POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Multitask segmentation from satellite imagery for burned area delineation and severity estimation

Supervisors

Prof. Paolo GARZA

Dr. Edoardo ARNAUDO

Dr. Luca BARCO

Candidate

Matteo MERLO

July 2023

Abstract

In recent years, the frequency and intensity of natural disasters has significantly and dangerously increased in Europe and in the world due to factors such as climate change, population growth and aggressive urbanization of rural areas. Every year, hundreds of wildfires destroy millions of hectares of forest. Rapidly delineating burned areas from satellite has become a crucial task for first responders and decision makers, to enhance the preparedness, response and recover phases during such crises.

The European Union and the European Space Agency are intensifying their efforts to accumulate information on natural disasters. Data about past catastrophic events are collected by Copernicus Emergency Management System (CEMS) and categorized according to the type of event. Exploiting wildfire EMS activations, the first objective of this thesis was the generation of a large dataset focused mainly on the European soil, collecting satellite imagery from Sentinel-2. The dataset includes different maps, including delineation and grading masks provided by EMS, as well as a cloud cover label to mask clouds in the images, thus reducing possible errors during inference.

Starting from this dataset, this thesis also proposes a multitask learning semantic segmentation approach for wildfire delineation and burn severity estimation. A multitasking scenario allows a model to jointly learn from both delineation and severity estimation of wildfires. Several state-of-the-art semantic segmentation models are tested to assess their performance in both burned area delineation and severity estimation, using post-wildfire images only. Experiments show that the combination of a large dataset and the multitask approach allows to reach robust results, achieving F1 scores over 90 considering the delineation, and RMSE scores lower than 0.9 for severity estimates.

”Man must rise above the Earth, to the top of the atmosphere and beyond,
for only thus will he fully understand the world in which he lives.”
Plato, Phaedo, IV century BC

Acknowledgements

I would like to extend my deepest gratitude to Edoardo Arnaudo and Paolo Garza for their guidance and willingness to supervise this thesis. My sincere thanks also go to Luca Barco from the LINKS Foundation. His patience, support, and constructive feedback throughout the course of this project and the writing of this thesis were invaluable.

I owe a great debt of gratitude to my family, whose unwavering support throughout my years at university was essential. I am deeply appreciative of the financial aid and the sacrifices made on their part, which enabled me to pursue my master's degree at the Polytechnic of Turin. My outward demeanor may not always show it, but I consider myself fortunate to have you in my life and for that, I am profoundly grateful.

A special thanks to the ICARUS team and all the people that i worked with, especially to my friends "Elettrodomestici" for the fantastic experience that brought me so much both from a human point of view and from a professional point of view

Lastly, but certainly not least, I would like to express my heartfelt thanks to the friends and acquaintances I've made during my university years. The future may be uncertain, but the memories of shared laughter, in-depth conversations, and shared experiences will always be treasured. I look forward to the multitude of new experiences that lie ahead on our life journeys, hoping that we will share as many as possible.

Table of Contents

List of Tables V

List of Figures VI

1 Introduction 1

2 Related Works and Background 4

2.1 Artificial Neural Networks 4

2.2 Convolutional Neural Networks 5

2.3 Training Process 7

2.4 Semantic Segmentation 9

2.5 Burned Area Delineation And Severity Estimation 10

3 Dataset 13

3.1 Data Sources 13

3.1.1 Copernicus EMS 13

3.1.2 Sentinel-2 16

3.1.3 Land cover 18

3.2 Data Processing 18

3.2.1 Cloud Masking 19

3.3 Dataset information 20

4 Methodology 23

4.1 Problem statement 23

4.2 Multitask approach 24

4.3 Network Architectures 25

4.3.1 U-Net 25

4.3.2 U-Net++ 26

4.3.3 DeeplabV3+ 27
4.3.4 PSPNet 29

5 Experiments **31**
5.1 Implementation Details 31
5.1.1 Experimental setup 31
5.1.2 Training process 32
5.2 Loss Functions 36
5.3 Evaluation metrics 38
5.4 Results 40
5.4.1 Delineation 40
5.4.2 Grading 43

6 Conclusions 46

A 47

Bibliography 50

List of Tables

3.1 Severity level and relative tag names in both damage assessment convention and relative color of the mask 16
3.2 Sentinel-2 bands description 17
3.3 CloudSen12 levels and color in the mask 20
3.4 Class imbalance in binary case 21
3.5 Class imbalance in multi-class case 21

5.1 Subdivision of the dataset 32
5.2 Python packages and versions installed 34
5.3 Summary of all training parameters 35
5.4 Data augmentation parameters 36
5.5 Summary of the evaluation metrics for both the binary case and the multiclass case 39
5.6 Results of wildfire delineation task 41
5.7 Results of damage severity estimation task 44

A.1 Dataset 48

List of Figures

2.1 Visual comparison between a biological neuron and a perceptron (Source [5]). 5
2.2 Artificial Neural Network (source [6]). 6
2.3 Convolutional layer (source [7]). 6
2.4 Learning rate scenarios (source [8]). 8
2.5 Burned areas reflect other wavelengths more strongly than vegetation and vice versa. Image source [19] 10

3.1 In clockwise direction: the Sentinel2 image with level L2A, FEP mask, DEL mask and GRA mask.The FEP mask is incomplete because it was made during an earlier stage of the fire. 15
3.2 Sample output from CloudSen12 20
3.3 Distribution of the activations on European region, the size of the point is logarithmic proportional to the fire extent 22

4.1 The architecture of the multitask segmentation model with the two segmentation heads 24
4.2 U-Net encoder-decoder structure (Source [11]). 26
4.3 U-Net++ consists of an encoder and decoder that are connected through a series of nested dense convolutional blocks (Source [12]). 28
4.4 DeepLabV3+ architecture with ASPP module (Source [14]). . . . 29
4.5 PSPNet architecture and PPM (Source [16]). 30

5.1 The distribution of the training (yellow dots), validation (red dots), and test set (blue dots). 33
5.2 Cosine annealing with exponential decay over the epoch 35
5.3 Example results of wildfire delineation task. 42
5.4 Example results of damage severity estimation task 45

Chapter 1

Introduction

Catastrophic hazards are of different nature, like floods, earthquakes, and severe storms. These events represent potential threats from many perspectives: humanitarian, economic, and environmental [1].

Among them, forest fires have always stood out for their destructive power and frequency. Floods, earthquakes and violent storms certainly represent a significant threat, but wildfires in particular have the capability to affect multiple areas simultaneously. Beside environmental destruction, they also cause a devastating impact on the economy. An additional level of difficulty is their propagation and spreading in remote or difficult-to-access areas: their detection and timely intervention to extinguish them can be problematic.

From an environmental perspective, local flora and fauna suffer from the destruction of the land, radically altering native ecosystems and endangering biodiversity. Therefore, the combustion of the forests emits a huge quantity of CO2 in the atmosphere, accelerating climate change. Additionally, the reduction of vegetation has long-term environmental impacts: the area's ability to absorb CO2 is compromised, and exposed soil without a healthy tree cover increases the risk of erosion, then causing events such as landslides and floods.

Finally, smokes and particulate emissions caused by fire contribute to air pollution, with potentially harmful effects on air quality and human health. Such phenomena when they occur can even last for days. Winds can push smoke into remote areas, even far away from the fires, as happened in early June 2023 in New York City, which was shrouded in smoke from Canadian wildfires [2].

From an economic aspect, fires also cause extensive damage to private property, causing significant economic impacts to the affected area. In addition, the efforts required to extinguish the forest fire and the subsequent recovery and restoration after the event must be taken into account. Local heritage and historic places may also suffer irreparable losses.

The restoration of an area after a fire could take a long time. Depending on the extent, damage severity, and the specific characteristics of the soil and ecosystem, a full recovery can take decades or even centuries [3]. Therefore, fire prevention becomes even more important: prompt intervention to extinguish fires in their early stages and monitor closely burned areas after the flames have been extinguished.

Global awareness of these kinds of global issues is steadily increasing, underlined by the atrocious realities of the 2022 wildfire season in Europe. As nations grapple with increased fire risk, and escalating drought conditions, the need for robust, coordinated international responses becomes ever more urgent [4].

Facing with this challenge, this thesis involves an implementation of an image segmentation model that can both identify burned areas and provide an accurate assessment of fire severity through the analysis of satellite imagery, through which it is possible to acquire data about ground conditions all around the world. The aim is to implement a post-fire detection system that can provide valuable information for post-event management.

In this context, the application of deep learning models is proving to be a promising approach for wildfire prevention and management. The objective of this thesis is intended to add a small contribution to the promotion of a more sustainable and resilient approach to fire risk management.

Specifically, the developed model aims to achieve two main objectives: delineation of burned areas and assessment of fire severity. Through a multitask approach, the model is trained to identify fire-affected areas with satellite imagery only, separating the burned area from the surrounding context. In parallel, the model is also able to determine fire severity. This analysis of the soil destruction is done considering factors such as the extent of the burned area and the variation in ground reflectance values of satellite images, which are closely correlated with the severity of the burned area.

The implementation of such multitask model is intended to offer a possible alternative over other traditional fire monitoring methods from literature, allowing more accurate and timely identification and assessment of burned areas.

In this thesis, in particular, several state-of-the-art semantic segmentation models are tested to assess their performance in both burned area delineation and severity estimation, by processing post-fire satellite acquisitions. The experimental results show that the combination of a large dataset and the multitask approach allows us to reach robust results. The final results obtained are compared with the existing state-of-the-art on the subject, underlying the better performance achieved with a multitask approach.

The remainder of this document is organized as follows. Chapter 2 offers an extensive exploration of the theoretical underpinnings of Deep Learning, then it proceeds to analyze the most recent advancements and methodologies used within the field of image segmentation, particularly focusing on tasks related to the delineation and severity estimation of wildfires. Chapter 3 provides a detailed account of how the dataset was created and the processing techniques employed on the satellite imagery. Subsequently, in Chapter 4 is outlined the structure and functionality of the multitask model, accompanied by a complete analysis of the models that are examined in this thesis. In Chapter 5, are summarized the methods used to test the model and assess its performance. The thesis concludes with Chapter 6, offering concluding thoughts and future improvements.

Chapter 2

Related Works and Background

This chapter provides an introduction to deep learning, focusing particularly on neural networks and their training processes. Secondly, it proceeds to discuss the main methods and techniques found in the literature that address the challenge of delineating burned areas and estimating the severity of damage from wildfires.

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computing systems inspired by the biological neural networks that constitute animals' brain. The main goal is to mimic the electrical interactions between neurons using a simplified mathematical model.

In our brain, biological neurons communicate with each other by receiving input messages through fibers known as dendrites. Analogously, an artificial neuron, also known as *perceptron*, receives information from other input neurons.

The connections between input neurons and perceptrons in ANNs are often described by the term *weights*. These weights signify the importance or weight of each specific input on the function of the perceptron. This concept is comparable to the synapses in our brain, the connections between dendrites and neurons.

Then, similarly to how the nucleus of a biological neuron uses signals from the dendrites to generate an output signal, the perceptron in an ANN processes the input values to generate output values.

Finally, much like how an axon in a biological neuron transports the output

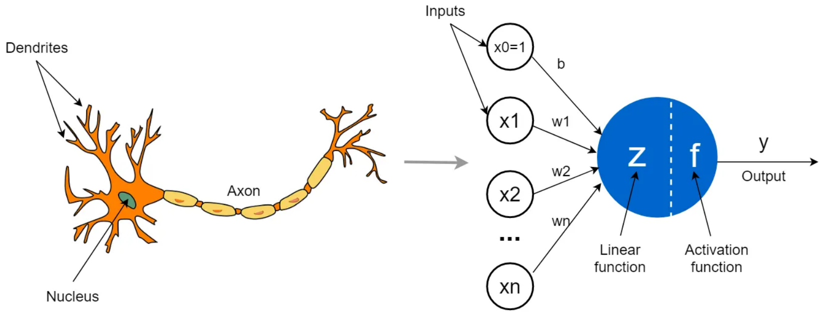

Figure 2.1: Visual comparison between a biological neuron and a perceptron (Source [5]).

signal, a perceptron in an ANN passes the output value onto the subsequent perceptrons.

Generally speaking about the structure, it consists of several layers, but essentially is possible to distinguish three different groups: an input layer, one or more hidden layers, and a final output layer. Each one of those contains multiple neurons, that represents individual nodes within the network structure. As previously explained, each node receives signals, or real numbers, processes them, and generates an output to be passed onto the next nodes. Each node and edge in the network is associated with weights and biases. During each iteration in train phase, these parameters are constantly updated to enhance and fine-tune the learning process.

2.2 Convolutional Neural Networks

One of the best-known classes of Deep Learning models, inspired by the organization of human vision, are the convolutional neural networks (CNNs). Their peculiarity is that each neuron has a narrow receptive field compared with total visual field, as occurs similarly in our visual cortex. As described above, this type of network consists of input layers, several hidden layers and an output layer. Of these layers, the hidden one typically consists of pooling, convolutional and fully connected layers. Figure 2.2 summarizes a schema of the different layers in a neural network.

