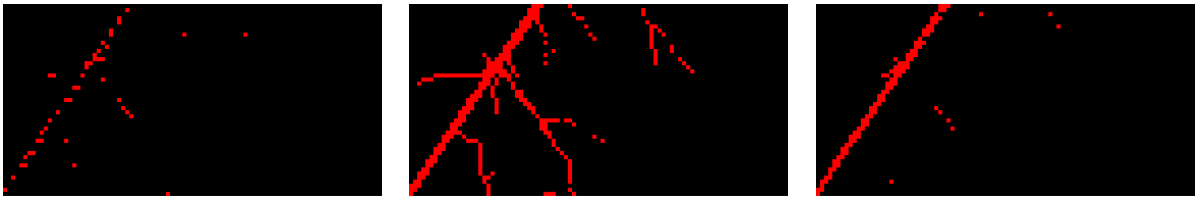
Further analysis was developed to examine the effect of larger amounts of input data on the models' evaluation metrics. For that reason, new 64x64 ("Medium 20%" dataset) and 32x32 ("Small 20%" dataset) pixel datasets were created, according to the rules described for the creation of Big Sample dataset. The Medium 20% dataset contains 14014 files and Small 20% dataset contains 96034. In the domain of additional experiments, the pre-trained baseline architectures were also introduced for further analysis.

As presented in appendix B.2, the combination that presents the highest F1-Score is the ResNet50 architecture with a batch size of 32 using Big Sample dataset (Higher F1-Score: 93.68%). However, it does not perform well in custom-built architectures, and it presents lack of detail in the sense that each block covers a larger area of non-road pixels than the actual road pixels. Since the Medium 20% dataset presents fairly high results (Higher F1-Score: 90.50%) when used as input data for the ResNet50 architecture, but also for the custom-built CPCPCP architecture (Higher F1-Score: 89.09%), it is concluded that 64x64 pixel is the block size to be used to pursue with the analysis. Adding to the results, it is also possible to examine in Figure 4.3, that medium-sized data presents a fair representation of the roads.

One final experiment was carried out, in order to conclude the early experiment's stage. It is necessary to compare the conditions that are based on the labelling attribution to the dataset's blocks. With that goal, three datasets containing medium-sized blocks were experimented on. The Medium 20% dataset, the Medium 10% and the Medium Center. Having the Medium 20% dataset already been created, the creation of the remaining two datasets will be described. The Medium 10% dataset represents a less conservative version of the initial condition, where "road" block images are selected if they contain at least 10% of the total number of pixels labeled as "road". The Medium 10% dataset consists of

44,530 blocks. Additionally, the Medium Center dataset consists on the selection through the existence of "road" pixels at the center of a block. If all four pixels at the center of a block are labeled as "road," then the entire block is classified as so. This experimental dataset consists of 14,392 blocks. Every dataset is balanced, with 50% of the blocks belonging to each class.

To enhance comprehension of how each condition affects the extraction of the "road" class, Figure 4.4 provides an illustrative example of the outputs for each condition. Under the Medium Center condition (Figure 4.4(a)), it is evident that the process tends to capture isolated road segments, resulting in an extraction pattern that deviates significantly from the actual ground truth. Contrarily, the conditions applied to create the Medium 20% and Medium 10% datasets result in the extraction of connected road segments. The significant difference between the Medium 20% and Medium 10% datasets lies in the issue identified earlier: narrow roads are often missed under the 20% condition (Figure 4.4(c)), whereas the 10% condition more consistently identifies all road segments closer to the actual ground truth (Figure 4.4(b)). Each of the aforementioned datasets was trained on two randomly selected architectures - CCPCCP and ResNet50, described in Chapter 3, Sub-Section 3.2.2.



(a) Medium Center dataset (b) Medium 10% dataset (c) Medium 20% dataset

FIGURE 4.4. Example of an image ground truth by condition applied.

When examining the general results of these experiments, in Appendix B.3, through the comparison of the F1-Score evaluation metric results, it is immediately possible to observe that the Medium Center dataset presents the lowest results, and that the Medium 20% dataset presents the highest. For that reason, it can be considered that the ideal dataset to pursue with the Dissertation is the Medium 20% - Table 4.2. However, later on the investigation, it was uncovered that having a condition that labels a block as "road" if it contains at least 20% of its pixels labeled as "road", presents a great negative impact on the results of the analysis. This occurs because, in many instances, the roads along the border are narrow, leading to a false representation of the ground truth.

This means that, in many cases, the condition attributed the label "no road" to blocks where there were visible roads. As the classifier presented the ability to correctly identify such roads, but the label criteria did not, the precision evaluation metric of the models was presenting a deep tendency to decrease in its values. Therefore, it was concluded that, even though Medium 10% dataset presents lower performance than Medium 20% dataset, it offers advantage in detail. For that reason, this less conservative approach is

preferred throughout this Dissertation, as it has the ability to identify more roads than the Medium 20% dataset - a major goal of this Dissertation.

Model Specifics		Train Set		Validation Set				
Model_BS	Data	Loss	Acc.	Loss	Acc.	Recall	Precis.	F1-Sco.
RESNET50_128	M. 20%	0.2207	0.9077	0.2312	0.9086	0.8938	0.9184	0.9059
RESNET50_32	M. 20%	0.2311	0.9042	0.2401	0.9061	0.9030	0.9042	0.9036
RESNET50_64	M. 20%	0.2213	0.9066	0.2335	0.9026	0.8713	0.9164	0.8933
CCPCCP_32	M. 20%	0.3069	0.8664	0.2972	0.8687	0.8776	0.8655	0.8715
CCPCCP_64	M. 20%	0.3217	0.8569	0.3386	0.8572	0.9290	0.8154	0.8685
CCPCCP_64	M. 10%	0.2591	0.8913	0.2670	0.8865	0.8619	0.8742	0.8680
CCPCCP_128	M. 10%	0.2850	0.8774	0.3138	0.8625	0.8563	0.8720	0.8641
RESNET50_128	M. 10%	0.3070	0.8641	0.3133	0.8649	0.8585	0.8691	0.8638
CCPCCP_128	M. 20%	0.3374	0.8533	0.3581	0.8498	0.8390	0.8614	0.8501
RESNET50_64	M. 10%	0.3729	0.8329	0.3486	0.8474	0.8547	0.8350	0.8447
RESNET50_32	M. 10%	0.3780	0.8326	0.3559	0.8463	0.8710	0.8155	0.8423
CCPCCP_32	M. 10%	0.3541	0.8444	0.3110	0.8620	0.7542	0.9124	0.8258

TABLE 4.2. Results of the training experiments' using Medium 10% and Medium 20% datasets.

Additionally, to expedite the following phases of the Dissertation, in the training process, a set of architectures were abandoned, for their poor performance in this phase. Through one final observation of Appendix B.2 and Table 4.3, it is clear that the MobileNetV1 architecture does not exhibit competitive performance values when compared to the other architectures.

Model Specifics	Train Set		Validation Set				
Model_BS	Loss	Accuracy	Loss	Accuracy	Recall	Precision	F1-Score
RESNET50_128	0.2207	0.9077	0.2312	0.9086	0.8938	0.9184	0.9059
RESNET50_32	0.2311	0.9042	0.2401	0.9061	0.9030	0.9042	0.9036
RESNET50_64	0.2213	0.9066	0.2335	0.9026	0.8713	0.9164	0.8933
MOBILENET_64	0.5188	0.7410	0.5172	0.7448	0.8143	0.7161	0.7620
MOBILENET_128	0.5152	0.7389	0.5144	0.7441	0.7623	0.7344	0.7481
MOBILENET_32	0.5206	0.7342	0.5182	0.7441	0.6934	0.7414	0.7166

TABLE 4.3. Comparison of the training experiments' results of the MobileNetV1 and ResNet50 architectures using Medium 20% dataset.

This conclusion stems from the consistently low F1-Score values exhibited by all models based on the MobileNetV1 architecture, across various block sizes, as well as the remaining evaluation metrics - observable in Table 4.3. Additionally, simpler custom-built architectures tend to produce inferior results, a trend evident when examining Table B.2. Consequently, CPCP-based architectures were also excluded from consideration as candidate architectures to pursue with the investigation.

CHAPTER 5

Training the Models

This Section provides a comprehensive overview of the training phase of the Dissertation. It begins with a detailed explanation of the processes conducted during this phase, justifying the decisions adopted and their intended outcomes. Following this, the results are outlined and analyzed, with the ultimate goal of identifying the top four best-performing models to proceed to the testing phase.

5.1. Methodology

In this section, an overview of the training phase process will be provided, summarizing the aspects outlined in the Dissertation that have led to this stage. As it is observable in Figure 5.1, the training phase is organized into five main stages: Data, Data Preparation, Dataset, Training the Model and Selection of the Top 4 Best Models.

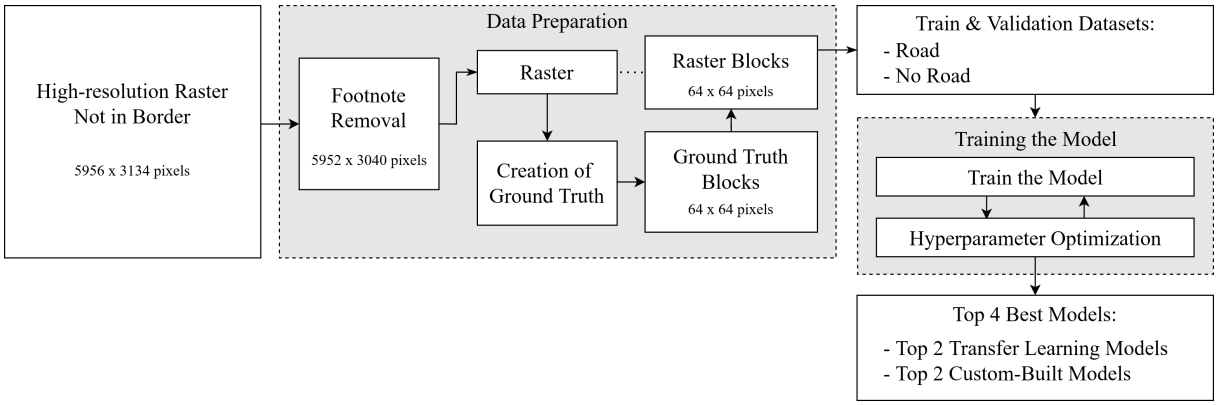


FIGURE 5.1. Workflow of the Training Phase.