One of the central elements in a neural network is the *convolutional layer*.

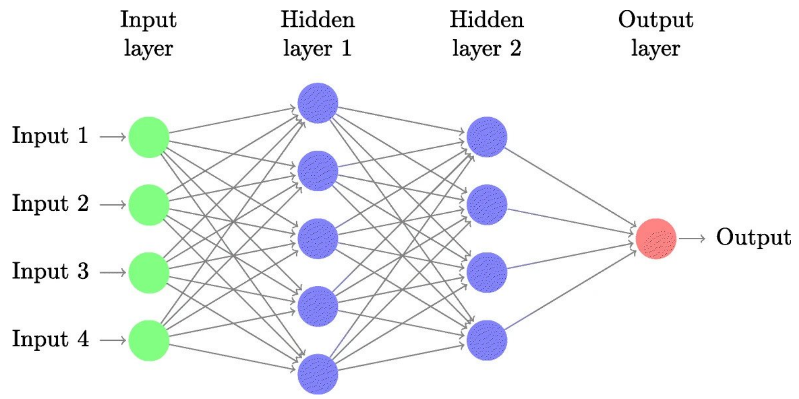

Figure 2.2: Artificial Neural Network (source [6]).

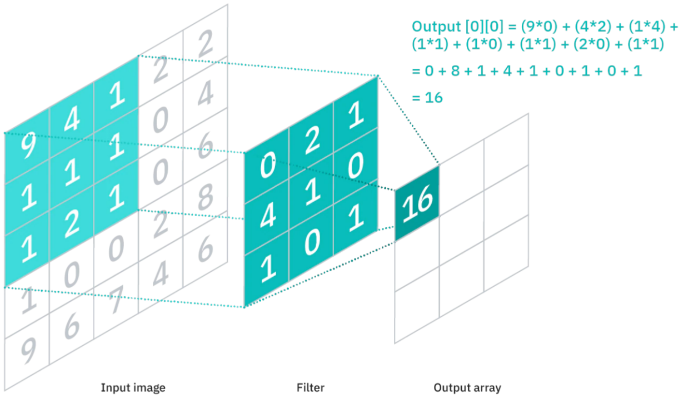

Figure 2.3: Convolutional layer (source [7]).

The main purpose of the network is to extract higher-level features from the input images. In the early hidden layers it is used to detect low-level features such as edges and intensity changes, while in the deeper layers it detects the higher-level features. The input image thus is convolved with a kernel (also known as filter), whose size can vary as the layer varies. This is typically much smaller than the size of the input image. Figure 2.3 shows an example of 3 × 3 filter convolution and moves over the entire image, even out of bounds. The application of a convolutional layer produces an activation map that describes the location and strength of a given feature in the input.

A *pooling layer* serves instead to decrease the spatial size of the feature maps by summarizing its content. The two most common techniques are average and max pooling. They respectively summarize the content of a patch with its average or maximum value.

In a typical scenario, a CNN ends with a *fully connected layer* in order to learn non-linear combinations of these features produced by the convolutional layers.

Each convolutional layer terminates with an *activation layer*, that improves the learning process from the data. Different types of activation functions are employed in hidden layers and output layers. The most common non-linear activations are: Softmax, Sigmoid, Rectified Linear Unit (ReLU).

2.3 Training Process

The training process of a neural network is a series of steps in which a model learn from input data from a dataset. It can be summarized in two main steps::

- *Feed-forwarding*: also known as forward propagation, is a process that involves passing the data through the neural network propagated forward, layer by layer, until it reaches the last layer that generates an output. Each layer of the network performs a transformation operation on the input data using the weights associated with the connections between the neurons. In essence, the feedforward process calculates the expected output of the neural network for a given input. Once the output is obtained, the loss is calculated using an objective function with respect to the desired output.

- *Back-propagation*: is the process in which the error just calculated is propagated backward in the neural network, in the opposite direction to forward propagation, from the final layer to the inner layers. The backpropagation method is employed to compute the gradient of the loss function in relation to each weight. It does this by using the chain rule and working layer by layer. Backpropagation estimates the gradient by iteratively progressing from the end to the beginning of the network, ensuring that intermediate terms in the chain rule are only computed once.

During training, the data are typically grouped into batches. A *batch* is a set of data taken from the training set that are processed simultaneously by the neural network. This iterative process is carried out over several epochs, and an *epoch* is a full pass over the entire dataset. The goal is to repeat this operation several times to allow the neural network to learn from the data repeatedly, gradually improving its performance. A batch could have different sizes, and it indicates how many training examples are used to compute the weight updates. A larger batch size can lead to greater stability in the training process, but it also requires more computational resources.

The *optimizer* is an algorithm used to adjust the weights of the neural network based on the gradients calculated during the backward propagation phase. Common optimizers include the Stochastic Gradient Descent (SGD) algorithm, adam, adamW, RMSprop, to name a few. Each optimizer has its own specific features and adjustments for controlling network learning.

The *learning rate* is used to control how quickly the weights are updated during training. It is a very important parameter to choose because it determines how quickly the neural network will converge to an optimal solution. A learning rate that is too high can cause oscillations or jumps in the search for optimal weights, never converging to the minimum, while a learning rate that is too low can slow down the training process considerably, without achieving optimal results. Figure 2.4 summarizes the different scenarios of changing the learning rate.

The *weight decay* is a regularization technique used in neural networks to prevent overfitting. It works by adding a penalty to the loss function, proportional to the size of the model's weights. This encourages the network to keep smaller weights, leading to simpler models that generalize better to unseen data. In essence, weight decay helps to balance the model's complexity and its learning capability.

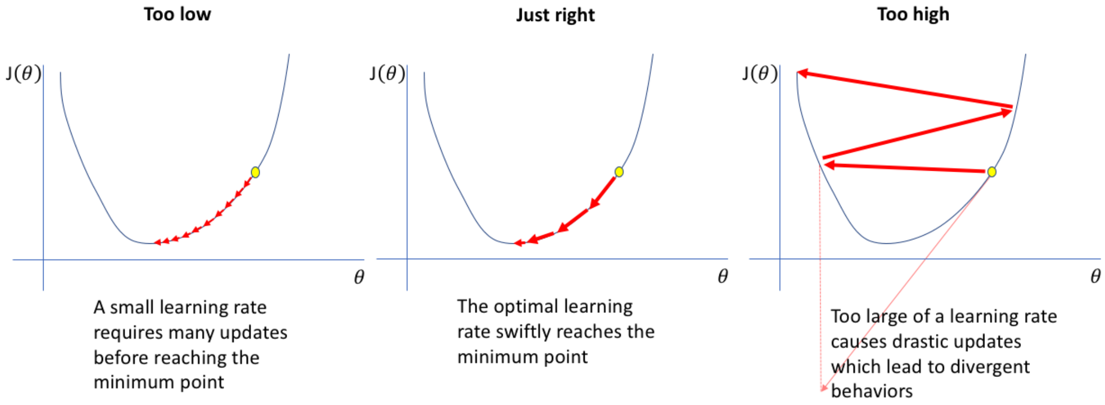
Too low
Just right
Too high
J(θ)
J(θ)
J(θ)
θ
θ
θ
A small learning rate requires many updates before reaching the minimum point
The optimal learning rate swiftly reaches the minimum point
Too large of a learning rate causes drastic updates which lead to divergent behaviors

Figure 2.4: Learning rate scenarios (source [8]).

2.4 Semantic Segmentation

Semantic segmentation is the task to classify each pixel of an image into one of predefined classes. This computer vision approach is particularly important in many applications such as autonomous driving, medical diagnostics, satellite image analysis [9, 10, 11].

The best known architectures are those based on the U-Net [11] and its variants, such as U-Net++ [12]. Initially developed in the biomedical field to segment medical images, they become over time a standard in the field of semantic segmentation. These models use an encoder-decoder structure to capture contexts at various levels of detail and reconstruct an output image with the same resolution as the input image. In particular, U-Net relies on the use of skip connections to transfer the detailed information from the encoder to the decoder, which helps to maintain fine details in the segmented image.

Another type of architectures that have been proposed are those based on DeepLab architecture [13]. DeepLab introduces an approach called dilated/atrous convolution, which is particularly effective for handling ambiguity of object contours in images. This method allows increasing the receptive range of the convolution operation without increasing the number of parameters. Then the model was extended in DeepLabV3+ [14], introducing an encoder-decoder architecture in combination with a module called Atrous Spatial Pyramid Pooling (ASPP), which allows the model to capture details at different scales. These models have found wide application in autonomous driving field [15].

Another influential architecture is the Pyramid Scene Parsing Network (PSPNet) [16]. Originally designed to address scene parsing tasks, PSPNet has also been widely applied in the field of semantic segmentation due to its ability to capture intricate details at multiple scales. The model employs a pyramid parsing module to aggregate context information at different levels of a feature hierarchy. By integrating features under different sub-region scales, PSPNet manages to recognize objects at different scales and capture global contextual information.

A recent development introduced in this field are Transformers-based networks for semantic segmentation. These networks, also known as Vision Transformers (ViT) [17], get inspiration from the Transformer architecture [18], originally designed for natural language processing, and then adapted for the computer vision task. The main idea is handling long-range relationships between pixels in images, to address the problem of CNN which analyzes only nearby pixels.

2.5 Burned Area Delineation And Severity Estimation

Semantic segmentation also finds a wide application in satellite image analysis. In particular, it is used to analyze terrain characteristics for tasks such as: land cover classification and delineation of areas affected by natural disasters or calamities such as wildfires and floods.

Over time, different methods are developed to analyze fires using satellite imagery analysis. Many of these methods take advantage of the variation in various wavelengths of light between pre- and post-fire. In Figure 2.5 is shown how the reflectance changes between the healthy and burned vegetation.

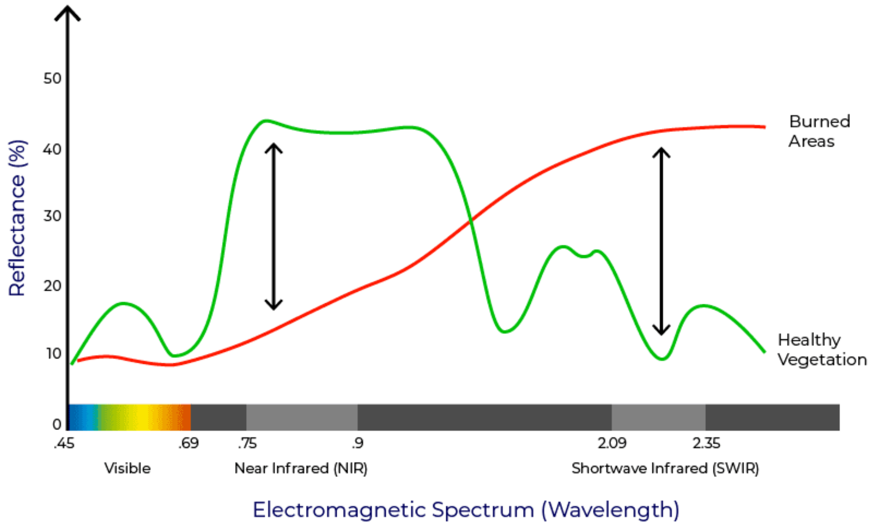

Figure 2.5: Burned areas reflect other wavelengths more strongly than vegetation and vice versa. Image source [19]

The extent of destruction caused by wildfires is typically assessed using two different indices. The choice of index depends on the data collection method – either manual acquisition via on-the-ground inspection [20], or through the use of remote sensors such as UAVs or satellites [21].

The Composite Burned Index (CBI) [20] is a measure that assesses the severity of damage caused by a fire. It is computed by considering various detailed factors, such as the condition and color of the soil, the extent of vegetation or fuel consumed, the regrowth of burned plants, the presence of new species colonizing the area, and the visible effects of scorching or blackening on trees. All those aspects are combined to create the CBI, which provides the most accurate estimation of the severity of damage. However, obtaining this index is a labor-intensive process as it relies on manually collecting data. This makes it particularly challenging and costly to apply on a large scale, especially for extensive regions. While the CBI offers a more nuanced understanding of the damage compared to other methods like the EMS, its computation becomes impractical when a significant amount of data needs to be collected manually.

Imagery obtained from remote sensing allows easier identification of fires. This is due to the specific spectral wavelengths of light that are particularly reactive to water, vegetation, and inert substances. Indices such as the Normalized Burn Ratio (NBR) further facilitate this process. It is possible to estimate the variations in soil conditions by calculating the difference NBR values from images of the same area taken at different times (before and after the wildfire). This kind of difference, named delta Normalized Burnt Ratio (dNBR)[22], is widely recognized as a reliable index for estimating the severity of burned regions. However, the thresholds set for determining severity levels can differ between image captures and might be contingent on the type of soil.