Accordingly, the first stage consists of the input data of the investigation, whose characteristics have been detailed in Section 3.1. The input data set for the training phase is composed of 150 raster images (5956x3134 pixels). As presented in Figure 5.1, the data used across the training phase of the Dissertation contains solely images that do not cover Angola's border coordinates.

During the Data Preparation stage, the input images underwent processing to generate a training dataset. Initially, the footnotes were removed from each image, resulting in images with dimensions of 5952x3040 pixels. In Figure 5.1, this processed output is termed a "Raster".

Following this, the ground truth for each Raster was manually produced, featuring a transparent background with red lines outlining visible roads. These ground truth files then underwent pixel-wise binarization, where the presence of a "road" was assigned a

value of 1, and the absence of roads ("no road") was assigned a value of 0. This step is illustrated in Figure 5.1 as "Creation of Ground Truth".

Subsequently, the dataset was created. The ground truth files were partitioned into 64x64 pixel segments and each segment was labeled based on the proportion of "road" pixels it contained. If a segment contained more than 10% of its pixels labeled as "road", it was labeled as "road"; if it contained less than 10%, it was discarded. Segments without any "road" pixels were shuffled, and an equal number of "road" and "no road" segments were retained. The remaining blocks without roads were also discharged. As a result, the raster files were partitioned and stored in accordance with the organization of the ground truth blocks.

After the data preparation stage, the dataset is now ready to be employed as input data for the model experiments. It comprises 52822 input blocks, evenly distributed between the two classes. Subsequently, the dataset is randomly partitioned into a training and validation set, with proportions of 80% and 20%, respectively.

Until this stage every task was performed locally, however, in order to optimize the performance of Deep Learning tasks, particularly during the resource-intensive training phase, a GPU was used to expedite the process [2]. To facilitate this, the dataset was securely stored in an Amazon Web Services (AWS), the Amazon Simple Storage Service (S3) bucket. This decision was motivated by AWS infrastructure's scalability, reliability, and high-performance capabilities. The robust security measures implemented by AWS ensured data protection throughout the entire process, guaranteeing confidentiality and integrity. Following this, an ml.g4dn.xlarge notebook instance was created on Amazon SageMaker to continue with the process. This notebook instance provides access to a GPU with NVIDIA T4 Tensor Core architecture and 16.0 GiB of memory, ensuring that all necessary conditions for the analysis are met.

In the training stage, each architecture underwent experimentation using the dataset. This process involved employing each of the architectures mentioned in Section 3.2 - except for the CPCP-based and MobileNetV1 architectures - along with hyperparameter optimization of batch sizes. Following the specifications outlined in Section 4.2.1, the models were trained using three different batch sizes (32, 64 and 128), utilizing Adam Optimizer with a fixed learning rate of 0.001, Binary Cross-Entropy Loss and employing data augmentation techniques.

In the final stage of the training phase, attention is directed towards analysing the results that the models have experimented with. The primary goal of this stage is to pinpoint and designate the two most promising models from both the custom-built and pre-trained categories. This selection aims to compare the performance and behavior of models across the different categories during the subsequent testing phase, enabling a comprehensive assessment of their capabilities.

5.2. Results

Despite the fact that the top-4 best performing models are derived from custom-built architectures, it has been decided to select the two best custom-built approaches along with the two best pre-trained approaches - results displayed in Table 5.1. This decision aims to facilitate a comparison between the performance of both approaches when handling the test data.

The CPCPCP+1D model, employing batches of size 32 for each iteration, emerges as the top performer of all models. Precision and Recall percentages of 89.51 and 86.72 exhibit balanced performance between both evaluation performance metrics. This behaviour indicates the model's proficiency in accurately recognizing road instances, while maintaining a relatively low false positive rate (high Precision), and effectively capturing most road instances in the dataset (high Recall). The precision score indicates that 89.51% of the time, the model accurately identifies "road" blocks, and the Recall score indicates that the model detects approximately 86.72% of all "road" instances in the dataset. Following a comparable trend, the second best-performing model is based on the custom-built CCPCCP architecture, with a batch size of 64. The model achieves an F1-Score value of 0.8766, a Precision value of 0.88, and a Recall value of 0.8731.

Model Specifics	Train Set		Validation Set				
Model_Batch Size	Loss	Accuracy	Loss	Accuracy	Recall	Precision	F1-Score
CPCPCP+1D_32	0.2956	0.8743	0.2724	0.8858	0.8672	0.8951	0.8809
CCPCCP_64	0.3009	0.8692	0.2876	0.8794	0.8731	0.8800	0.8766
RESNET50_128	0.3070	0.8641	0.3133	0.8649	0.8585	0.8691	0.8638
RESNET50+2D_32	0.3269	0.8562	0.3149	0.8634	0.8681	0.8570	0.8625

TABLE 5.1. Results of the 4 best performing models, selected by the F1-Score, during the training phase.

Regarding the selection of pre-trained models, although they tend to yield slightly lower overall results, the disparity between their performances is insignificant. The Resnet50 architecture with a batch size of 128 per iteration emerges as the top pre-trained model, with an F1-Score of 0.8638 and balanced Recall and Precision values of 0.8585 and 0.8691, respectively. Similarly, the second-best pre-trained model achieved balanced performance, with an F1-Score value of 0.8625, using the ResNet50+2D architecture with a batch size of 32.

Through the examination of Appendix C, where the results of the training phase are detailed, it is noticeable that the models that present a tendency for lower F1-Score values exhibit higher asymmetry between Recall and Precision scores. This behaviour implies that those models struggle to balance minimizing false positives and maximizing true positives, which affects their overall performance, particularly reflected in the F1-Score. The models exhibiting this behavior tend to have the highest precision scores and the lowest recall scores, which suggests that these models are more conservative in their positive predictions, prioritizing the reduction of false positive classifications even if it

means that many positive instances are also being missed. Ultimately, this results in a low true positive rate.

Contrarily, the models presenting higher F1-Score values typically do not have the highest Recall nor Precision values individually. Instead, they demonstrate the most balanced results across both metrics. This balance implies that these models effectively minimize both false positives and false negatives, achieving a harmonious compromise between Precision and Recall (Appendix C).

Contributing to a comprehensive overview of the training results, the analysis unveils distinct trends in the performance of pre-trained and custom-built models. Pre-trained models consistently demonstrate F1-Scores within a narrow range - F1-Scores ranging from 84.23% to 86.38% -, indicating minimal influence from experimented hyperparameter optimization and complexity augmentation techniques. In contrast, custom-built architectures present broader variations in F1-Scores - with F1-Scores ranging from 79.50% to 88.09% -, suggesting heightened sensitivity to these factors. These insights emphasize the critical importance of meticulously selecting model types and optimization strategies tailored to the specific attributes of the dataset and task at hand.

Based on the considerations outlined above, the models selected for the test phase are as follows: 1) CPCPCP+1D model with batch size 32 per iteration; 2) CCPCCP model with batch size 64; 3) Resnet50 model with batch size 128; and 4) Resnet50+2D with batch size 32.

CHAPTER 6

Testing the Models

In this Chapter, the details of the investigation of the test phase will be provided. Initially, an exposition of the processes that were carried out in this phase will be presented, accompanied by a description of the rationale behind the decisions made and their underlying intent. Subsequently, the results will be presented and interpreted, aiming ultimately to identify the strengths and weaknesses of the Dissertation. Finally, in Subsection 6.3 an overview of the results is provided, supporting the conclusions drawn from the Dissertation's findings.

6.1. Methodology

Similarly to the training phase, as it is illustrated in Figure 6.1, the test phase is organized into eight stages: Data, Data Preparation, Testing the Model, Performance Evaluation, Creation of Color-Coded Images, Post-Processment, Point of Entry Identification and creation of the Point of Entry Map.

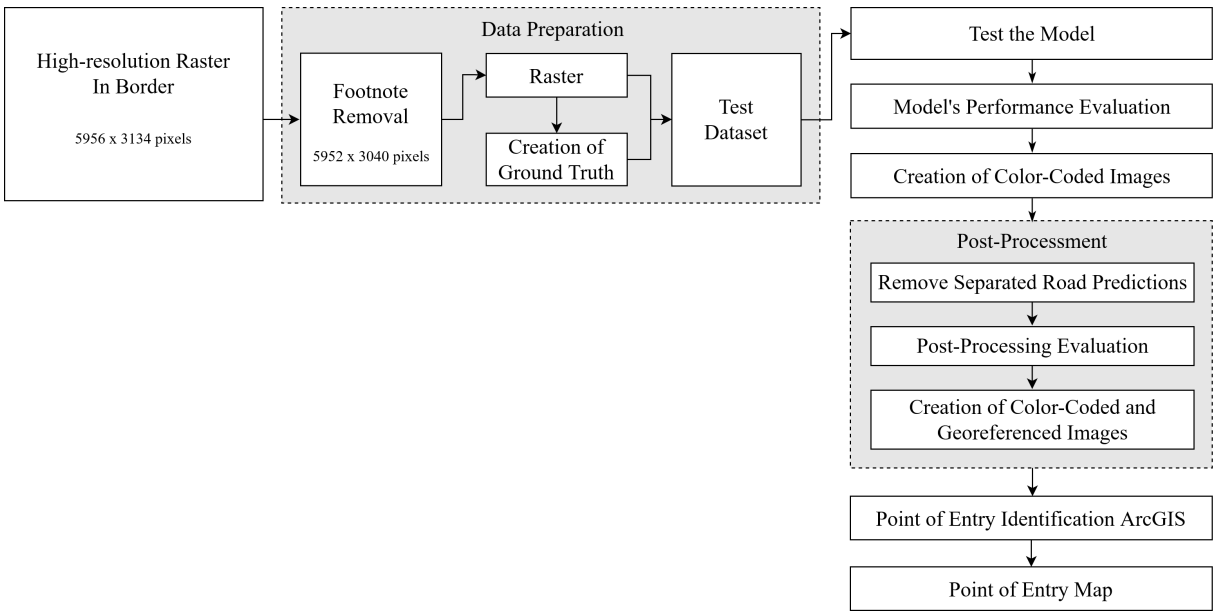


FIGURE 6.1. Workflow of the Testing Phase.

During the test phase, new and unseen data containing solely imagery from the border area is used. As it is presented in Figure 6.1, the data that is being tested in this phase underwent similar pre-processing techniques to the train data, except for the storage process and the attribution of blocks that contain less than 10% of their blocks labeled as "road", which were stored in the "no road" class. Instead of saving block images in a local directory, the data was stored in an array. After the completion of the described

process, the test dataset array comprises a total of 126759 samples, with 9026 samples labeled as “road”, and 117733 samples labeled as “no road”.