Similarly, the calculation of the delta Normalized Difference Vegetation Index (dNDVI) provides a measure of vegetation density change between two time periods, used to assess the vegetative cover over a specific area. The dNDVI is calculated as the difference between pre- and post-fire NDVI values, therefore this index measure the change over time of the vegetation quantity and quality. A negative dNDVI means a decrease of the vegetation, which could be caused deforestation or a wildfire.

In general, state-of-art methodologies utilize dNBR and other indices derived from spectral bands gathered from satellites. These are typically used either for computing thresholds or implementing machine learning approaches. Most of these techniques rely on the comparison of satellite images taken before and after wildfires, which could be not always feasible. Therefore, satellite images often contain a high level of noise, due to atmospheric conditions, which can make it difficult to calculate these indices. This implies that these methods are not universally applicable but depend on the region under consideration.

In recent years, various convolutional neural network (CNN)-based architectures have been developed for classification and segmentation from high-resolution satellite data. Various paper and researches architectures have been tested for fire detection.

Double-step U-Net [23] implements a CNN composed of a series of two U-Net models linked together, one for each tasks: the “Binary Classification U-Net” and the “Regression U-Net”. The first U-Net creates a mask of fire delineation and the second one infers on the resulted mask to make a regression on different severity damage levels.

A possible use of transformers for delineating the burned area is proposed by Rege Cambrin et al.[24]. In this paper, a SegFormer is successfully deployed to analyze the delineation of the burned area only using post fire image.

Another relevant example is developed using a Deep Siamese Morphological Neural Network (DSMNN-Net) [25], a network that takes as input both the pre- and post-fire image. It outputs the severity of the burned area.

Chapter 3

Dataset

This chapter is focused on the description and creation of the dataset. Specifically, in the first part shows the sources and the services used to download the data needed to create the dataset, while the second part describes how the data are reviewed and preprocessed.

3.1 Data Sources

3.1.1 Copernicus EMS

Considering the necessity to monitor climate change and natural disasters, the European Union decided to establish the Copernicus program: a program dedicated to Earth observation and environmental monitoring. Using a series of Sentinel satellites, Copernicus acquires and provides accurate and timely information on issues regarding climate and the environment.

Copernicus EMS (Emergency Management Service)[26] is a service part of the Copernicus program, which is primarily related to emergency management with a particular focus on catastrophic events or natural disasters.

In Rapid Mapping, several types of disasters are monitored: floods, volcanic eruptions, earthquakes, storms, and fires. Each monitored event is identified by a unique code formatted as EMSRXXX, where XXX is the specific identifier of the event.

In each event, also called activation, could be present one or more areas of interest (AoI): each area is defined within a bounding box that covers the area damaged or affected by the catastrophic event. Different types of product [27] may be available for each activation; those considered in this thesis are:

- *First Estimation Product* (FEP): It provides a very fast, but not very detailed, early assessment of the affected locations within the AOI. The assessment is derived from the first available post-event imagery. In some cases the fire is still ongoing. This product can be used to provide a first estimate of the affected area and to better define the requirements for the following more detailed products.

- *Delineation product* (DEL): It describes the impact of the event over the area of interest, showing the affected area and the affected assets, without providing a measure of the level of damage. In this case the fire event is ceased.

- *Grading product* (GRA): It provides information on the level of severity of the damage, its spatial distribution and extent. These products are derived from post-event satellite imagery and include the level, magnitude or damage grades concerning a specific disaster type. If possible, the products publish relevant and up-to-date information specific to the affected area, such as settlements, transport networks, industries, and others. The damage levels provided range from 0 (not burned area) to 4 (completely destroyed)

The activation's products provided by Copernicus use two different damage assessment conventions [28]: EMS-98 and Copernicus EMS. The EMS-98 severity level ranges from 0 (unburned sub-area with *No damages*) to 4 (*Completely Destroyed* sub-area). Intermediate values are used to represent the following situations; *Negligible to slight damage*, *Moderately Damaged* and *Highly Damaged*. The intermediate values in Copernicus EMS severity scale are *Possibly Damaged* and *Damaged*.

In the Table 3.1 are summarized all the damage levels, that is the actual mask index, and relative mask color. These classifications are used by emergency services and those estimating the extent of damage caused by a disaster and for planning subsequent relief, recovery and reconstruction operations. In the figure 3.1 a sample is reported as qualitative example of the difference type of masks.

Every activation map available on the Copernicus site is saved in the shapefile format, which is a data format for storing geospatial information. For this thesis, only the GeoJSON files are considered. Within them, the geometric information is stored as polygons and multipolygons. A polygon consists of a list of points that

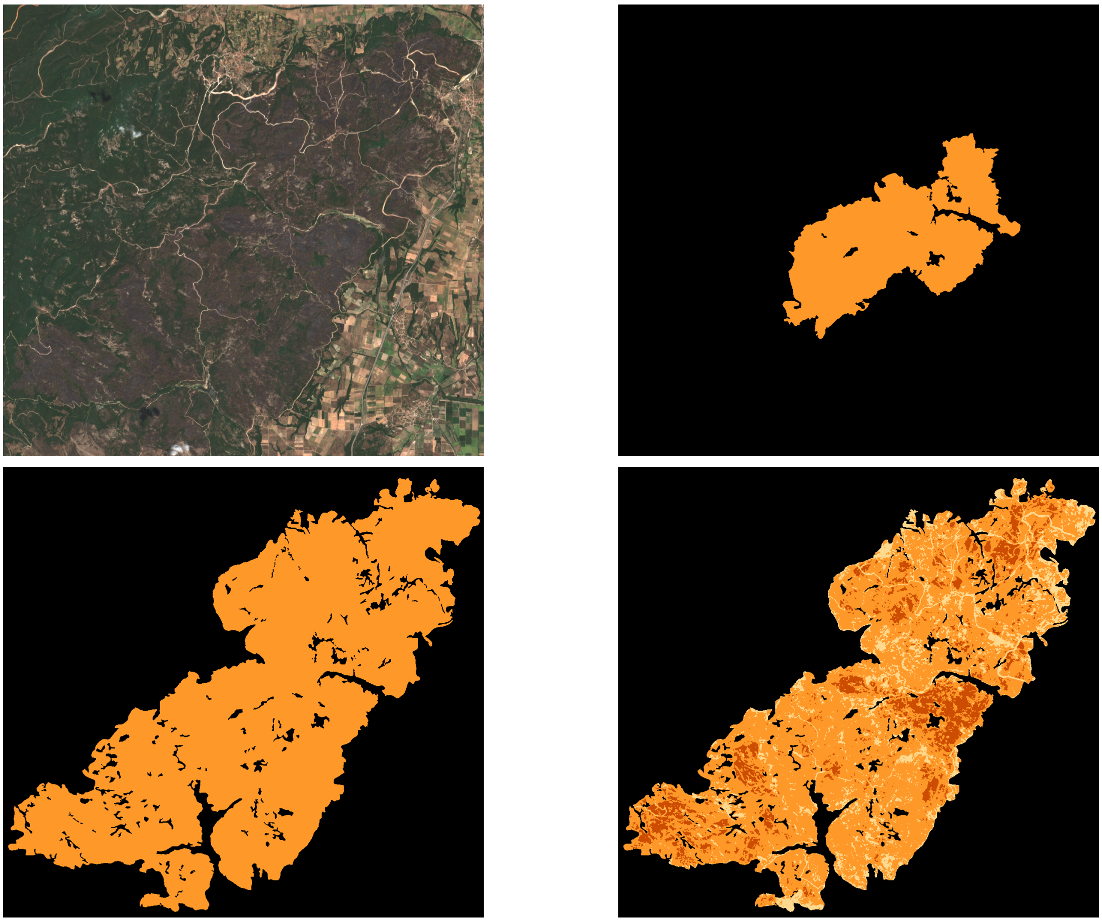

Figure 3.1: In clockwise direction: the Sentinel2 image with level L2A, FEP mask, DEL mask and GRA mask.The FEP mask is incomplete because it was made during an earlier stage of the fire.

constitute the vertices of the same, a multipolygon on the other hand consists of a list of polygons.

In particular, GeoJSON files with the following formatted string were considered for each activation:

- `EMSRXXX_AOIYY_TYPE_PRODUCT_areaOfInterestA.json`, where TYPE can be ”GRA”, ”DEL” or ”FEP”, defines the AOI YY of that particular activation XXX where the event happened.

- `EMSRXXX_AOIYY_TYPE_PRODUCT_observedEventA.json`, where TYPE can be ”DEL” or ”FEP”, defines the multipolygons geometry for the wildfire

Damage	**Copernicus EMS**	**EMS-98 Classes**	**Pixel value**
0	No visible damage	No visible damage	⬛
1	-	Negligible to slight damage	🟩
2	Possibly damaged	Moderate damage	🟨
3	Damaged	High damage	🟧
4	Destroyed	Destruction	🟥

Table 3.1: Severity level and relative tag names in both damage assessment convention and relative color of the mask

delineation for an AOI YY of the activation XXX.

- `EMSRXXX_AOIYY_GRA_PRODUCT_naturalLandUseA.json` defines the various multipolygons geometry for the grading damage levels for an AOI YY of the activation XXX.

A validity mask was created and stacked in each mask. It is used to bound the mask in case the Sentinel-2 image inside the given bounding box turns out to be clipped, filling in the NO Data blanks points with value 255. This geospatial information obtained from the files are the ground truth mask in the training process of the segmentation model.

3.1.2 Sentinel-2

Sentinel-2 satellites provide 10 meter high-resolution images in the visible and infrared bandwidths. The data collected are particularly used for ground and vegetation monitoring and detection of burned areas.

Sentinel-2 service can provide two kinds of products: Level-1C and Level-2A. Level-1C provides raw data, while Level-2A is generated by applying an algorithm for atmospheric reflectance correction on Level-1C products, resulting in an orthoimage Bottom-Of-Atmosphere (BOA) corrected product. The applied correction reduces the noise brought by natural conditions, like air turbulence and fog, and the influence of aerosols, producing a more qualitative image which highlights ground information.

Sentinel-2 provides a total of 12 bands, each with a different resolution and property, that are useful for monitoring vegetation and burned areas. Table 3.2

summarizes all 12 bands and relative description of information carried inside. In addition, Sentinel-2 offers higher resolution images than other missions such as MODIS and Landsat.

Bands	**Description**	**Wavelength (nm)**	**Resolution (m)**
B1	Ultraviolet	443	60
B2	Blue	490	10
B3	Green	560	10
B4	Red	665	10
B5	Red Edge 1	705	20
B6	Red Edge 2	740	20
B7	Red Edge 3	783	20
B8	Near Infrared (NIR)	842	10
B8A	Red Edge 4	865	20
B9	Water Vapour	945	60
B10	SWIR - Cirrus	1375	60
B11	SWIR 1	1610	20
B12	SWIR 2	2190	20

Table 3.2: Sentinel-2 bands description

SentinelHub is an online service that provides access to Earth observation data from many sources, including the European Union Sentinel satellites. In particular, for this thesis, Sentinel-2 L2A data are collected using this service.

All images were downloaded from the SentinelHub service with a size that ranges from a minimum of 512 to a maximum of 2048 pixels.

In each query to SentinelHub[1] the following parameters are needed:

- the bounding box of AOI, each of them consists of two tuples of coordinates <Longitude, Latitude>, which indicate the top-left and bottom-right edges of the area.
- the percentage of clouds in the Sentinel image, the max value is set to 10% of the cloudiness of the Sentinel-2 image. Despite this, some images were still cloudy, so a manual search were done, trying to improve the general quality.
- the time interval in which to search for available Sentinel-2 images within that AOI. For each activation, satellite acquisitions were requested in a time

[1] The images were downloaded using the APIs provided by SentinelHub

interval from a few days after the activation date to 30 days after. In any case, the focus was to find images closer as possible to the date of the event, if available. For some activations, it was not possible to download images even earlier than 30 days after the event.

- the type of product to download, in this case L2A.

3.1.3 Land cover

In addition to the masks just created, the dataset is expanded to include also land cover masks. These masks contain information about the type of land cover for those pixels. This can be useful in a variety of applications, such as monitoring land use changes and assessing the impacts of natural disasters such as wildfires. Since the taxonomy of land cover classes is not unique, the maps are collected from several dataset: ESRI 10m Annual Land Use Land Cover with 9 classes [29], ESRI 2020 Global Land Use Land Cover with 10 classes [29] and Esa WorldCover 2020 [30].

ESA WorldCover 2020 is a land cover map produced by the European Space Agency (ESA) using Sentinel-2 images of the year 2020. It provides detailed information about 12 different land cover classes, including forests, croplands, grasslands, wetlands, water bodies, urban areas, and bare soil, among others.

In all land cover masks, the label value is shifted down, so the value 0 generally associated to No-data is changed to 255.