Subsequently, the dataset was tested on the selected models, and their performance was evaluated based on various metrics, such as accuracy, recall, precision, and F1 score. These metrics provide valuable insights into the effectiveness of the models in accurately classifying roads within the dataset.

As output, a color-code system was created to recreate the original images, according to the confusion matrix schema. Therefore, for each 64x64 pixel block, a classification is attributed. The color-code dictates the output as the following information:

- True Positive Predictions: Green
- True Negative Predictions: Blue
- False Positive Predictions: Yellow
- False Negative Predictions: Red

Upon analyzing the color-coded outputs, the 8-adjacency post-processing technique was applied to remove isolated positive predictions i.e. outliers. This approach consists of applying a condition that states that if the block is predicted to have a road and none of the 8 neighbouring blocks also presents a positive prediction, then the block is post-processed to not have a road.

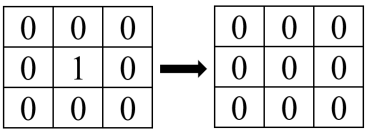
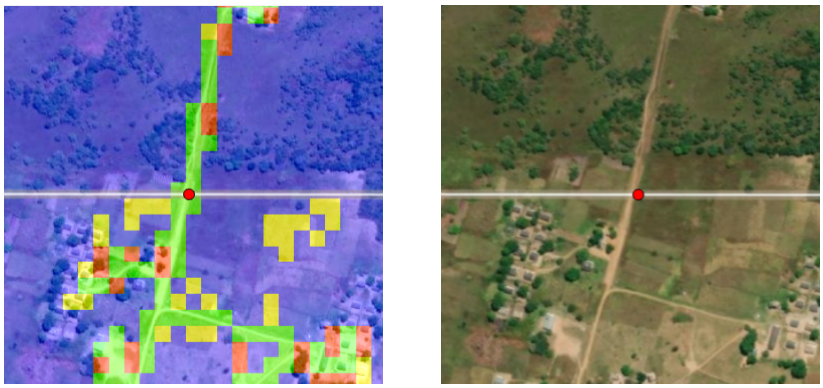


FIGURE 6.2. Post-Processing Technique Schema: 8-Adjacency.

The post-processed, color-coded images were then combined with the original images, georeferenced and saved. The output was then uploaded into ArcGIS Pro to provide visual assistance for identifying new Points of Entry by land in Angola.



(a) Example of a Point of Entry Situation - Color-Coded image.

(b) Example of a Point of Entry Situation - WorldView-3.

FIGURE 6.3. ArcGIS Pro: Point of Entry identification.

In Figure 6.3, a demonstration of how the Points of Entry were integrated into ArcGIS is portrayed. Whenever a True Positive situation (denoted by a green square) intersects the land border-line, a Point of Entry is introduced - marked by a red dot -, as illustrated in Figure 6.3(a). In Figure 6.3(b), a representation of the ArcGIS Pro's base map with the land border-line and Point of Entry marked is presented. Notably, in Figure 6.3(b), a road intersects the border-line, indicating the necessity of introducing a new Point of Entry into the database. For each test set image, this process was meticulously executed.

6.2. Results

This Section is organized into three Subsections. First, the test phase results are presented, along with an examination of the identified issues that support the necessity for subsequent post-processing techniques. Accordingly, the second Subsection describes the results obtained after applying the 8-Adjacency post-processing technique.

6.2.1. Original Results

This Subsection introduces an overview of the results obtained from the test phase, aimed at evaluating the comparative ability of the selected models to classify road blocks accurately.

Model_Batch Size	Accuracy	Recall	Precision	F1-Score
CCPCCP_64	0.9394	0.7043	0.5592	0.6234
CPCPCP+1D_32	0.9073	0.7887	0.4198	0.5480
RESNET50+2D_32	0.9024	0.7642	0.4023	0.5271
RESNET50_128	0.8912	0.7769	0.3733	0.5042

TABLE 6.1. Original test results.

Examining Table 6.1, the tendency for the approaches to achieve consistently lower F1-Score values is immediately observed, compared to the previously achieved results during the training phase. Across all models, a noticeable decrease of approximately 0.2 in this performance metric is observed, indicating a significant behavioral shift. This behaviour is primarily due to a decrease in both Recall and Precision evaluation metrics. While a decrease of around 0.1 values is observed in the Recall results, indicating that each model is detecting approximately 10% fewer "road" instances in the test phase compared to the training phase, the decrease in Precision values is more pronounced, with decreases of approximately 0.4 values observed. This means that while in the training phase, models were able to accurately identify "road" image segments around 90% of the time, in the test phase, the models accurately identified roads in about half the time.

The extreme discrepancy between Recall and Precision evaluation metrics contributes to a substantial decrease in the F1-Scores. On the contrary, an opposite trend is observed within the Accuracy results, where an increase in values is noted across every model. This suggests that the models may not be generalizing well to new, unseen data.

When comparing the F1-Score results of each model, aside from the drop in its values compared to the training phase, it is noticeable that the best-performing model is

CCPCCP using a batch size of 64. In the test phase, this model achieved an F1-Score of 0.6234, Accuracy of 0.9394, Recall of 0.7043 and Precision of 0.5592. It is noteworthy that the remaining models achieved considerably higher results within the Recall evaluation metric. However, due to higher decreases in Precision values, their F1-Score results are lower. These results in Precision indicate that the models are too liberal in attributing positives, presenting a high tendency to predict false positives.

The observed variations in evaluation metrics between the training and test results raise concerns about the model's generalisation ability. Specifically, while the Accuracy values improved, other metrics experienced a decline. This discrepancy initially suggests a potential issue with overfitting, where the model performs well on the training data but fails to generalize to unseen data. However, upon further examination, it becomes apparent that the test dataset may not fully represent the diversity in the training dataset.

Despite the efforts to ensure dataset representativeness, retrospective analysis reveals shortcomings in this aspect. The failure to adequately capture the variability and complexity of real-world scenarios within the test dataset has led to discrepancies in model performance between the training and testing phases. As a result, the model may exhibit a biased performance evaluation, with inflated Accuracy metrics masking deficiencies in other crucial performance indicators. Addressing this discrepancy necessitates reevaluating the dataset curation process, focusing on enhancing diversity and inclusivity to better reflect the conditions presented within the border area.

An overview of Table 6.1 shows an evident trend for Custom-Built approaches to outperform the Pre-Trained approaches. At the same time, each type of approach, has presented a change in order. Individually, the best-performing Custom-Built approach in the training phase, CPCPCP+1D (batch size: 32), was outperformed by the second-best, CCPCCP (batch size: 64) model, in the test phase. Similarly, the ResNet50 (batch size: 128) model was outperformed by the ResNet50+2D (batch size: 32) model in the test phase.

Recalling the color-code introduced in Section 6.1, to support the visual analysis of the model's outputs, illustrated in Figure 6.4, it becomes evident that a significant number of false positive predictions (represented as yellow) are predicted. This corroborates the observed Precision results, adding to the fact that the models are falsely predicting more positive classifications than what was positively classified in the ground truth.

Moreover, it is observable that false positive predictions tend to present several isolated cases that lack connection to other "road" elements, which directly impacts the Precision and consecutively F1-Score results of the test. Furthermore, it is evident that false positive predictions often manifest as isolated cases with no connection to other "road" elements. This directly influences the Precision and, consequently, the F1-Score results of the test.

6.2.2. Post-Processing Results

In an attempt to address the prevalence of false positive predictions observed in the testing of the models, and the consequential impact on Precision and F1-Score results, an

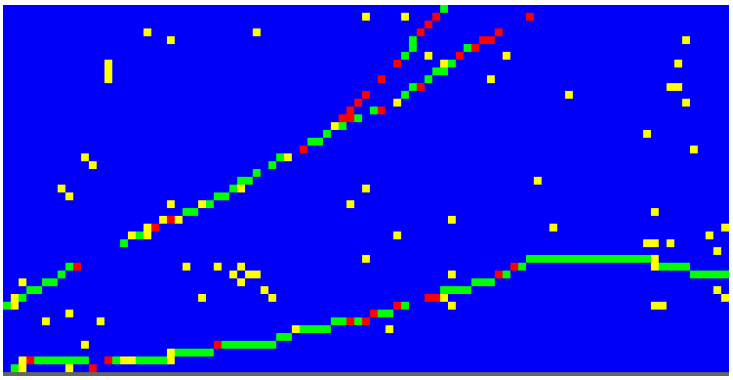


FIGURE 6.4. Example of a color-coded image Output of the Original Results (CCPCCP; BS:64).

8-Adjacency post-processing technique was applied to the test outputs. The 8-Adjacency process is detailed in section 6.1.

Observing Table 6.2, it is immediately possible to conclude that the overall F1-Score results improved. By reclassifying isolated positive predictions as negative predictions, the 8-Adjacency technique aims to improve Precision results by minimizing false positive predictions. The Accuracy results also presented improvements.

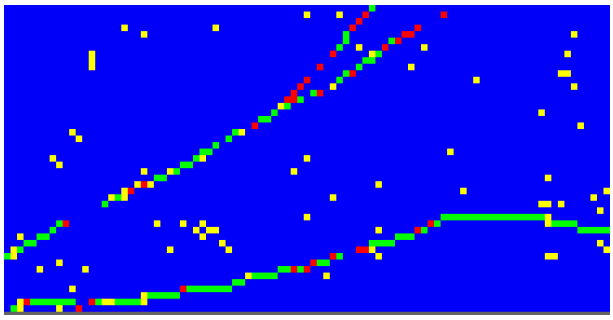
Model_Batch Size	Accuracy	Recall	Precision	F1-Score
CCPCCP_64	0.9446	0.6726	0.5985	0.6334
CPCPCP+1D_32	0.9179	0.7701	0.4548	0.5719
RESNET50+2D_32	0.9120	0.7424	0.4314	0.5457
RESNET50_128	0.9014	0.7587	0.3988	0.5228

TABLE 6.2. Post-Processing Results.

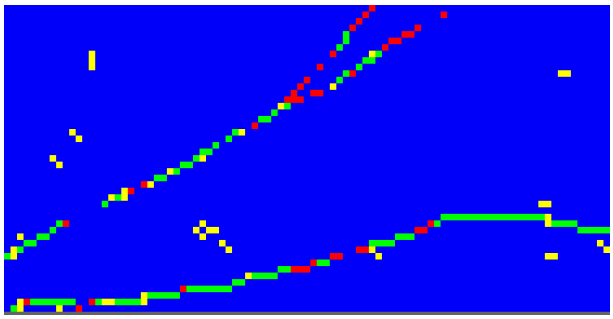
However, refining false positive predictions comes at a cost, primarily impacting Recall results. When positive predictions are reclassified as negative based on their isolation, some true positive instances may also be affected, resulting in a reduction in Recall. This happens because true positive instances that are isolated in the data may also be mistakenly reclassified as negative due to the adjacency criterion.