3.2 Data Processing

After the creation of the mask, each image is inspected to evaluate the general quality of the mask and Sentinel-2 image. Specifically, the criterion considered are:

- presence of clouds or smokes in the Sentinel-2 image, especially on the affected part. The overall noise quality of the image is evaluated visually.
- errors in the Sentinel image, such as wrong values and distortion inside.
- the burned part is fully included within the validity mask generated.
- the mask actually overlaps perfectly the burned area through a visual inspection. In some cases, minor errors in the mask are corrected.

Most AOIs have been cropped from the original size provided by Copernicus to avoid excessive imbalance between the burned and unburned areas. This will be relevant later when the RandomBatchGeosmapler will be addressed in Section 5.1.2. Furthermore, with image cropping it is possible avoiding unwanted presence of past fires, prior the event in question and therefore not present in the mask provided by Copernicus.

3.2.1 Cloud Masking

As previously mentioned, a common problem that might arise in the analysis of satellite images is the presence of clouds and smokes. In fact, clouds can obscure important soil features and cause errors or inaccuracies in image analysis.

Not for all AOIs is it always possible to find an image with clear skies. So to tackle this problem the clouds should be masked, through a process of identifying them and their shadows.

Previous work [31] has often used Sentinel band analysis, through a threshold scheme to detect the reflectance generated by clouds, and then identify the cloudy area. However, this approach is not without challenges. Clouds could have a variety of shapes, sizes, colors and textures, which makes precise identification difficult. In addition, it could be difficult to distinguish cloud shadows from terrestrial features, such as bodies of water.

In this thesis, the CloudSen12 model has been used[32], a semantic segmentation model for cloud identification. It is developed and trained from Sentinel-2 images and the source code is freely available on GitHub [2].

CloudSen12 was trained using various accurately labeled images, and then each result was validated by several other people before being reintroduced into the training process. In this way, the labeling errors on the original training set were greatly limited. Through this model, a mask was obtained, which will later be useful for removing or ignoring image areas covered by clouds or clouds' shadows.

The output mask of CloudSen12 has 4 different levels. Each level indicates the probability that a given pixel belongs to that class. The levels are:

- Layer 0: indicates the prediction in percentage of clear sky.
- Layer 1: indicates the prediction in percentage of heavy and thick clouds.

[2] https://github.com/cloudsen12

- Layer 2: indicates the prediction in percentage of light clouds/smoke/fog. This layer was not considered later, because it is subject to major noise and false positive predictions.

- Layer 3: indicates the prediction in percentage of shadow of clouds.

Table 3.3 summarize all the different levels and relative color in the mask. Figure 3.2 reports a sample of the output of CloudSen12.

Output layer mask	**Description**	**Pixel value**
0	Clear sky	🟦
1	Thick clouds	⬜
2	Thin clouds/fog/smoke	▩
3	Cloud's shadow	⬛

Table 3.3: CloudSen12 levels and color in the mask

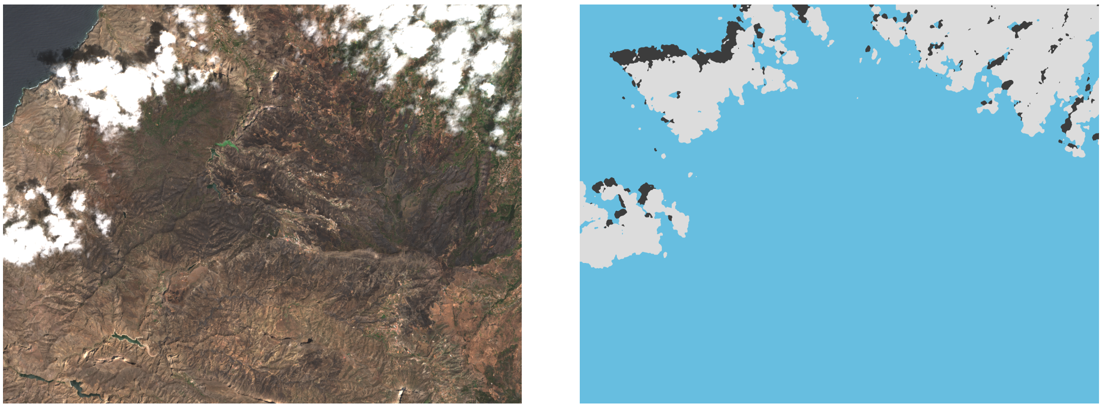

Figure 3.2: Sample output from CloudSen12

3.3 Dataset information

The dataset comprises all activation available on Copernicus under the tag wildfire from March 2017 to April 2023. In total 243 AoI distributed mainly in Europe are downloaded from Copernicus. The total number of images correctly preprocessed are 330 with a dimension that ranges from 512 to 2048 pixel per side.

Figure 3.3 shows all the areas in Europe area affected by forest fires during the time interval considered in this thesis. The circles determine the position of the fires considered, while the diameter of the circle is in a logarithmic way proportional to the size of the affected area. Other activations are distributed among Australia, Mexico and Chile.

Table 3.4 shows the imbalance distribution of labels inside the dataset, since most of the pixels belong to the unburned class. It is obvious that passing from the binary case to the multiclass problem, the complexity of the problem increases. In fact, class imbalance becomes even more evident when considering the severity level, as reported in Table 3.5.

Classes	**Distribution (%)**	**Occurrence**
0 - Unburned	78.83	385010425
1 - Burned	21.17	103310515

Table 3.4: Class imbalance in binary case

Classes	**Distribution (%)**	**Occurrence**
0 - Unburned	78.83	385010425
1 - Negligible damage	0.86	4192020
2 - Moderate damage	4.82	23538904
3 - High damage	9.94	48541862
4 - Destroyed	5.55	27129749

Table 3.5: Class imbalance in multi-class case

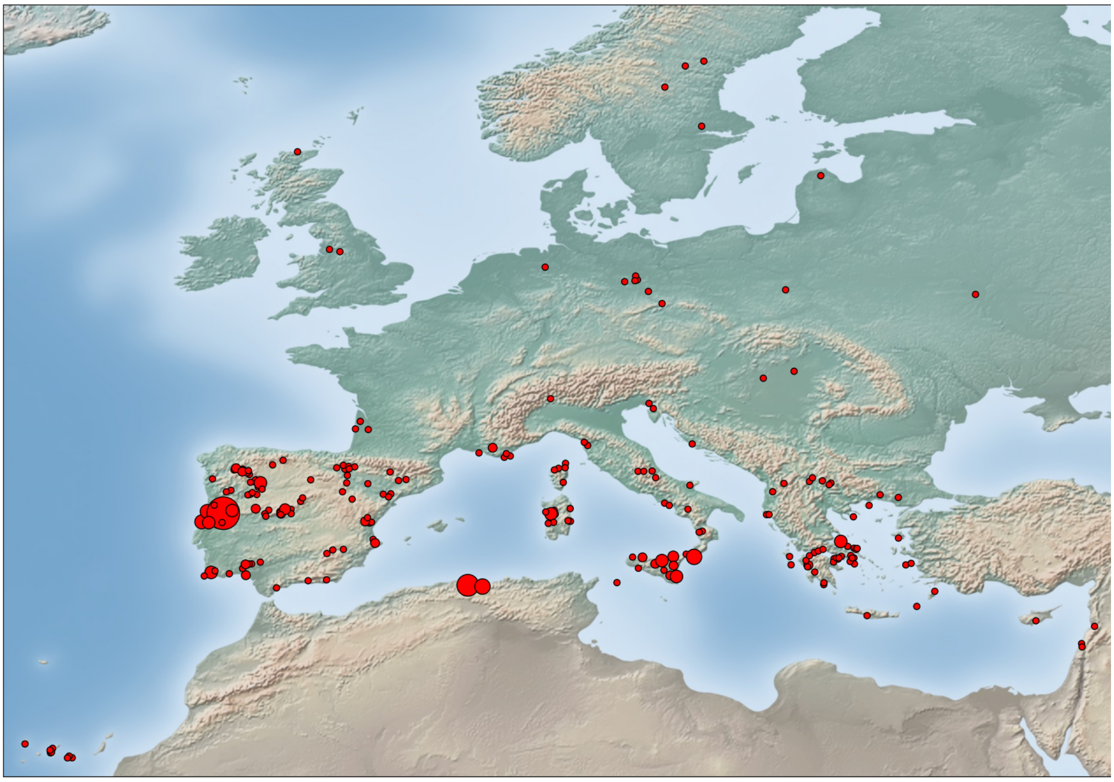

Figure 3.3: Distribution of the activations on European region, the size of the point is logarithmic proportional to the fire extent

Chapter 4

Methodology

In this chapter, the multitask approach is presented alongside with the details for each of the architectures tested.

4.1 Problem statement

This thesis is focused on delineating the burned area and estimating the severity of damage caused by wildfires in different sub-regions of a wildfire-affected area at the same time. A multitask learning approach in semantic segmentation is proposed jointly to address these two tasks, aiming to improve the efficiency and accuracy of wildfire management efforts. Given a post-fire Sentinel-2 12 bands L2A satellite image of an area affected by a wildfire, the goal is to classify the probability of whether an area is burned or not and predict a continuous value between 0 and 4 for each pixel in the post-fire image. This value approximates the corresponding values on the Copernicus EMS damage severity grading map, where severity levels are represented by whole numbers within the same range. The problem is configured in two sub-tasks: a binary classification and a regression task. The first task is to predict a binary target variable that defines the burned area, while in the second task, the target variable is a numerical feature, which is used to represent ordered severity values.

4.2 Multitask approach

Multitask learning is a deep learning approach that involves training a model on multiple related tasks at the same time. The goal is to exploit common relationships and information between tasks to improve the effectiveness and accuracy of the model [33].

In the context of semantic segmentation, multitask learning can be used to train a neural network model to perform multiple segmentation tasks simultaneously. In other words, instead of training a separate model for each type of segmentation task, a single model can be trained to perform all these tasks jointly.

In this thesis, the model was trained to perform two main tasks: delineation and severity estimation of burned areas. For each Sentinel-2 image input, two output masks are obtained. The first one, delineation mask, is used to identify and delineate the areas involved in the fire. The second one, grading mask, assesses the severity of damage to the affected region.

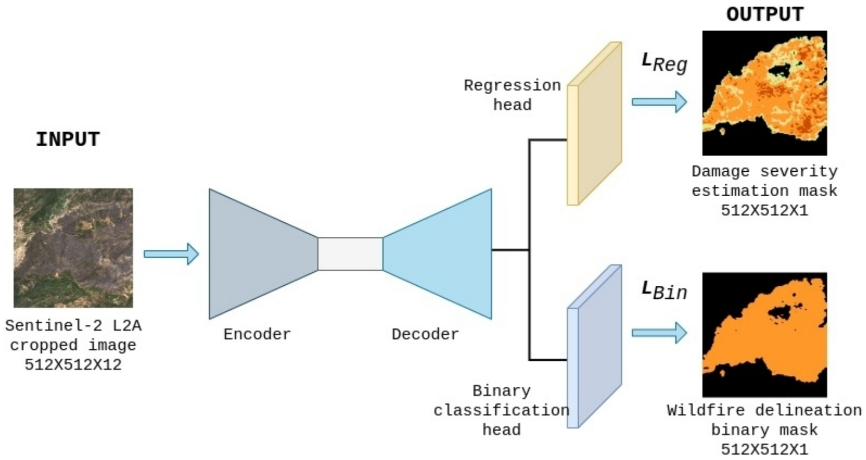
OUTPUT
INPUT
Regression head
L_{Reg}
Damage severity estimation mask 512X512X1
Sentinel-2 L2A cropped image 512X512X12
Encoder
Decoder
L_{Bin}
Binary classification head
Wildfire delineation binary mask 512X512X1

Figure 4.1: The architecture of the multitask segmentation model with the two segmentation heads

To do so, an additional segmentation head was added to the segmentation model. This is the final module of the network that produces the output. It is responsible for processing the features and producing a segmentation map in which each pixel in the image is classified into a certain category. For the binary segmentation head, the output is a probability of each pixel to belong to burned or unburned class. On the other hand, the segmentation head for regression is responsible for the damage severity task, predicting a continuous value for each input.

These heads consist of one or more 2D convolutions with 1×1 kernels, followed by an up-sampling operation to bring the image to its original size. Then, its task is to refine and detail the information learned from the model backbone to produce an accurate and detailed segmentation map. A schema of the multitask model is showed in Figure 4.1.

This approach present some advantages: an improvement of the computational efficiency, since multiple model tasks can be performed with a single model and joint training on multiple tasks allows learning more general and efficient representations of the data, potentially improving performance on each individual task.

4.3 Network Architectures

For this thesis, different architectures and models are implemented and tested with multitask semantic segmentation. As briefly mentioned in Chapter 2, the networks are: U-Net, U-Net++, DeepLabV3+ and PSPNet.

4.3.1 U-Net

U-Net is a convolutional neural network (CNN) that has been designed originally for biomedical image segmentation.

The name "U-Net" comes from the U-shape of the model, which is symmetrical with respect to the central axis. This shape consists of two main parts: the contractive path (encoder) and the expansive path (decoder). This family of architectures is called encoder-decoder structure.

The encoder or contractive path is designed to extract the salient features of the input image. It consists of a series of convolution blocks with 3 × 3 kernels, each followed by a max pooling operation for downsampling. Each convolution block consists of one or more convolutional layers followed by an activation function, such as ReLU (Rectified Linear Unit). The goal of the encoder is to progressively reduce the size of the input image but increase the number of channels of the extracted features.

The decoder or expansive path, on the other hand, aims to reconstruct the segmented image from the encoder output. Each step of the decoder consists of an up-sampling operation (increasing image size) followed by an up-convolution operation with 2 x 2 kernels, which halve the number of total channels. In addition, during up-sampling, data from the encoder corresponding to the same scale are also concatenated to allow detailed information during reconstruction.

The final part of the U-Net consists of a convolution layer with 1x1 kernel, which reduces the number of channels of extracted features to the desired output size. This layer uses a sigmoid activation function to produce a segmented image in which each pixel represents the probability of belonging to a given class.

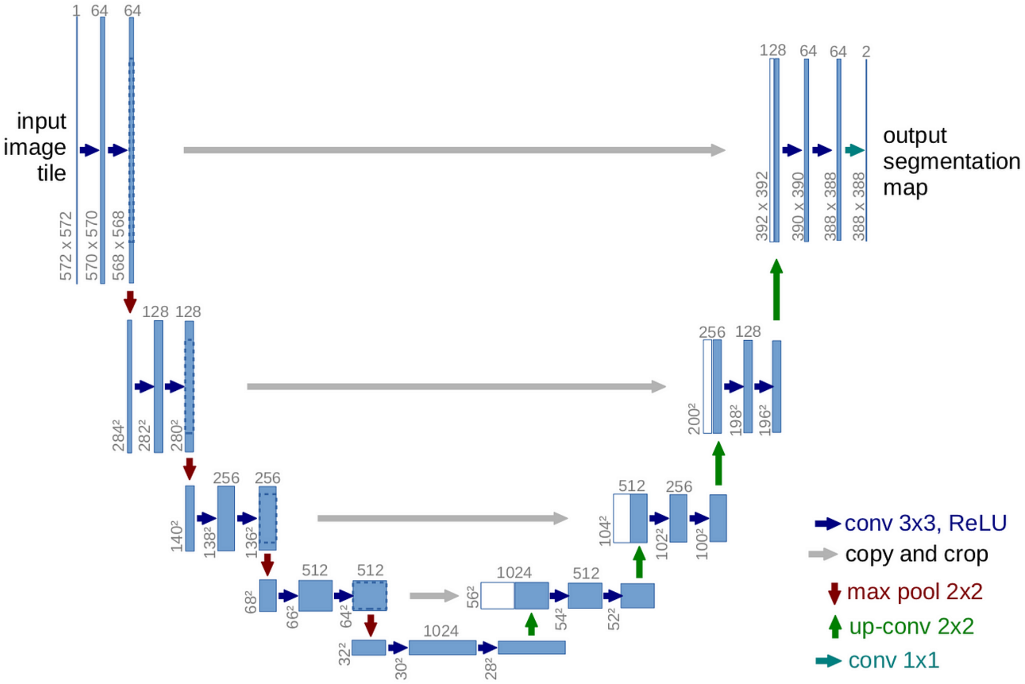

Figure 4.2: U-Net encoder-decoder structure (Source [11]).

4.3.2 U-Net++

U-Net++ is one of many variants of the U-Net architecture presented over the years. The original U-Net is improved by the introduction of a new segmentation architecture based on nested and dense skip connections between encoders and decoders. The model can more effectively capture fine-grained details of the foreground objects when high-resolution feature maps from the encoder network are gradually enriched prior to fusion with the corresponding semantically rich feature maps from the decoder network. [12]

The structure of the U-Net++ however has several changes from the standard U-Net:

- use of layered encoders: in the standard U-Net, the encoder consists of a series of convolutional layers followed by spatial dimension reduction by pooling operations. In U-Net++, encoders are organized in a tree structure, where each resolution layer has two branches working at different resolutions. This means that each level has both a downsampling path and its own upsampling path.
- use nested skip connections: in U-Net++, in addition to skip connections between encoders and decoders on corresponding levels, nested skip connections are introduced between encoder branches at different resolutions and decoders. This allows for a broader flow of information and greater integration of features extracted at different scales.
- a feature concatenation: in the decoder branches of U-Net++, features extracted from the corresponding encoder branches are concatenated, instead of simply joined as in standard U-Net. This helps to preserve detailed information at different scales and facilitate the learning of complex relationships.
- an addition of lateral connections: U-Net++ introduces lateral connections between encoders and decoders at the same resolution level. This promotes local information exchange between encoders and decoders, improving feature integration.

4.3.3 DeeplabV3+

DeepLabV3+ introduced in [14] is a semantic segmentation model part of DeepLab architecture, which integrates a decoding module to refine the boundary of the objects. The model employs an encoder-decoder structure and utilizes Atrous Convolution (known as dilated convolution) and Atrous Spatial Pyramid Pooling (ASPP) modules to efficiently capture at different scale contextual information. By exploiting the atrous convolution, DeepLabv3+ maintains high-resolution feature maps, thereby preserving detailed information important for accurate segmentation.

The ASPP module is designed to handle objects at various scales. It combines image-level features and multiple levels of context by applying atrous convolution

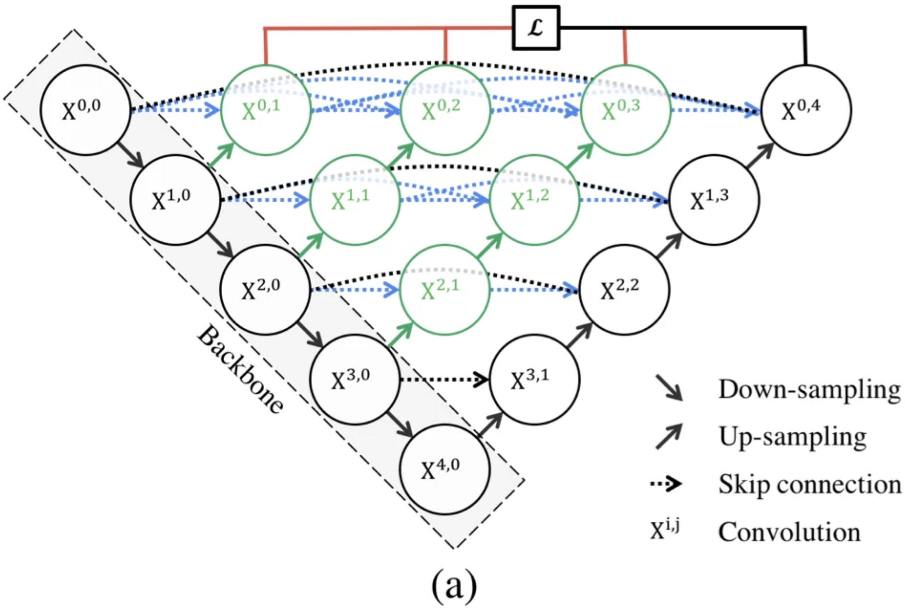

Figure 4.3: U-Net++ consists of an encoder and decoder that are connected through a series of nested dense convolutional blocks (Source [12]).

at several rates, resulting in a map of aggregated feature. This innovation enables DeepLabv3+ to better manage images with objects of different scales, enhancing its versatility and effectiveness.

In addition, DeepLabv3+ introduces a simple decoder module to refine the segmentation results, particularly along object boundaries. This refinement yields more accurate and visually pleasing segmentation results. Moreover, the network incorporates depthwise separable convolution as an alternative to conventional strided convolution, decreasing the computational cost and the number of parameters, leading to more efficient model training and inference. First applying a convolution on the individual input channels, and then combining the results using a 1x1 convolution. This reduces the amount of computation required without losing the ability to capture image features.

Another key component of DeepLabV3+ is the decoding module. After extracting features from the image using the dilated, depth-separable convolution, the decoding module uses a 1x1 convolution to reduce the number of channels, followed by a bilinear up-sampling operation to increase the image size. The decoding module then combines the low- and high-resolution features to achieve

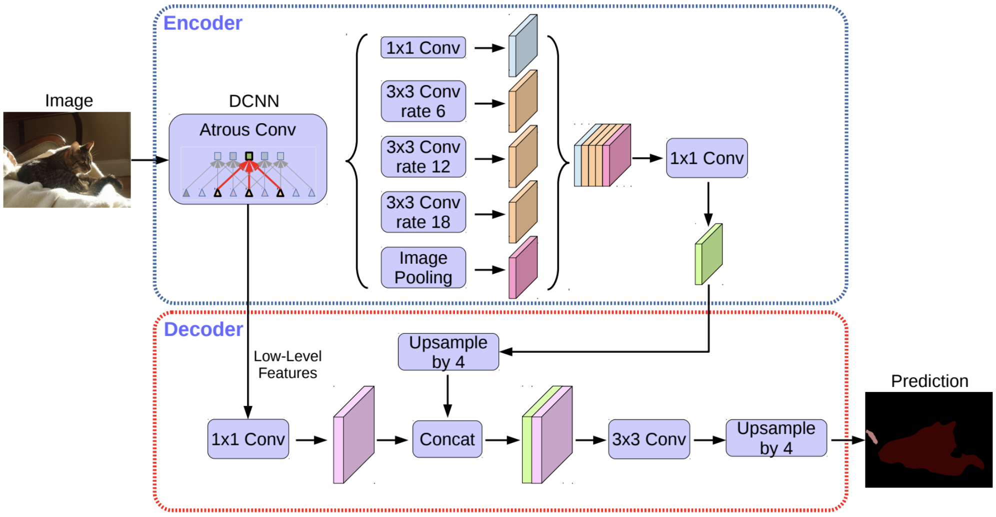

Figure 4.4: DeepLabV3+ architecture with ASPP module (Source [14]).

more accurate segmentation.

4.3.4 PSPNet

The Pyramid Scene Parsing Network, or PSPNet, as introduced in [16], is a segmentation model based on the idea that for a comprehensive understanding of the scene, it goes beyond the focus of local details. It necessitates a consideration of the overall image context as well.

Therefore, PSPNet incorporates a component called the Pyramid Pooling Module, or PPM, which takes as input the feature maps of an image and divides them into different regions of varying sizes. In this way, the module is able to capture information at different scales. The module calculates the average pooling for each region, thus capturing the most general information of that region. These pooled features are then upsampled and concatenated with the original feature map. This process is performed at multiple scales, allowing the network to capture context information from different-sized regions.

Furthermore, PSPNet employs an auxiliary loss during training, encouraging better learning of features in the deeper layers. The idea behind the auxiliary loss is to mitigate the problem of vanishing gradients, which can occur in deep neural networks and can make learning difficult for earlier layers. This is achieved by applying a simpler version of the Pyramid Pooling Module on a shallower layer

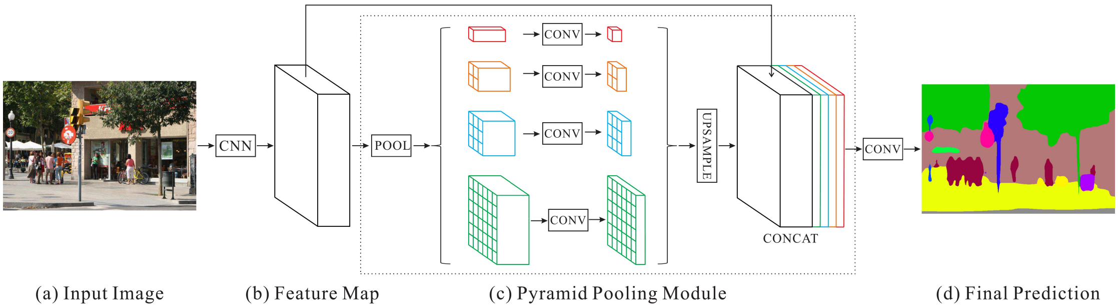

Figure 4.5: PSPNet architecture and PPM (Source [16]).

of the network and calculating a supplementary loss. This auxiliary loss helps improve the overall performance of the network, especially in the presence of small or thin objects.

This approach allows the PSPNet to maintain detailed information at the local level, thanks to classical convolutions, but also to incorporate information at the global level, thanks to the Pyramid Pooling module.

Chapter 5

Experiments

This chapter presents the configuration used to train and test the model, then a brief explanation of the losses and optimization used to solve the task is done. Finally, the experimental results are summarized and commented.