Analyzing the color-coded post-processing output example, illustrated in Figure 6.5, it can be inferred that most isolated false positive instances have been eliminated. As a result, the visualization of the model's predictions was simplified, allowing the main extracted features to be more easily observed by eliminating noise from the figure outputs.

Implementing the 8-Adjacency post-processing technique not only enhanced the overall evaluation results but also elevated the observation of the outputs by reducing the noise in the image outputs.



(a) Original Output.



(b) Post-Processing Output.

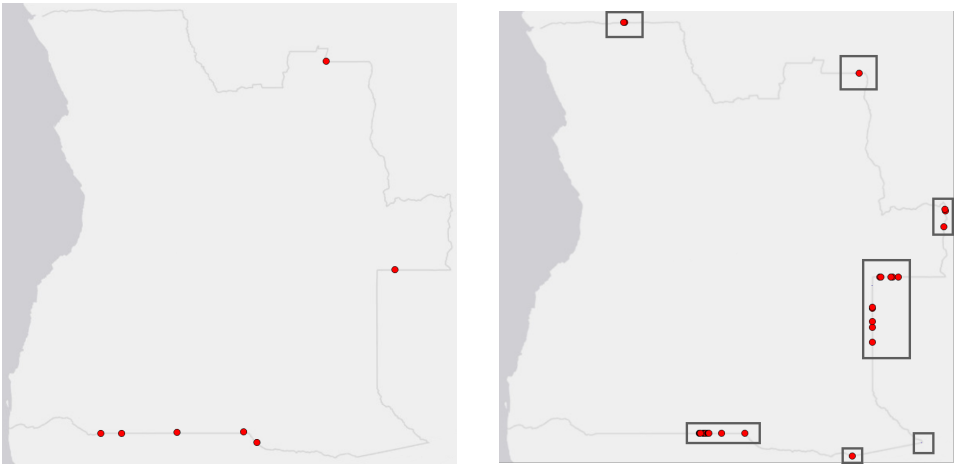
FIGURE 6.5. Comparative example of a color-coded image Output (CCPCCP; BS:64).

6.2.2.1. *Point of Entry Map*

In a final effort, the Post-Processed images of the best-performing model (CCPCCP (Batch Size: 64)) were introduced into an ArcGIS Pro base map, to manually identify new Points of Entry throughout Angola's land boundaries. For visualization purposes, two Point of Entry Maps were created: the first containing the IOM's Point of Entry data (Figure 6.6(a)) and the Dissertation's Point of Entry data (Figure 6.6(b)).

The IOM's Displacement Tracking Matrix Point of Entry Map consists in a replica of the Point of Entry data [10], and was created aiming at enhancing the comparative experience for the reader. The Dissertation's Point of Entry Map is composed of the intersections between the identified roads and Angola's land border, predicted by the CCPCCP (Batch Size: 64) model - as shown in Figure 6.6(b).

Upon the examination of the Point of Entry Maps, it becomes evident that even though the approach employed in the Dissertation contains images within the boundaries of the rectangles presented in the image, it harvests a significantly higher number of Points of Entry compared to the IOM's initiative. Furthermore, considering that the Dissertation's approach is limited to analyzing only 1.5% of Angola's land border, the results achieved are particularly promising. Despite this constrained dataset, the investigation successfully identifies 47 Points of Entry, elaborated in Appendix D.



(a) IOM's identified Points of Entry, by land.

(b) Predicted Points of Entry, by land.

FIGURE 6.6. Comparison between the Points of Entry identified in the DTM's Report and the Points of Entry identified in the investigation.

6.3. Discussion

By identifying patterns and challenges that emerged during the Dissertation, this subsection presents a discussion and interpretation of the Dissertation's findings. Starting with an examination of the ground truth data, it is possible to identify constraints related to mislabeling. This issue arises when the line is too narrow or passes through a small portion of the block, leading to failure to adequately represent the true representation of the road - i.e., resulting in mislabeled blocks. An example of this issue can be observed in Appendix E.3, examining false positive instances. This case reveals instances where the ground truth is outperformed by the predictive model, presenting situations where the ground truth fails to accurately capture the presence of roads, ultimately leading to falsely incorrect model predictions.

To handle this problem, it is imperative to refine the ground truth creation process to ensure that road boundaries are strongly delineated. This issue may involve revisiting the criteria used to define road segments and implementing stricter guidelines for ground truth annotation. Altering the condition for selecting "road" blocks through the minimization of the percentage of pixels classified as "road" when attributing class "road" to a block - from 10% to 5%, or even the existence of "road" pixels in the image - should be reconsidered. There could also be experimented a semi-automatic labelling approach [34][5][18], where a set of the data is manually labeled and another larger set is automatically labeled and trained.

The prediction of road networks in small villages consists of another constraint that has been identified. Areas where the organization of houses and roads lacks a discernible pattern consists in additional challenges for the predictive model. Examining Appendix E.2, it is possible to observe inconsistencies in road detection in small villages. While the main roads are correctly identified by the model, smaller roads exhibit varying degrees of

detection. Some of these are correctly identified, while others are missed. The difficulty in accurately detecting road networks in small villages can be attributed to the lack of clear patterns in the arrangement of houses and roads. Unlike urban or suburban areas where roads follow distinct layouts, villages often feature irregular road networks that may not conform to conventional patterns. As a result, the model may struggle to generalize and learn the features of these road networks effectively.

To mitigate this concern, an increase in the amount of data may contribute to a solution, but also the addition of precision in the labeling process, tailored to small villages, may be helpful. This approach would involve meticulous annotation of road segments within village areas, considering the unique characteristics and configurations of road networks in such settings. Additionally, incorporating contextual information such as land use patterns, building densities and geographical features surrounding villages may enhance the model's ability to distinguish between roads and other features within these areas [30][17]. By integrating such contextual cues into the training process, the model can better differentiate between road segments and accurately identify road networks within small village settings.

Another issue that relates to what was described above concerns the misclassification of agricultural areas as "road" by the model. This misclassification occurs when the model incorrectly identifies certain land features, such as agricultural fields, as "road" segments. Appendix E.1 provides examples of this misclassification, where the model erroneously labels blocks of land that exhibit characteristics commonly associated with agricultural areas as roads. One common pattern observed in these misclassified areas is the presence of clear boundaries or delineations, often indicative of agricultural fields. These boundaries may manifest as distinct lines or demarcations separating different parcels of land, such as crop fields or pastures. The misclassification of agricultural areas as roads can be attributed to several factors, including the similarity in visual features between agricultural fields and roads, as well as the complexity of land cover and land use patterns in satellite imagery.

It may be necessary to refine the model's training data and augment its learning capabilities to better differentiate between agricultural areas and roads, to address this issue. This could involve incorporating additional training samples that represent a diverse range of agricultural landscapes and land cover types, allowing the model to learn the distinguishing features of each class more effectively. Furthermore, leveraging contextual information such as geographic metadata, seasonal variations in land cover and crop rotation patterns may aid in improving the model's ability to discriminate between roads and agricultural areas [17][30]. By integrating such contextual cues into the training process, the model can develop a more nuanced understanding of the spatial characteristics and visual cues associated with different land features, reducing the incidence of misclassifications.

Indeed, while some false positive predictions may indicate misclassifications, it is important to recognize that not all instances are necessarily erroneous. In fact, the model has demonstrated an ability to identify "road" situations where there appears to be some form of passage or connectivity between agricultural areas or fields. This observation highlights the model's capability to discern subtle spatial patterns and recognize features that may not be immediately obvious to the human eye. In many cases, what may initially appear to be a misclassification, can actually represent legitimate pathways or access routes that facilitate movement between agricultural plots or fields. Rather than viewing these predictions as false positives, they should be regarded as valuable insights into the nuanced interactions between land use patterns and transportation infrastructure.

Another situation where the model exhibits shortcomings is its prediction of blocks that cover water bodies. In the example provided in Appendix E.3, a river located on the left side of the image was incorrectly classified as "road" by the model. This misclassification can be attributed to several factors, one of which is the limited detail available in the imagery used for the task. The imagery that has been used consists of RGB bands, which may not adequately capture the distinctive spectral characteristics of water bodies. As a result, the model may struggle to differentiate between roads and water bodies, particularly in areas where they exhibit similar visual patterns or features as roads.

One potential solution to address this issue is to leverage multispectral imagery. Unlike RGB imagery, multispectral imagery captures information across hundreds of narrow and contiguous spectral bands, allowing for more precise characterization of surface materials and features, including water bodies. Multispectral imagery is particularly effective in capturing the unique spectral signatures associated with water, such as its high reflectance in the near-infrared region and distinctive absorption features in the visible and near-infrared spectra. By incorporating multispectral data into the training and prediction process, the model can improve its ability to accurately distinguish between roads and water bodies, reducing the likelihood of misclassifications [30][8][15][17].

In an effort to tackle the misclassification instances outlined in the previous paragraphs, adopting a multi-class classification approach [12][8][35][31] could also offer a viable solution. The challenges associated with classifying roads, small villages, agricultural areas and water bodies share similarities, making it difficult for the model to correctly distinguish between them - especially using a small dataset. Training the model to recognize not just roads but also various types of land use and land cover gives it a more comprehensive understanding of the landscape. Another potential approach to contribute to the mitigation of misclassification instances is to explore the use of multi-spatial resolution data in the analysis. While the Dissertation focuses on a single spatial resolution size (0.5m), a multi-spacial resolution approach should be experimented. Leveraging data with varying spatial resolutions - such as the 0.3m spatial resolution data available in ArcGIS Pro. The integration of multi-resolution data into the analysis can help capture

finer details and nuances, improving the model's ability to accurately detect and classify features across diverse environments [33][5][8][15][31].

The last constraint that is identified relates to the block's size. For an instance, the use of blocks with a dimension of 64x64 pixels is adequate for analyzing situations as thin road segments, however, it fails to cover large road segments. These issues arise when distinguishing between large roads and other features, such as agricultural areas. The model struggles to effectively learn the boundaries of large roads, relying solely on color information, which erroneously classifies agricultural areas as roads. Appendix E.1 provides examples of this limitation, with instances of false positives occurring in agricultural areas that exhibit patterns similar to large roads. Conversely, the 64x64 pixels block can also be too large for the information it is labeled as. In situations where roads are very thin, representing a small portion of the total pixels of the block, the model may erroneously learn its characteristics as "road", erroneously classifying an entire area as a road. This can lead to false predictions of non-road characteristics as road features.