5.1 Implementation Details

Training neural networks is a complex and resource-demanding process, necessitating appropriate data preparation and hyperparameters selection. This section provides an overview of the conducted experiments, thoroughly outlining both the training and assessment methodologies employed.

5.1.1 Experimental setup

The purpose of this thesis is to evaluate the multitask approach by training and comparing the performances of the models previously described: U-Net, U-Net++, DeepLabv3+, and PSPNet. Each model is equipped with a ResNet50 encoder backbone. ResNet50 is chosen because it is best suited for the complexity of the wildfire delineation task. In the group of different residual networks, ResNet50 has a good balance between performance and computational cost, moreover more complex ResNets are prone to overfit the data. After training, each model is tasked with performing wildfire delineation and assessing the severity of fire damage, using a validation dataset that was kept separate from the training dataset.

Each model is trained considering all bands of Sentinel-2. So the network receives as input a tensor of size $512 \times 512 \times 12$ and the ground truth masks of wildfire delineation and severity estimation.

The dataset has been divided into training, validation, and test set. As reported in section 4.3, the total count of large-scale images downloaded and preprocessed are 330. For the subdivision of the dataset, we consider the same dataset employed in previous works [34], as effectively identifies a subset of our samples. The idea is to train the multitask model on activations that are not already considered in other works. In total, the test set includes 25 AOI for a total of 72 images, which is approximately of 24% of the total dataset. For the validation set, the data should be independent and identically distributed to the train and test. So the AOIs selected are distributed as much as possible between the various geographical locations and climates. A total of 38 images are considered for model validation all around Europe. The rest of the dataset is considered as training set: 220 images. Table 5.1 summarizes the size of each set, while all activations grouped in each set are listed in the Appendix Table A.1. In Figure 5.1 shows the distribution of all activations divided between the training – validation - test set.

	Training set	Validation set	Test set	Total
Number Images	220	38	72	330

Table 5.1: Subdivision of the dataset

The experiments were run on a workstation with an Intel Core i9-7940X 3.10GHz with 128GB of RAM and 4x GTX 1080Ti. Data analysis and data processing were performed through python and scikit-learn, while neural network models were developed and trained using PyTorch framework. All python packages and versions are summarized in Table 5.2.

5.1.2 Training process

The training process is an important stage in the development of machine learning models. In this thesis, different training components are employed such as a custom RandomBatchGeoSampler for sampling, AdamW optimizer for weights updates, and a Cosine Annealing scheduler with exponential decay for adjusting the learning rate.

The *RandomBatchGeoSampler* is a geosampler from TorchGeo library and it is used to manage the geographical distribution of satellite data during the training process. It provides a strategy for randomly sampling batches of data

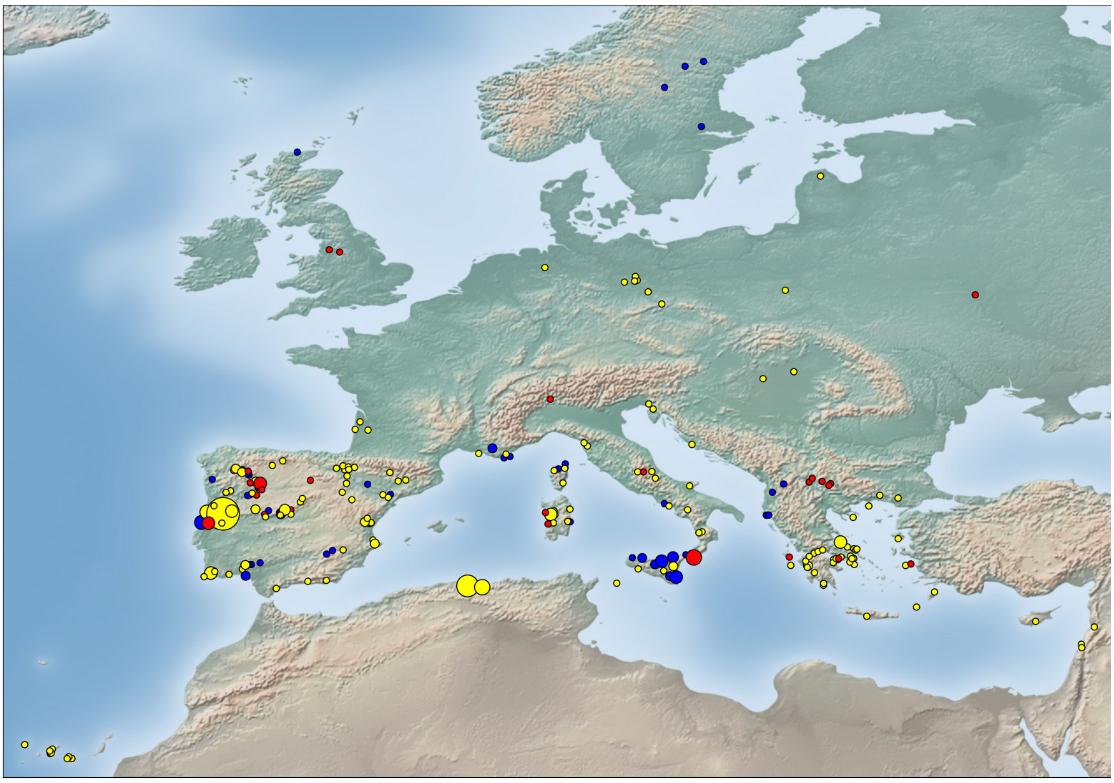

Figure 5.1: The distribution of the training (yellow dots), validation (red dots), and test set (blue dots).

in a geographically distributed manner. All tile sampled have a dimension of 512 × 512 pixels. Given that the dataset is inherently unbalanced, the sampler is constrained to store in each batch at least 60% of images that have more than 30% of burned area. This ensures that each batch has on average 40% burned area, rebalancing the label distribution. Once the tiles are sampled, the cloud mask, created as shown in Section 3.2.1, is superimposed on the tile and all pixels labeled as clouds (level 1) or cloud shadows (level 3) are set to 255. The value 255 is used as an index to ignore, which means that this particular pixel is ignored when losses are calculated on the prediction mask during training.

In all experiments, the model training iterates for 100 epochs, with each epoch consisting of 120 samples: each model is trained with 12000 sample 512 × 512.

The optimization method used in our experiments is *AdamW*, an extension of the traditional Adam optimizer. AdamW includes a weight decay parameter that provides a regularization effect, leading to more generalized models and

Library	**Version**
python	3.6.16
albumentations	1.3.0
cloudsen12	0.07
numpy	1.23.5
pandas	1.5.3
rasterio	1.3.6
matplotlib	3.6.3
scikit-learn	1.3.0
segmentation_models_pytorch	0.3.2
torch	1.13.1
torchgeo	0.4.0
torchmetrics	0.11.3
torchvision	0.14.1

Table 5.2: Python packages and versions installed

preventing overfitting. In our configuration, the weight decay parameter is set to 0.001 and the initial learning rate is 0.0003.

The learning rate is an important hyperparameter that controls how much the weights of our model are being updated during training. Therefore, choosing an appropriate learning rate scheduler is important for good results. The scheduler used in this thesis is a variant of the Cosine Annealing scheduler, with an exponential decay. The Cosine Annealing is a Cyclical Learning Rate [35] and implemented as a scheduler with exponential decay. This scheduler adjusts the learning rate according to a cosine function and an exponential function, starting with a high rate and gradually reducing it. The main idea is the cooling schedule with a warm restart to explore the features space, trying to minimize the result and escaping from local optima. This suggestion is derived from the simulated annealing optimizations. The qualitative graph of the scheduler is shown in Figure 5.2

This combination of all those training tools allows the model to converge faster in the early stages of training when the learning rate is higher and then fine-tune the weights with a smaller learning rate towards the end of the training, providing a balance between global and local exploration of the weights space.

The best combination of parameters adopted for every experiment in this work is reported in Table 5.3:

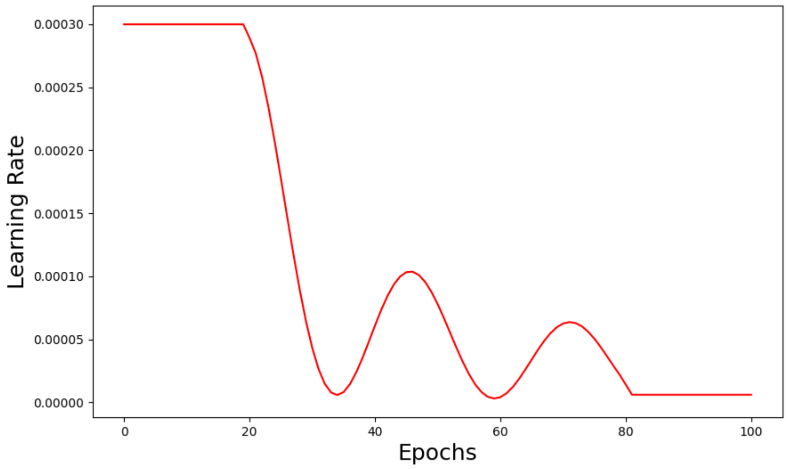

Figure 5.2: Cosine annealing with exponential decay over the epoch

Parameter	**Value**
Batch size	8
Epochs	100
Sample per epochs	240
Encoder	ResNet50
Optimizer	AdamW
Learning Rate	0.0003
Weight Decay	0.0001
Scheduler	Cosine annealing, exp. decay
Loss function	DICE

Table 5.3: Summary of all training parameters

During the training process, a data augmentation is considered. This is a strategy used to increase the diversity of available data for training a model without collecting new data. It helps to reduce overfitting by creating a more diverse set of training samples, thereby improving the model's ability to generalize to unseen data.

The Python library for data augmentation considered in this thesis is *albumentations*, which provides a wide range of techniques for image augmentation. This is the pipeline of augmentation applied during the training: flips, translation, scaling, rotations, shear, changing brightness or contrast of the image. In particular, shear is a geometric transformation applied to an image, that involves a shifting position of pixels in a given direction, thus distorting the image along a specific axis. The shear operation is performed by sliding rows or columns of pixels horizontally or vertically, creating a shearing effect. The random brightness and contrast adjusts the brightness and contrast of the images. This ensures that the model is more robust to possible variations in illumination due to atmospheric variations or different acquisition times during the day. Table 5.4 summarizes all the transformations used in the pipeline with relative values applied.

Transformation	**Probability**	**Values**
Flip	0.5	/
Translate percentage	0.5	0.2
Scaling	0.5	[0.8, 1.2]
Rotation	0.5	360
Shear	0.5	[-20, 20]
Random Brightness	0.5	0.1
Random Contrast	0.5	0.1

Table 5.4: Data augmentation parameters

All blank areas resulting from transformation are filled with reflection on the edge of the image, maintaining the continuity of important visual features. By using these different augmentation techniques, it is intended to ensure a greater diversity of possible image variations in the training set, allowing the model to increase its robustness and generalization capabilities.

5.2 Loss Functions

During training, model performance is evaluated by a loss function, which quantifies the degree to which model predictions align with ground truth. In the multitasking approach, a combination of different loss functions, chosen specifically for each task, is employed. Specifically, a Cross-Entropy (CE) or Dice loss is employed for the binary segmentation task, while a Mean Squared Error (MSE) loss is employed for the gravity estimation task.

Cross Entropy loss, also known as log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. The core of Cross Entropy loss lies in the logarithmic function, which is a key component of its formula:

$$CE = -\sum_{i=1}^{N} y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i) \tag{5.1}$$

The logarithm in the Cross Entropy formula is monotonically increasing, which means that its value increases as the input value increases. When p_i is close to 1, the value of $log(p_i)$ approaches 0, and hence the loss is small. However, as p_i gets closer to 0, $log(p_i)$ goes towards negative infinity, which increases the loss dramatically. Since the goal is typically to minimize the loss function, and the output of the logarithm for numbers between 0 and 1 is negative, adding a negative sign to the formula the graph flips upwards. As a result, the loss is positive and increases as the predicted probability diverges from the actual label.

DICE loss is essentially a measure of overlap between two samples. The DICE coefficient is defined as twice the area of overlap between the predicted and actual regions divided by the total number of pixels in both regions. This measure ranges from 0 to 1 where a Dice coefficient of 1 denotes perfect and complete overlap. Hence, DICE loss is calculated subtracting this dice coefficient:

$$DICE = 1 - \text{DICE cofficient} = 1 - \frac{2 \cdot |Y \cap \hat{Y}|}{|Y| + |\hat{Y}|} \tag{5.2}$$

where Y represents the true region, $\hat{Y}$ is the predicted region, and $|\cdot|$ denotes the cardinality or the number of pixels in the region.

For the severity estimation task, we use the *Mean Squared Error (MSE) loss.* This choice is due to the nature of the task, which is essentially a regression problem. The MSE is computed as the average squared difference between the estimated values and the ground truth. The formula is:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{5.3}$$

where N is the number of instances, y_i is the true value, and $\hat{y}_i$ is the predicted value.