To handle this problem, experimenting with different block sizes and incorporating techniques such as adaptive sliding window analysis may help improve the model's ability to identify and classify roads of varying sizes and characteristics accurately. Refining the training process and optimizing the model architecture can mitigate the impact of this limitation and enhance the accuracy and reliability of road detection in satellite imagery.

CHAPTER 7

Conclusions and recommendations

Concluding this Dissertation, it is possible to state that a binary image classification approach can effectively and accurately identify new PoE on the territory of Angola. By predicting roads intersecting the country's border lines, this Dissertation introduces a new approach to identifying PoE, by leveraging image classification techniques, contributing to the advancement of road detection algorithms in satellite imagery analysis. This confirms research question number 3, where it is proposed that an image classification-based approach can identify relevant nodes in complex networks of land roads and borders with accuracy.

As the Dissertation has revealed a significantly higher number of identified PoE, comparing to the IOM's COVID-19 Impact on Points of Entry program, research question number 4 is verified. Even though the Dissertation analyzes data corresponding to a small fraction of Angola's land border that does not cover the area of the identified PoEs by the IOM's program, it has successfully identified 47 PoE, conversely to the 7 PoE identified by the IOM's program. This improvement also proves the Dissertation's capacity to outperform the OSM database, as it is the data source of the IOM's COVID-19 Impact on PoEs program.

Even though the Dissertation confirmed the validity of the above-discussed research questions (3 and 4), it also uncovered unexpected challenges, ultimately contributing to the non-verification of research questions 1 and 2. As has been discussed, fully automated methods for ground truth extraction (OSM) have been proven to be inadequate for the Dissertation due to their insufficient detail and completeness, contradicting research question number 1. Moreover, as the most effective approach for this task is the CCPCCP model using a batch size of 64, research question number 2 is also contradicted, as it states that Pre-Trained models can present better performance in the extraction of roads from satellite imagery, than custom-built models.

Moving forward, it is crucial to highlight the challenges uncovered by the Dissertation, which should be considered in future studies in the field. The observation of a general tendency for the test results to decrease in F1-Score, Recall and Precision scores while improving the Accuracy results when compared to the training phase, presents cause for concern. This behavioral shift implies potential issues with the model's generalization to unseen data as a consequence of disparities between the training and the testing data. Addressing this issue is crucial for improving the overall reliability and generalization of the model. The addition of new data that covers a wider range of scenarios can be one approach to mitigate this behavioural shift.

Even though post-processing techniques have been employed, concerns persist regarding the model's performance related to the prevalence of false positive predictions. 8-Adjacency post-processing technique has shown better F1-Score and Precision results at the expense of decreasing Recall scores. Additional exploration of post-processing techniques to reduce false positive instances while minimizing the impact on Recall can enhance the overall accuracy and reliability of model predictions, which could have a valuable impact on future work.

This Dissertation has also unveiled several challenges related to overall misclassification occurrences. Focusing on the moderation of such situations, several additional approaches can be explored, including multi-class classification, multi-spatial resolution, semi-automatic labelling, the use of multi-spectral imagery, integrating contextual data, and the use of adaptive sliding windows.

By adopting a multi-class classification approach to the problem [12][8][35][31], the models can learn to differentiate between various land cover classes beyond roads and non-roads. Additional classes could include vegetation, water bodies, buildings or agricultural areas, allowing the models to capture a more nuanced understanding of the spatial characteristics and spectral signatures of different land cover types. This approach could mitigate confusion between similar land cover types, such as roads, agricultural areas, or water bodies.

In this dissertation, single spatial resolution data is utilized [33][5][8][15][31]. Nonetheless, additional exploration should be experimented on by integrating multi-spatial resolution data. Incorporating data with diverse spatial resolutions can enhance the robustness of the analysis by capturing additional landscape characteristics. This could improve the model's ability to accurately detect and classify road features across diverse environments.

Improving the ground truth creation process to ensure an accurate representation of road boundaries and features is essential for enhancing the reliability of training datasets and mitigating misclassifications is another aspect to consider in future research related to the topic. There should also be tested a semi-automatic approach [34][5][18], where a set of the data is manually labeled and another larger set is automatically labeled and trained.

Exploring alternative data sources, such as multi-spectral imagery, could also contribute to enhancing the reliability of this Dissertation [30][8][17][15]. By leveraging the analysis with multi-spectral imagery, it becomes possible to capture subtle spectral signatures associated with different land cover classes, facilitating more accurate discrimination between road features and other environmental elements. Unlike RGB imagery, multi-spectral data captures information across several spectral bands, providing a more comprehensive view of the electromagnetic spectrum. This rich spectral information enables more precise characterization of surface materials and features - roads, vegetation, water bodies and urban structures.

The integration of contextual information [30][17] can further enhance the classification's accuracy. Contextual information, such as geographic metadata, land use patterns and spatial relationships between different features, can aid in distinguishing between roads and other land cover classes. For instance, by analyzing changes in land cover patterns over time, the model can adapt its classification criteria to account for temporal variations and improve its ability to differentiate between roads and other features.

To address misclassification problems that may be associated with the block's size, the use of an adaptive sliding window can be proven to be a solution. By training the model using a combination of different patch sizes, with larger patch sizes specifically targeted towards areas with larger road segments or agricultural areas it would allow the model to capture more detailed spatial information and boundary features associated with roads. Such an approach could reduce the likelihood of misclassifications, particularly in areas with diverse land cover types.

While the Dissertation provides valuable insights contributing to the development of road identification in satellite imagery, it also underscores the need for continued research and innovation in this field. By critically evaluating previous topics and adapting the methodologies accordingly, advancements can be achieved in the state-of-the-art of satellite image analysis.

Bibliography

[1] Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Accessed: December 20, 2023. Springer, 2018. ISBN: 978-3-319-94463-0. DOI: \url{https://doi.org/10.1007/978-3-319-94463-0}. URL: %5Curl%7Bhttps://www.academia.edu/42981452/Neural_Networks_and_Deep_Learning_Charu_C_Aggarwal%7D.

[2] AWS. *What's the Difference Between GPUs and CPUs?* https://aws.amazon.com/compare/the-difference-between-gpus-cpus/?nc1=h_ls. Accessed: December 20, 2023.

[3] Ana Beduschi. "Harnessing the potential of artificial intelligence for humanitarian action: Opportunities and risks". In: *International Review of the Red Cross* 104 (919 Apr. 2022), pp. 1149–1169. ISSN: 16075889. DOI: 10.1017/S1816383122000261.

[4] Rodrigo F. Berriel et al. *Deep Learning Based Large-Scale Automatic Satellite Crosswalk Classification*. Sept. 2017. DOI: 10.1109/LGRS.2017.2719863. URL: http://arxiv.org/abs/1706.09302%20http://dx.doi.org/10.1109/LGRS.2017.2719863.

[5] Derrick Bonafilia et al. "Building High Resolution Maps for Humanitarian Aid and Development with Weakly-and Semi-Supervised Learning". In: 2019, pp. 1–9.

[6] Pete Chapman et al. *CRISP-DM 1.0: Step-by-step data mining guide*. 2000.

[7] Copernicus. *Copernicus in detail*. https://www.copernicus.eu/en/about-copernicus/copernicus-detail. Accessed: December 20, 2023.

[8] Ilke Demir et al. "DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images". In: May 2018, pp. 172–181. DOI: 10.1109/CVPRW.2018.00031. URL: http://arxiv.org/abs/1805.06561%20http://dx.doi.org/10.1109/CVPRW.2018.00031.

[9] Zohreh Dorrani. "Road Detection with Deep Learning in Satellite Images". In: *Majlesi Journal of Telecommunication Devices* 12 (1 2023), pp. 43–47. DOI: 10.30486/mjtd.2023.1979006.1024.

[10] IOM DTM. *Methodology for IOM COVID-19 Impact on Points of Entry and Other Key Locations of Internal Mobility*. Oct. 2020.

[11] GISGeography. *Passive vs Active Sensors in Remote Sensing*. https://gisgeography.com/passive-active-sensors-remote-sensing/. Accessed: December 20, 2023. 2023.

[12] Carolina Gonçalves et al. "Automatic detection of Acacia longifolia invasive species based on UAV-acquired aerial imagery". In: *Information Processing in Agriculture*

9 (2 June 2022), pp. 276–287. DOI: https://doi.org/10.1016/j.inpa.2021.04.007.

[13] Lei He et al. "Road Extraction Based on Improved Convolutional Neural Networks with Satellite Images". In: *Applied Sciences (Switzerland)* 12 (21 Nov. 2022). ISSN: 20763417. DOI: 10.3390/app122110800.

[14] Tamara Keijzer et al. *Detecting Roads From Space: Testing the Potential of Sentinel-1 SAR Imagery and Deep Learning for Automated Road Mapping*. Mar. 2022.

[15] Weijia Li et al. "Deep learning based oil palm tree detection and counting for high-resolution remote sensing images". In: *Remote Sensing* 9 (1 2017). ISSN: 20724292. DOI: 10.3390/rs9010022.

[16] Chuang Zhang Lichen Zhou and Ming Wu. "D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction." In: 2018, pp. 182–186.

[17] D. Lu and Q. Weng. *A survey of image classification methods and techniques for improving classification performance*. 2007. DOI: 10.1080/01431160600746456.

[18] Emmanuel Maggiori et al. "Convolutional Neural Networks for Large-Scale Remote Sensing Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55 (2017), pp. 645–657. DOI: 10.1109/tgrs.2016.2612821. URL: https://inria.hal.science/hal-01369906.

[19] Gary Marcus. *Deep learning: A critical appraisal*. https://arxiv.org/abs/1801.00631. Accessed: December 20, 2023. 2018.

[20] Earth Data NASA. *Active Sensors, Information about active sensors used for NASA EOSDIS remote sensing of Earth science data.* https://www.earthdata.nasa.gov/learn/backgrounders/active-sensors. Accessed: December 20, 2023. 2021.

[21] Earth Data NASA. *Passive Sensors, Learn about passive sensors used for remote sensing of NASA Earth science data.* https://www.earthdata.nasa.gov/learn/backgrounders/passive-sensors. Accessed: December 20, 2023. 2021.

[22] Earth Data NASA. *What is Remote Sensing? Tutorial on remotely-sensed data, from sensor characteristics, to different types of resolution, to data processing and analysis.* https://www.earthdata.nasa.gov/learn/backgrounders/remote-sensing. Accessed: December 20, 2023. 2019.

[23] Nathaniel Raymond and Ziad Al Achkar. *Data preparedness: connecting data, decision-making and humanitarian response*. 2016.

[24] Embaixada da República de Angola nos Estados Unidos da América. *O País*. https://angola.org/o-pais/. Accessed: December 20, 2023.

[25] Rajalingappaa Shanmugamani. *Deep learning for computer vision: Expert techniques to train advanced neural networks using TensorFlow and Keras*. Accessed: December 20, 2023. Packt Publishing Ltd., 2019. ISBN: 978-1-78829-562-8. URL: %5Curl%7Bhttps://studylib.net/doc/25735059/moore--stephen-shanmugamani--rajalingappaa---deep-learnin...%7D.

[26] Siamak Khorram and Frank H. Koch and Cynthia F. van der Wiele and Stacy A. C. Nelson. *Remote Sensing*. Springer Science and Business Media, 2012.

[27] Sidharth. *Convolutional Neural Network (CNN): Architecture Explained — Deep Learning*. https://www.pycodemates.com/2023/06/introduction-to-convolutional-neural-networks.html?utm_content=cmp-true. Accessed: April 16, 2024. 2023.

[28] Humanitarian OpenStreetMap Team. *What We Do*. https://www.hotosm.org/what-we-do. Accessed: May 8, 2024.

[29] Maxar Technologies. *Maxar's High-Resolution Vivid Basemaps Enhances Esri ArcGIS Living Atlas of the World*. https://www.maxar.com/press-releases/maxar-s-high-resolution-vivid-basemaps-enhances-esri-arcgis-living-atlas-of-the-world. Accessed: December 20, 2023. 2022.

[30] Aaron Thegeya et al. "Application of Machine Learning Algorithms on Satellite Imagery for Road Quality Monitoring: An Alternative Approach to Road Quality Surveys". Dec. 2022. DOI: 10.22617/WPS220587-2. URL: https://www.adb.org/publications/machine-learning-satellite-imagery-road-quality-monitoring.

[31] Devis Tuia, Claudio Persello, and Lorenzo Bruzzone. *Domain adaptation for the classification of remote sensing data: An overview of recent advances*. June 2016. DOI: 10.1109/MGRS.2016.2548504.

[32] USGS. *What is remote sensing and what is it used for?* https://www.usgs.gov/faqs/what-remote-sensing-and-what-it-used. Accessed: December 20, 2023. 2017.

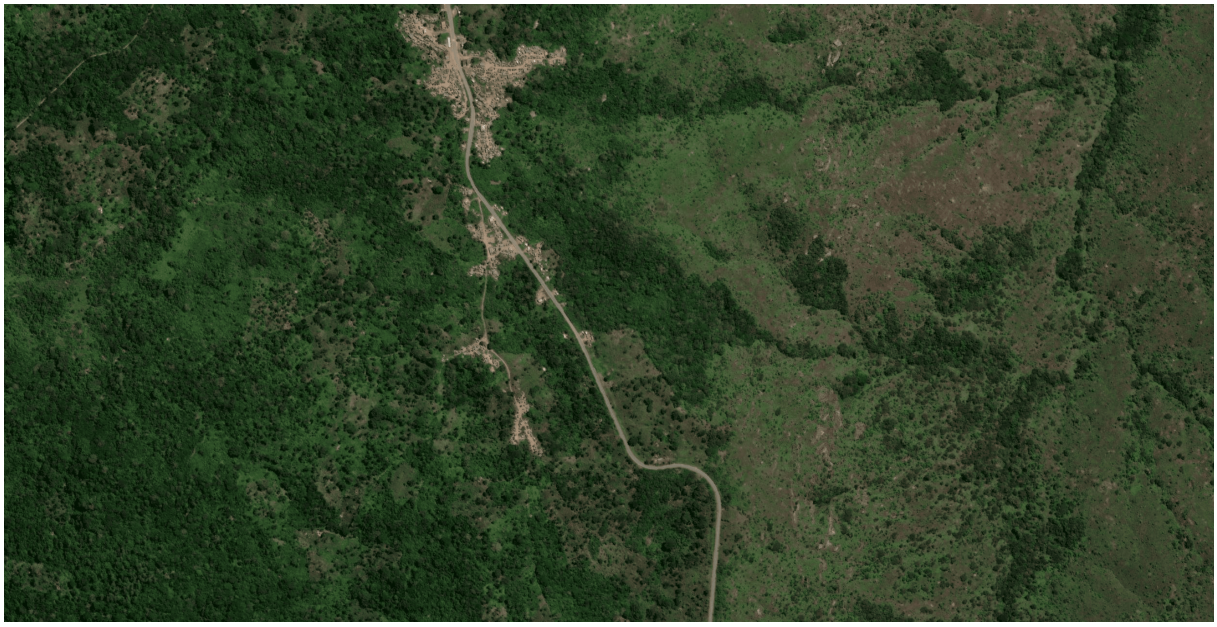
[33] Jiang Xin et al. "Road extraction of high-resolution remote sensing images derived from DenseUNet". In: *Remote Sensing* 11 (21 Nov. 2019), pp. 1–18. ISSN: 20724292. DOI: 10.3390/rs11212499.

[34] Kaili Yang et al. "Semi-Automatic Method of Extracting Road Networks from High-Resolution Remote-Sensing Images". In: *Applied Sciences (Switzerland)* 12 (9 May 2022). ISSN: 20763417. DOI: 10.3390/app12094705.

[35] Wenzhi Zhao, Shihong Du, and William J. Emery. "Object-Based Convolutional Neural Network for High-Resolution Imagery Classification". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10 (7 July 2017), pp. 3386–3396. ISSN: 21511535. DOI: 10.1109/JSTARS.2017.2680324.

APPENDIX A

Detailed Example of the Prepared Data.



(a) Example of a Raster file.

(b) Example of a Ground Truth file.

APPENDIX B

Results that support the choice of the dataset

B.1. Results of the sample datasets that support the establishment of the block's size of the dataset

Model Specifics		Train Set		Validation Set				
Model_BS	Data	Loss	Acc.	Loss	Acc.	Recall	Precis.	F1 Score
CCPCCP_32	Medium	0.2847	0.8877	0.2620	0.8996	0.9167	0.8893	0.9028
CPCPCP_64	Small	0.3419	0.8520	0.2809	0.8794	0.9094	0.8611	0.8846
CCPCCP_32	Small	0.3676	0.8396	0.3199	0.8665	0.9312	0.8277	0.8764
CCPCCP_64	Big	0.2923	0.8764	0.2864	0.8757	0.8569	0.8941	0.8751
CCPCCP_128	Small	0.3372	0.8507	0.3305	0.8637	0.9221	0.8290	0.8731
CPCPCP_32	Medium	0.3018	0.8734	0.3373	0.8812	0.9312	0.8120	0.8675
CCPCCP_64	Medium	0.2842	0.8845	0.3296	0.8656	0.8514	0.8801	0.8655
CCPCCP_32	Big	0.3458	0.8491	0.3165	0.8665	0.8333	0.8967	0.8638
CCPCCP_128	Big	0.3115	0.8645	0.3676	0.8490	0.8804	0.8322	0.8556
CPCP_128	Big	0.3521	0.8343	0.3559	0.8444	0.9004	0.8134	0.8547
CCPCCP_64	Small	0.3563	0.8417	0.3581	0.8527	0.8496	0.8590	0.8543
CPCPCP_32	Big	0.3146	0.8654	0.3476	0.8481	0.8732	0.8354	0.8539
CPCP_32	Big	0.3245	0.8629	0.3338	0.8637	0.8207	0.8865	0.8523
CCPCCP_128	Medium	0.2630	0.8932	0.3842	0.8536	0.8641	0.8224	0.8427
CPCPCP_128	Medium	0.2306	0.9107	0.5762	0.8462	0.8297	0.8561	0.8427
CPCP_128	Small	0.3723	0.8283	0.4260	0.8352	0.8877	0.8007	0.8420
CPCPCP_64	Medium	0.3013	0.8746	0.3919	0.8333	0.8750	0.8063	0.8392
CPCP_64	Small	0.4009	0.8152	0.3905	0.8287	0.8243	0.8537	0.8387
CPCP_32	Small	0.4636	0.7814	0.4060	0.8260	0.8533	0.8079	0.8300
CPCPCP_128	Small	0.3975	0.8157	0.3862	0.8306	0.8134	0.8472	0.8300
CPCP_128	Medium	0.3377	0.8442	0.3952	0.8232	0.8949	0.7731	0.8296
CPCPCP_32	Small	0.5005	0.7660	0.4238	0.8140	0.8424	0.8017	0.8215
CPCP_64	Big	0.3659	0.8341	0.6043	0.8158	0.8388	0.8024	0.8202
CPCP_32	Medium	0.4031	0.8166	0.4262	0.8122	0.8315	0.8053	0.8182
CPCPCP_128	Big	0.2968	0.8741	0.3734	0.8407	0.7699	0.8638	0.8142
CPCPCP_64	Big	0.2984	0.8748	0.6068	0.8333	0.7880	0.8161	0.8018
CPCP_64	Medium	0.4106	0.8067	0.6187	0.7781	0.8986	0.7147	0.7962

B.2. Results of the additional experiments to support the choice of the block's size of the dataset