Then the overall loss for the network is calculated by summing up the losses from each task:

$$Loss = L_{Bin} + L_{Reg} \tag{5.4}$$

where L_{Bin} is the binary classification loss on the wildfire delineation task, while L_{Reg} is the regression loss on the severity estimation task. This combines both losses into a single value, which the model then tries to minimize during training.

5.3 Evaluation metrics

This section provides an overview of evaluation metrics employed to assess models in both binary segmentation task and multiclass segmentation task, for wildfire delineation and severity level estimation respectively.

In machine learning, evaluating the developed model and its output using various metrics is fundamental, considering the nature of the problem and the distribution of classes. Four main metrics are commonly used for classification problems: Accuracy, Precision, Recall and F1-Score.

Accuracy is the ratio of correctly predicted observations to the total observations. It is a measure of how many classifications your model got right out of all the classifications it made. However, it can be a misleading indicator of model performance if your data is imbalanced, as in our case.

Instead, F1 score is better for handling imbalanced situations because is the harmonic mean of precision and recall, providing a balance between false positives and false negatives. Precision considers the correctness of predictions: it measures what fraction of pixels predicted as a certain category, like being within a burned area, actually coincide with the Ground Truth. In contrast, Recall assesses the estimator's capacity to detect all the pixels associated with a certain class, according to the Ground Truth. Hence, Recall calculates the proportion of pixels that were correctly predicted. This makes F1 less sensitive to class imbalances because it does not consider true negatives, which often outweigh true positives in unbalanced data. This is the main reason why F1 is more informative than accuracy and is used as an evaluation metric in this thesis.

For multiclass problems, the same evaluation metrics aforementioned can still be applied in the same way to assess the model's performance. All the formulas are summarized for both binary case and multiclass case in the table 5.5.

Evaluation metric	**Binary case**	**Multiclass case**
Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$	$Accuracy = \frac{\sum_{i=0}^{4} T_{ii}}{\sum_{i=0}^{4}\sum_{j=0}^{4} T_{ij}}$
Precision	$Precision = \frac{TP}{TP+FP}$	$Precision_i = \frac{T_{ii}}{\sum_{j=0}^{4} T_{ji}}$
Recall	$Recall = \frac{TP}{TP+FN}$	$Recall_i = \frac{T_{ii}}{\sum_{j=0}^{4} T_{ij}}$
F1-score	$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$	$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall\ _i}$

Table 5.5: Summary of the evaluation metrics for both the binary case and the multiclass case

Another metric to evaluate the performance of the segmentation model is the *Jaccard Index*, also known as the Jaccard similarity coefficient or *Intersection over Union (IoU)*. It measures the similarity between two sets by calculating the ratio of their intersection to their union. In the context of multiclass classification, the Jaccard Index is computed as follows:

$$Jaccard\ Index = \frac{|\text{Intersection}|}{|\text{Union}|} \tag{5.5}$$

where |Intersection| is the cardinality of the intersection between the predicted and actual classes, and |Union| is the cardinality of their union. It ranges between 0 and 1, with a value of 1 indicating a perfect match between the predicted and actual sets.

Regarding the prediction of damage severity level, it should be used a metric that actually measures the goodness as the distance of prediction value from the ground truth. For this type of measurement, a regression metric such as *Mean Squared Error* is the most suitable.

The *Root Mean Squared Error*, abbreviated as RMSE is a variant of MSE that provides a more interpretable measure of error, that has the same scale as the target variable, making it easier to interpret. It is calculated by taking the square root of the MSE. The formula for RMSE is given by:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{5.6}$$

RMSE provides an estimate of the average absolute error between the predicted values and the actual target values.

In both cases, a lower value indicates better performance, with a value of 0 indicating a perfect match between predicted and actual values.

5.4 Results

This section presents and discusses the results of the experiments detailed in the previous sections. The results are reported for each tested model and are categorized by task: delineation and grading. The first task evaluates the models' capabilities in identifying burned and unburned areas, while the second task assesses their performance in predicting the severity level of wildfire damage. Additionally, the scores are compared with the state-of-the-art DSU-Net [23].

5.4.1 Delineation

In the binary classification task the evaluation metrics used to assess the models are Precision, Recall, F1-Score, and IoU. The results, as reported in the table 5.6, are generally good. The DSU-Net is employed as a reference in this analysis. The obtained results are fairly impressive, achieving a Recall rate exceeding 91%. This demonstrates that the model is able to correctly identify a significant portion of the burned area, thus closely approaching the results of other models. However, the model's performance is hindered by a lack of Precision, indicating a propensity to generate a significant number of false positives.

The U-Net shows high Recall, but slightly lower Precision, which means that the network overestimates the burned areas, predicting many false positives. The U-Net++ instead shows the best performances overall, with high Precision and Recall. DeepLabV3+ achieves very similar results as U-Net++. PSPNet results comparable with other models, though they do not particularly stand out.

Interestingly, the F1-scores and IoU scores for all models, except for DSU-Net, are very close, indicating that these models similarly and effectively capture the overlap between the predicted and actual wildfire areas. Overall, the best performance is achieved by U-Net++, distancing from the baseline of DSU-Net by more 8 points on F1-score and 13% on IoU. The reason behind the success of U-Net++ could be the dense and nested skip connection of the model, that better captures the overall context of the image.

Figure 5.3 shows some prediction samples from the test set. It can be clearly seen from activation EMSR281 and EMSR209, that a possible cause of the lower performance of DSU-Net is probably due to the difficulty in correctly identifying large water bodies. One hypothesis about this issue could be that there are no large areas with water bodies in its training set, and the model interprets the dark color of the water surface as a burned area. Also, it should be mentioned that the small size of the training set may have affected the overall performance. The dataset used to train the other models in this thesis is as much as 10 times larger than that of DSU-Net.

We also note that trained models, in comparison to DSU-Net, encounter challenges in differentiating areas affected by ash or volcanic flows, such as in the activation EMS213_01 (Vesuvius). These models often inaccurately categorize most of the volcanic zone as an area affected by wildfire.

Evaluation	**Models**				
Metrics	**DS U-Net**	**U-Net**	**U-Net++**	**DeepLabV3+**	**PSPNet**
Precision	80.97	88.71	**92.1**	92.02	90.12
Recall	91.32	**95.81**	93.41	93.27	94.34
F1-Score	84.48	92.21	**92.75**	92.64	92.18
IoU	73.13	85.19	**86.38**	86.26	85.23

Table 5.6: Results of wildfire delineation task

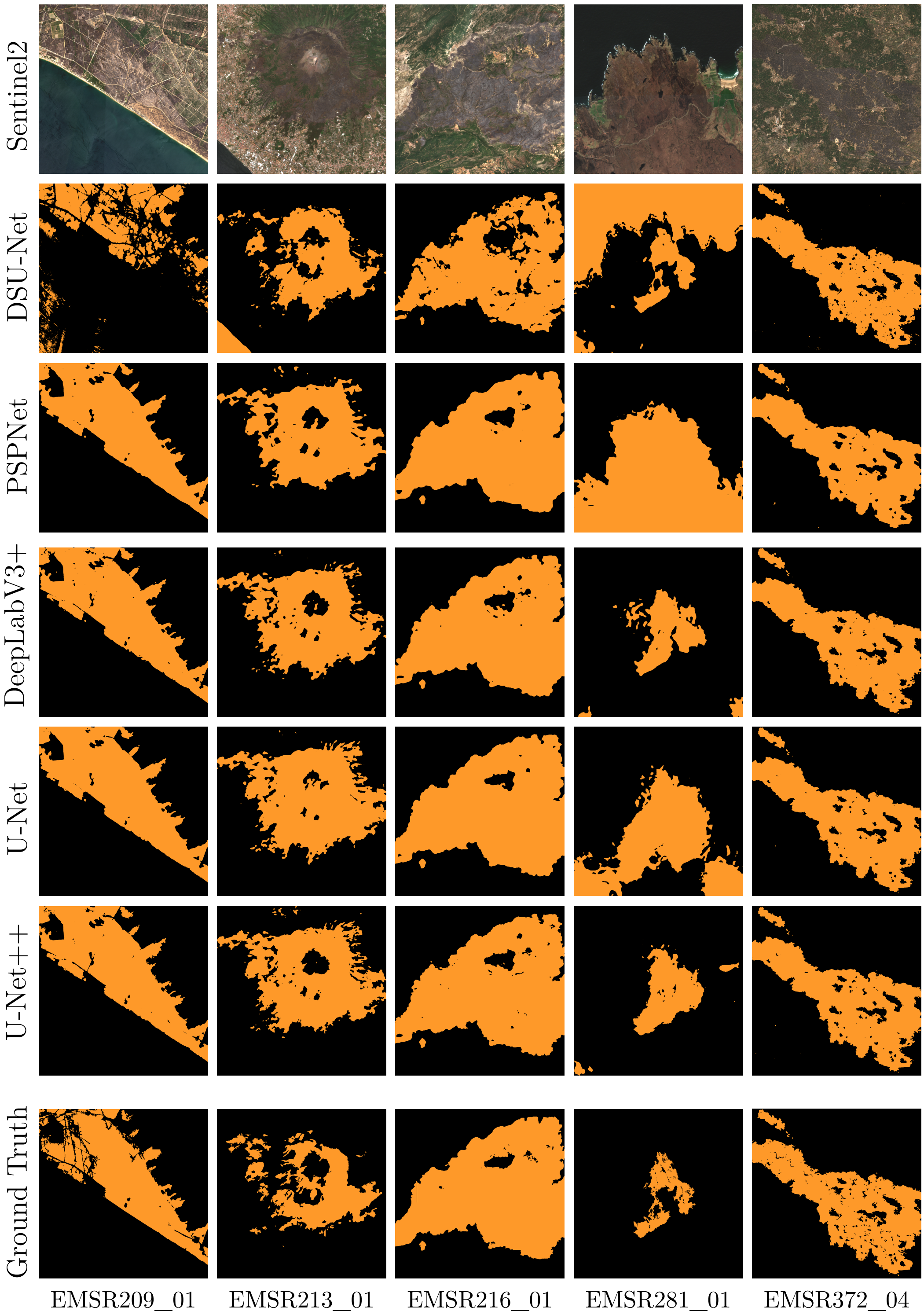

Figure 5.3: Example results of wildfire delineation task.

5.4.2 Grading

In the regression task the evaluation metrics used to assess the models are F1-Score multiclass case, IoU and RMSE. The results, as reported in Table 5.7, are divided in F1-score, IoU and RMSE for every severity level, reported as an ordinal number for the sake of space. Severity levels are mapped as follows; 0 stands for *No damage*, 1 stands for *Negligible to slight damage*, 2 stands for *Moderately Damaged*, 3 stands for *Highly damaged*, and 4 stands for *Completely destroyed*.

In a first analysis, as it is for delineation results, the DSU-Net behaves as expected. The model's prowess in identifying unburned areas, denoted as class 0, is underlined by a high F1-score of 97 and an Intersection over Union (IoU) of 94%. However, this robust performance sharply contrasts with its low efficacy in recognizing class 4. Indeed, the model demonstrates substantial effectiveness in identifying the first two classes as evidenced by the RMSE, which subsequently declines as the severity of the wildfire escalates.

The U-Net generally achieves fairly good results. The RMSE is quite lower than DSU-Net in classes 3 and 4, where the results are respectively lower by 0.6 and 0.4 points. The overall average of RMSE is lower than the baseline of the DSU-Net by 0.2.

U-Net++ achieves similar results to U-Net but gets lower results in class 4. However, the RMSE values for the U-Net++ result are quite comparable to DeepLabV3+. Both DeepLabV3+ and PSPNet demonstrate strong performance in identifying the correct class, in particular level 4, as indicated by their higher IoU and F1-scores. This also contributes to them achieving the lowest RMSE for that particular class. In terms of overall performance, U-Net surpasses all others, a contrast to the previous section where U-Net++ was the top-performing model. This suggests that U-Net excels particularly when dealing with continuous values.

Figure 5.4 shows the prediction samples for the severity estimation task. As before DSU-Net encounters the most problem. This is because a bad classification from the Binary U-Net of double-step can lead the Regression U-Net to make more mistakes because it will consider 0-valued-regions as unburned and every other unburned region not detected by the Binary U-Net will be considered as burned [23]. The issue of incorrect classification of water bodies is amplified. Evidently, in the case of activation EMSR281, water is incorrectly classified as severity level 2. This significantly undermines the model's performance. Furthermore, the model struggles to accurately identify higher severity classes, such as levels 3 and 4.