Model Specifics		Train Set		Validation Set				
Model_BS	Data	Loss	Acc.	Loss	Acc.	Recall	Precis.	F1-Sc.
RESNET50_32	Big Sample	0.1458	0.9418	0.1821	0.9411	0.9402	0.9335	0.9368
RESNET50_64	Big Sample	0.1453	0.9406	0.1786	0.9392	0.9312	0.9397	0.9354
RESNET50_128	Big Sample	0.1316	0.9468	0.1826	0.9383	0.9384	0.9283	0.9333
RESNET50_128	Medium 20%	0.2207	0.9077	0.2312	0.9086	0.8938	0.9184	0.9059
RESNET50_32	Medium 20%	0.2311	0.9042	0.2401	0.9061	0.9030	0.9042	0.9036
RESNET50_64	Medium 20%	0.2213	0.9066	0.2335	0.9026	0.8713	0.9164	0.8933
CPCPCP_64	Medium 20%	0.2456	0.8978	0.2547	0.8915	0.8727	0.9098	0.8909
CPCPCP_32	Medium 20%	0.2880	0.8791	0.2823	0.8790	0.8924	0.8722	0.8822
CPCPCP_128	Medium 20%	0.2679	0.8844	0.2831	0.8758	0.8685	0.8847	0.8765
CCPCCP_64	Big Sample	0.2923	0.8764	0.2864	0.8757	0.8569	0.8941	0.8751
CCPCCP_32	Medium 20%	0.3069	0.8664	0.2972	0.8687	0.8776	0.8655	0.8715
CCPCCP_64	Medium 20%	0.3217	0.8569	0.3386	0.8572	0.9290	0.8154	0.8685
CCPCCP_64	Small 20%	0.3212	0.8609	0.8695	0.8695	0.8510	0.8827	0.8666
CCPCCP_32	Small 20%	0.3381	0.8531	0.3093	0.8667	0.8496	0.8786	0.8639
CCPCCP_32	Big Sample	0.3458	0.8491	0.3165	0.8665	0.8333	0.8967	0.8638
CPCPCP_32	Small 20%	0.3575	0.8421	0.3193	0.8632	0.8428	0.8776	0.8598
CPCP_128	Small 20%	0.3411	0.8512	0.3272	0.8585	0.8670	0.8516	0.8592
CPCPCP_128	Small 20%	0.3333	0.8530	0.3128	0.8653	0.8269	0.8931	0.8587
CCPCCP_128	Small 20%	0.3211	0.8606	0.3289	0.8578	0.8613	0.8513	0.8563
CCPCCP_128	Big Sample	0.3115	0.8645	0.3676	0.8490	0.8804	0.8322	0.8556
CPCP_128	Big Sample	0.3521	0.8343	0.3559	0.8444	0.9004	0.8134	0.8547
CPCPCP_64	Small 20%	0.3560	0.8420	0.3340	0.8534	0.8684	0.8407	0.8543
CPCPCP_32	Big Sample	0.3146	0.8654	0.3476	0.8481	0.8732	0.8354	0.8539
CPCP_32	Big Sample	0.3245	0.8629	0.3338	0.8637	0.8207	0.8865	0.8523
CCPCCP_128	Medium 20%	0.3374	0.8533	0.3581	0.8498	0.8390	0.8614	0.8501
CPCP_64	Small 20%	0.3797	0.8303	0.3641	0.8401	0.8352	0.8555	0.8452
CPCP_32	Medium 20%	0.4057	0.8162	0.3836	0.8244	0.9304	0.7710	0.8432
CPCP_64	Medium 20%	0.3971	0.8153	0.4006	0.8233	0.8706	0.7992	0.8334
RESNET50_128	Small 20%	0.4001	0.8169	0.3888	0.8217	0.8594	0.7894	0.8229
CPCP_32	Small 20%	0.4204	0.8062	0.3863	0.8259	0.8057	0.8383	0.8217
CPCP_64	Big Sample	0.3659	0.8341	0.6043	0.8158	0.8388	0.8024	0.8202
RESNET50_64	Small 20%	0.4075	0.8131	0.3946	0.8210	0.8315	0.8085	0.8198
CPCPCP_128	Big Sample	0.2968	0.8741	0.3734	0.8407	0.7699	0.8638	0.8142
RESNET50_32	Small 20%	0.4164	0.8097	0.3974	0.8203	0.7790	0.8336	0.8054
CPCPCP_64	Big Sample	0.2984	0.8748	0.6068	0.8333	0.7880	0.8161	0.8018
MOBILENET_128	Big Sample	0.4553	0.7830	0.4731	0.7772	0.8062	0.7594	0.7821
MOBILENET_32	Big Sample	0.4497	0.7821	0.4714	0.7689	0.8279	0.7335	0.7778
MOBILENET_64	Medium 20%	0.5188	0.7410	0.5172	0.7448	0.8143	0.7161	0.7620
MOBILENET_128	Medium 20%	0.5152	0.7389	0.5144	0.7441	0.7623	0.7344	0.7481
MOBILENET_64	Big Sample	0.4518	0.7855	0.4730	0.7808	0.6938	0.8029	0.7444
MOBILENET_32	Medium 20%	0.5206	0.7342	0.5182	0.7441	0.6934	0.7414	0.7166
MOBILENET_128	Small 20%	0.6558	0.6026	0.6577	0.6031	0.7070	0.5804	0.6375
MOBILENET_64	Small 20%	0.6563	0.6008	0.6595	0.6036	0.6669	0.5826	0.6219
MOBILENET_32	Small 20%	0.6563	0.6021	0.6585	0.6047	0.6591	0.5820	0.6182

B.3. Results of the additional experiments to support the choice of the condition to use in the creation of the 64x64 block size dataset

Model Specifics		Train Set		Validation Set				
Model_BS	Dataset	Loss	Acc.	Loss	Acc.	Recall	Precis.	F1-Sco.
RESNET50_128	M. 20%	0.2207	0.9077	0.2312	0.9086	0.8938	0.9184	0.9059
RESNET50_32	M. 20%	0.2311	0.9042	0.2401	0.9061	0.9030	0.9042	0.9036
RESNET50_64	M. 20%	0.2213	0.9066	0.2335	0.9026	0.8713	0.9164	0.8933
CCPCCP_32	M. 20%	0.3069	0.8664	0.2972	0.8687	0.8776	0.8655	0.8715
CCPCCP_64	M. 20%	0.3217	0.8569	0.3386	0.8572	0.9290	0.8154	0.8685
CCPCCP_64	M. 10%	0.2591	0.8913	0.2670	0.8865	0.8619	0.8742	0.8680
CCPCCP_128	M. 10%	0.2850	0.8774	0.3138	0.8625	0.8563	0.8720	0.8641
RESNET50_128	M. 10%	0.3070	0.8641	0.3133	0.8649	0.8585	0.8691	0.8638
CCPCCP_128	M. 20%	0.3374	0.8533	0.3581	0.8498	0.8390	0.8614	0.8501
CCPCCP_32	M. Center	0.3183	0.8591	0.3056	0.8704	0.8003	0.8989	0.8467
RESNET50_64	M. 10%	0.3729	0.8329	0.3486	0.8474	0.8547	0.8350	0.8447
RESNET50_32	M. 10%	0.3780	0.8326	0.3559	0.8463	0.8710	0.8155	0.8423
CCPCCP_64	M. Center	0.2913	0.8722	0.2938	0.8769	0.7968	0.8783	0.8356
CCPCCP_32	M. 10%	0.3541	0.8444	0.3110	0.8620	0.7542	0.9124	0.8258
RESNET50_64	M. Center	0.4956	0.7545	0.4613	0.7960	0.7978	0.7813	0.7895
RESNET50_128	M. Center	0.4933	0.7573	0.4603	0.7960	0.7760	0.8000	0.7878
RESNET50_32	M. Center	0.4980	0.7551	0.4745	0.7905	0.7821	0.7731	0.7776
CCPCCP_128	M. Center	0.2620	0.8889	0.2643	0.8837	0.9191	0.6278	0.7460

APPENDIX C

Results of the Training phase

Model Specifics	Train Set		Validation Set				
Model_BS	Loss	Acc.	Loss	Acc.	Recall	Precis.	F1 Score
CPCPCP+1D_32	0.2956	0.8743	0.2724	0.8858	0.8672	0.8951	0.8809
CCPCCP_64	0.3009	0.8692	0.2876	0.8794	0.8731	0.8800	0.8766
CCPCCP+1D_128	0.2690	0.8864	0.2850	0.8797	0.8420	0.8896	0.8651
CCPCCP_128	0.2850	0.8774	0.3138	0.8625	0.8563	0.8720	0.8641
RESNET50_128	0.3070	0.8641	0.3133	0.8649	0.8585	0.8691	0.8638
RESNET50+2D_32	0.3269	0.8562	0.3149	0.8634	0.8681	0.8570	0.8625
RESNET50+1D_128	0.3192	0.8590	0.3130	0.8645	0.8513	0.8733	0.8622
CPCPCP_128	0.3081	0.8666	0.2990	0.8741	0.8814	0.8435	0.8620
RESNET50+1D_32	0.3261	0.8548	0.3182	0.8651	0.8781	0.8435	0.8604
CPCPCP+1D_128	0.2871	0.8773	0.2852	0.8819	0.8308	0.8918	0.8602
RESNET50+2D_64	0.3159	0.8601	0.3179	0.8628	0.8672	0.8518	0.8594
RESNET50+2D_128	0.3112	0.8648	0.3129	0.8644	0.8500	0.8658	0.8578
RESNET50+1D_64	0.3161	0.8609	0.3154	0.8632	0.8493	0.8638	0.8565
CPCPCP_32	0.3129	0.8651	0.2922	0.8759	0.7921	0.9234	0.8527
RESNET50_64	0.3729	0.8329	0.3486	0.8474	0.8547	0.8350	0.8447
RESNET50_32	0.3780	0.8326	0.3559	0.8463	0.8710	0.8155	0.8423
CPCPCP+1D_64	0.3056	0.8693	0.2849	0.8785	0.7655	0.9209	0.8361
CPCPCP_64	0.3063	0.8677	0.3005	0.8719	0.7538	0.9366	0.8353
CCPCCP_32	0.3541	0.8444	0.3110	0.8620	0.7542	0.9124	0.8258
CCPCCP+1D_32	0.3115	0.8672	0.2931	0.8794	0.7190	0.9228	0.8082
CCPCCP+1D_64	0.2750	0.8845	0.2765	0.8849	0.7067	0.9086	0.7950

APPENDIX D

Points of Entry Identified

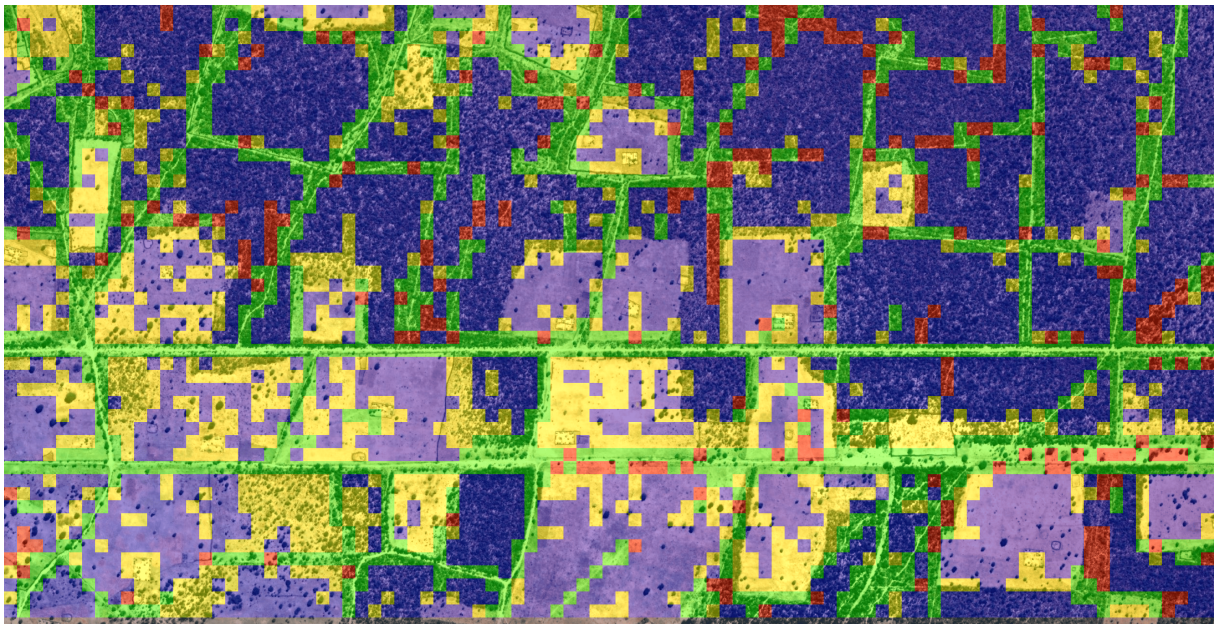
PoE ID	Coordinate	
01	-14.837489	21.993344
02	-14.417424	21.991025
03	-14.41986	21.991006
04	-14.257085	21.990375
05	-13.000377	22.544621
06	-13.000383	22.548848
07	-13.000373	22.550338
08	-13.000373	22.550682
09	-13.000593	22.727183
10	-17.390539	17.232142
11	-17.390512	17.288274
12	-17.390355	17.294509
13	-17.390355	17.296961
14	-17.390355	17.306013
15	-17.390355	17.309587
16	-17.390381	17.312066
17	-17.390277	17.31564
18	-17.390277	17.31851
19	-17.390277	17.377651
20	-17.390277	17.379685
21	-17.390329	17.384851
22	-17.390277	17.392155
23	-17.390225	17.396538
24	-17.390147	17.399773
25	-17.390147	17.754437
26	-17.390408	18.40992
27	-18.024785	21.433108
28	-13.880708	21.988753
29	-13.882456	21.988779
30	-13.852715	21.988753
31	-13.000009	22.202308
32	-12.999983	22.229752
33	-13.000348	22.515805
34	-11.58957	24.001533
35	-11.150486	24.035917
36	-11.109893	24.023395
37	-11.111458	24.023577
38	-7.282289	21.622344
39	-5.866924	15.047642
40	-5.865857	15.014183
41	-17.390623	17.138309
42	-17.390606	17.14821
43	-17.390623	17.143051
48	-17.390616	17.1528
45	-17.390642	17.154678
46	-17.390642	17.160417
47	-17.390572	17.160883

APPENDIX E

Examples of output images

E.1. Output Number 1

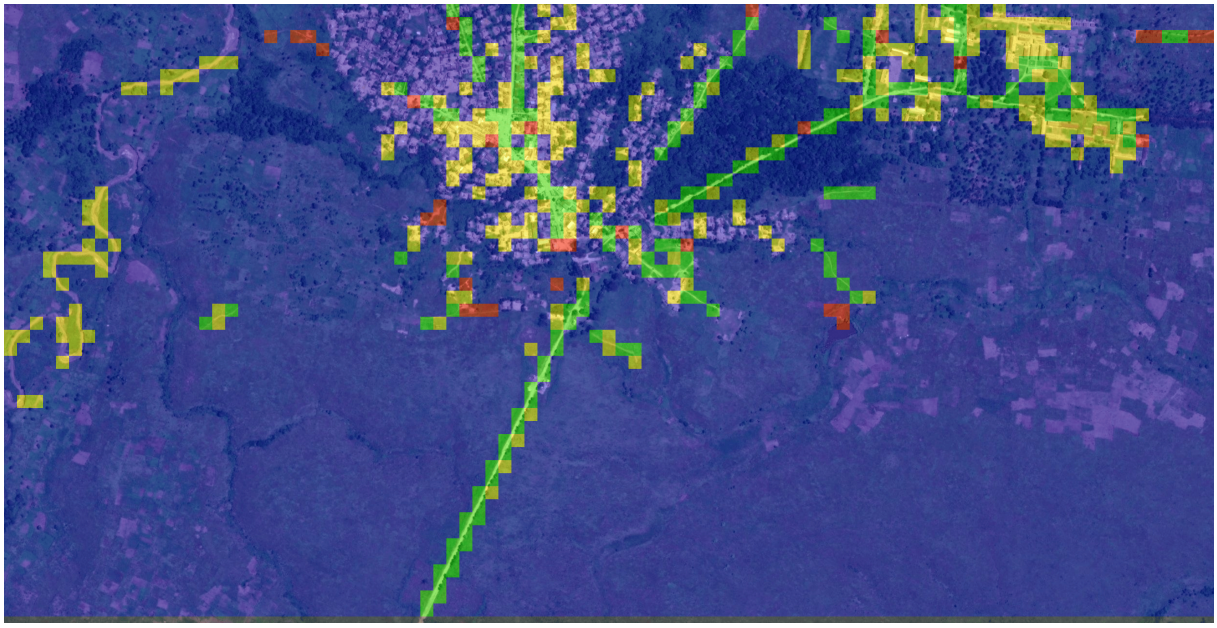
(a) Original Input Image.



(b) Color-Coded Output Image

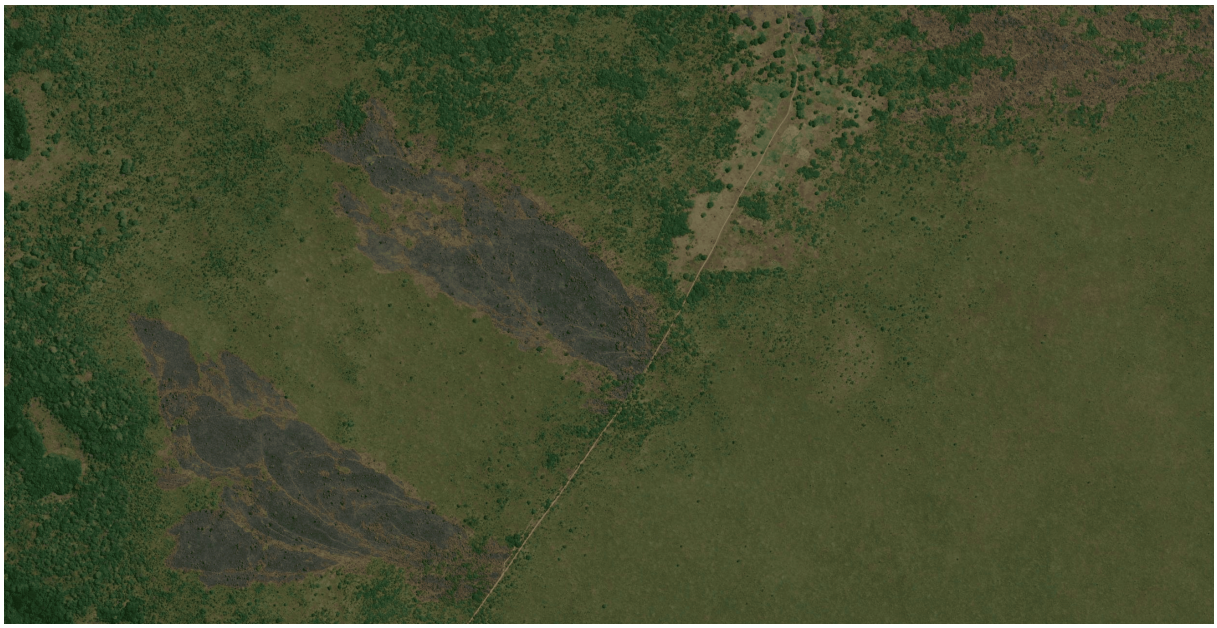
E.2. Output Number 2

(c) Original Input Image.

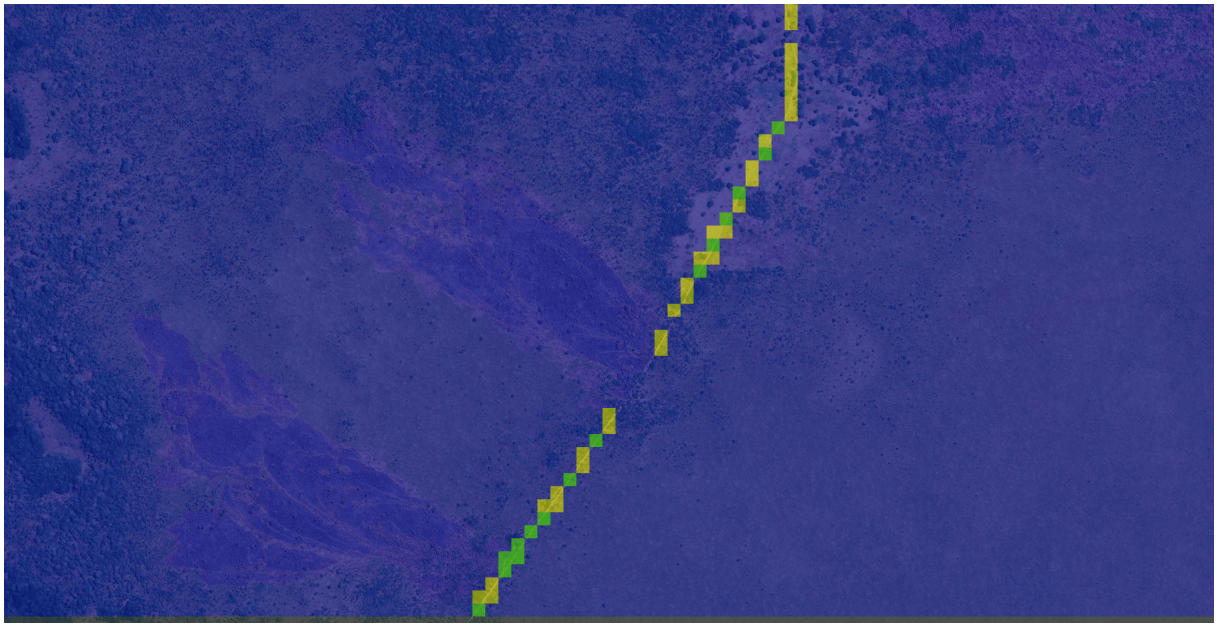


(d) Color-Coded Output Image

E.3. Output Number 3



(e) Original Input Image.



(f) Color-Coded Output Image