Evaluation Metrics	**Severity**	**Models**				
		DSU-Net	**U-Net**	**U-Net++**	**DLV3+**	**PSPNet**
F1-Score	0	97.37	98.47	98.64	98.63	98.52
	1	11.90	17.89	17.52	17.65	16.58
	2	28.85	24.44	23.74	24.72	26.85
	3	45.63	55.93	54.77	51.34	53.38
	4	02.42	12.99	06.94	26.83	20.42
	Mean	37.34	41.94	40.32	**43.83**	43.15
IoU	0	94.89	96.99	97.33	97.31	97.09
	1	06.33	09.82	09.61	09.68	09.01
	2	16.85	13.92	13.47	14.10	15.50
	3	29.56	38.87	37.71	34.54	36.41
	4	01.22	06.94	03.59	15.49	11.37
	Mean	29.77	33.30	32.34	**34.24**	33.87
RMSE	0	0.2385	0.2695	0.2756	0.2586	0.2713
	1	0.8825	1.0756	1.1278	1.1271	1.1267
	2	1.0664	0.8998	0.9537	0.9520	0.9177
	3	1.3613	0.7327	0.7745	0.8268	0.8203
	4	1.8409	1.3057	1.3110	1.2777	1.2845
	Mean	1.0779	**0.8526**	0.8885	0.8884	0.8841

Table 5.7: Results of damage severity estimation task

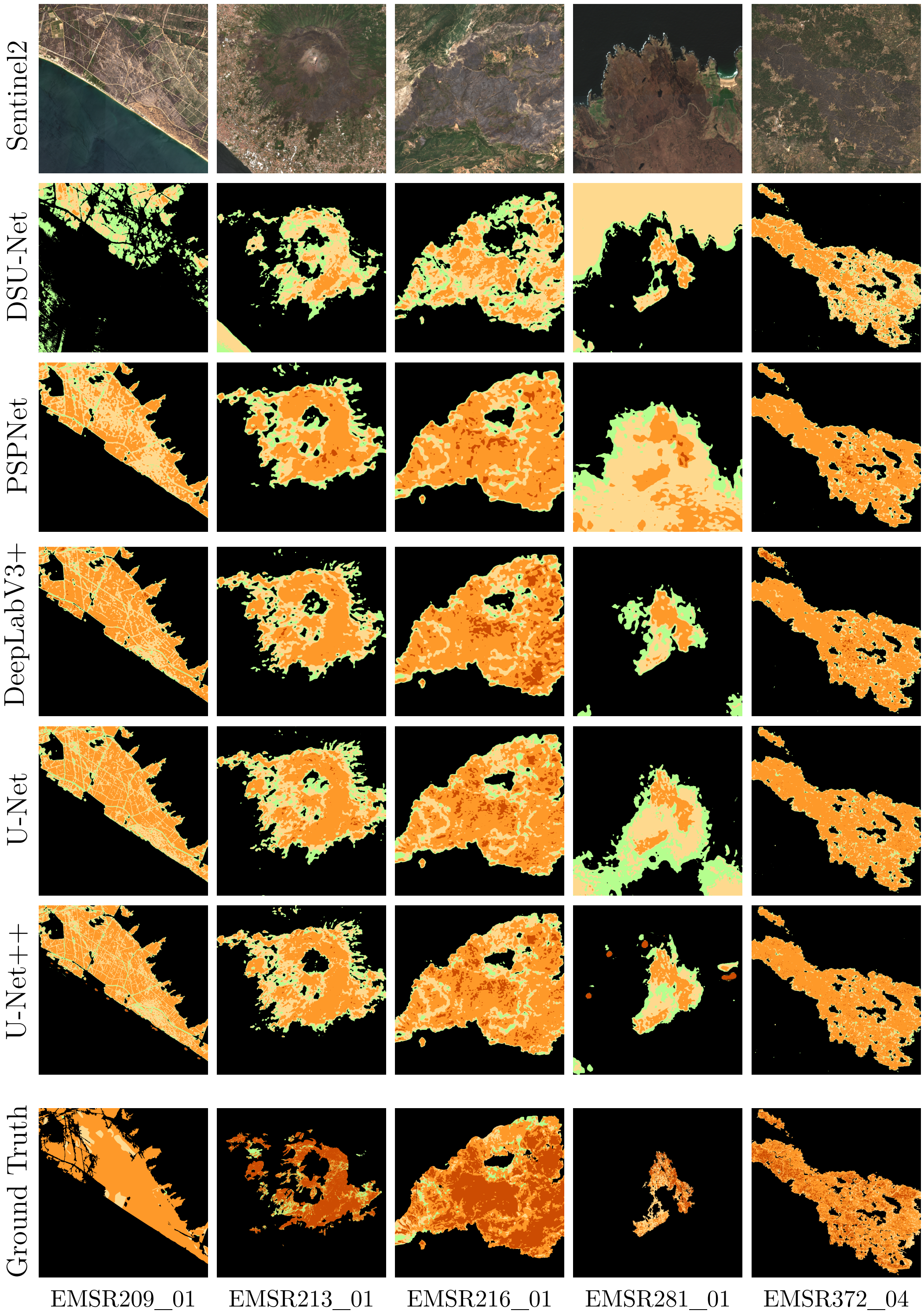

Figure 5.4: Example results of damage severity estimation task

Chapter 6

Conclusions

The purpose of this work is to evaluate the performances of several state-of-the-art models using a large dataset of Sentinel-2 imagery of wildfire, focused primarily on the European region. Compared to the literature, which commonly uses pre-fire and post-fire satellite acquisitions, this approach uses only of post-fire data. The combination of this sizable dataset with the multitask approach has demonstrated good results, exceeding past studies in this field. These results illustrate the effectiveness of our approach in both identifying areas affected by wildfire and accurately estimating the severity of the damage.

Notably, the cloud cover label integrated into the dataset significantly reduced possible errors during inference, demonstrating the potential of the technique in real-world application, where cloud cover can often obscure imagery.

As seen from the chapter 5.4, model performances, especially in a severity estimation task, offer a margin of improvement.

A potential future expansion of this thesis could be the inclusion of land cover information as an additional task in the multitask approach. Land cover data provides valuable information about vegetation types, which could contribute to a more comprehensive understanding of wildfire patterns and spreads, improving the general results.

Furthermore, the exploration and incorporation of Transformer-based models, like SegFormer, represent another interesting direction for future work. These models have shown promising results in various fields, and their capability to handle long-range dependencies in data may potentially enhance the accuracy of wildfire delineation and severity estimation tasks.

Appendix A

In this appendix are reported all activations in the dataset, divided by training-validation-test set.

Set	Activations
Validation	EMSR278_01, EMSR291_01, EMSR291_02, EMSR401_01, EMSR401_02, EMSR435_01, EMSR449_01, EMSR508_01, EMSR510_01, EMSR516_01, EMSR521_01, EMSR529_02, EMSR529_03, EMSR529_04, EMSR529_05, EMSR534_01, EMSR571_01, EMSR589_01, EMSR602_01, EMSR602_02, EMSR602_03, EMSR602_04, EMSR602_05, EMSR602_06, EMSR602_09, EMSR602_10, EMSR606_01
Test	EMSR207_01, EMSR209_01, EMSR210_01, EMSR211_01, EMSR213_01, EMSR213_02, EMSR213_03, EMSR213_06, EMSR213_07, EMSR213_08, EMSR213_09, EMSR213_10, EMSR213_11, EMSR214_01, EMSR214_02, EMSR214_05, EMSR214_06, EMSR216_01, EMSR217_01, EMSR217_02, EMSR217_03, EMSR217_04, EMSR219_01, EMSR221_01, EMSR227_01, EMSR237_01, EMSR239_01, EMSR248_01, EMSR248_03, EMSR248_04, EMSR250_01, EMSR250_04, EMSR252_01, EMSR254_01, EMSR254_02, EMSR254_03, EMSR254_04, EMSR281_01, EMSR290_03, EMSR298_02, EMSR298_05, EMSR298_06, EMSR302_01, EMSR365_01, EMSR368_01, EMSR371_01, EMSR372_04, EMSR373_01

Set	Activations
Training	EMSR213_05, EMSR213_13, EMSR213_14, EMSR213_15, EMSR213_16, EMSR213_17, EMSR213_18, EMSR213_19, EMSR213_20, EMSR213_21, EMSR213_22, EMSR214_03, EMSR230_01, EMSR247_01, EMSR248_02, EMSR250_02, EMSR250_03, EMSR259_01, EMSR288_01, EMSR295_01, EMSR299_01, EMSR300_01, EMSR300_02, EMSR303_01, EMSR305_01, EMSR307_01, EMSR316_01, EMSR331_01, EMSR344_01, EMSR353_01, EMSR360_01, EMSR362_01, EMSR363_01, EMSR367_01, EMSR369_01, EMSR369_02, EMSR369_03, EMSR370_01, EMSR374_01, EMSR375_01, EMSR377_01, EMSR378_01, EMSR380_01, EMSR381_01, EMSR382_01, EMSR389_01, EMSR390_01, EMSR396_01, EMSR396_02, EMSR396_03, EMSR408_02, EMSR426_01, EMSR426_02, EMSR428_01, EMSR430_01, EMSR440_01, EMSR443_01, EMSR447_01, EMSR448_01, EMSR453_01, EMSR455_01, EMSR457_01, EMSR458_01, EMSR462_01, EMSR500_01, EMSR506_01, EMSR510_01, EMSR512_01, EMSR515_01, EMSR522_01, EMSR523_02, EMSR523_05, EMSR523_08, EMSR525_01, EMSR526_01, EMSR527_01, EMSR527_02, EMSR528_01, EMSR530_01, EMSR531_01, EMSR532_01, EMSR533_01, EMSR533_02, EMSR537_01, EMSR538_01, EMSR539_01, EMSR540_01, EMSR541_01, EMSR542_01, EMSR543_01, EMSR544_01, EMSR545_01, EMSR547_01, EMSR560_01, EMSR573_01, EMSR576_01, EMSR578_01, EMSR578_02, EMSR579_01, EMSR579_02, EMSR579_04, EMSR579_05, EMSR579_06, EMSR579_07, EMSR579_08, EMSR580_01, EMSR581_01, EMSR581_02, EMSR582_01, EMSR582_02, EMSR583_01, EMSR587_01, EMSR588_01, EMSR590_01, EMSR591_01, EMSR592_01, EMSR592_02, EMSR593_01, EMSR594_01, EMSR595_01, EMSR596_01, EMSR597_01, EMSR598_01, EMSR599_01, EMSR599_02, EMSR600_01, EMSR601_01, EMSR603_01, EMSR605_01, EMSR607_01, EMSR608_01, EMSR609_01, EMSR610_01, EMSR613_01, EMSR617_01, EMSR618_01, EMSR620_01, EMSR621_01, EMSR623_01, EMSR624_01, EMSR625_01, EMSR625_02, EMSR626_01, EMSR627_01, EMSR628_01, EMSR632_01, EMSR633_01, EMSR638_01, EMSR656_01

Table A.1: Dataset

Bibliography

[1] United Nations Office for Disaster Risk Reduction. *Making Development Sustainable: The Future of Disaster Risk Management. Global Assessment Report on Disaster Risk Reduction.* https://www.preventionweb.net/english/hyogo/gar/2015/en/home/GAR_2015/GAR_2015_1.html. Accessed: yyyy-mm-dd. 2015 (cit. on p. 1).

[2] The Guardian. *Canada Wildfire Smoke Returns, Affecting New York Air Quality.* 2023. URL: https://www.theguardian.com/us-news/2023/jun/27/canada-wildfire-smoke-returns-new-york-air-quality (cit. on p. 1).

[3] Giacomo Certini. «Effects of fire on properties of forest soils: a review». In: *Oecologia* 143.1 (2005), pp. 1–10. DOI: 10.1007/s00442-004-1788-8. URL: https://link.springer.com/article/10.1007/s00442-004-1788-8 (cit. on p. 2).

[4] Joint Research Centre. *EU 2022 Wildfire Season Was Second Worst on Record.* 2023. URL: https://joint-research-centre.ec.europa.eu/jrc-news-and-updates/eu-2022-wildfire-season-was-second-worst-record-2023-05-02_en (cit. on p. 2).

[5] John Doe. «The Concept of Artificial Neurons — Perceptrons in Neural Networks». In: *Towards Data Science* (2023). URL: https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc (cit. on p. 5).

[6] G. Holmgren, P. Andersson, A. Jakobsson, et al. «Artificial neural networks improve and simplify intensive care mortality prognostication: a national cohort study of 217,289 first-time intensive care unit admissions». In: *Journal of Intensive Care* 7 (2019), p. 44. DOI: 10.1186/s40560-019-0393-1 (cit. on p. 6).

[7] *Real-World Applications of Convolutional Neural Networks*. https://vitalflux.com/real-world-applications-of-convolutional-neural-networks/ (cit. on p. 6).

[8] Jeremy Jordan. *Learning rate*. https://www.jeremyjordan.me/nn-learning-rate/. 2023 (cit. on p. 8).

[9] Senay Cakir, Marcel Gauß, Kai Häppeler, Yassine Ounajjar, Fabian Heinle, and Reiner Marchthaler. *Semantic Segmentation for Autonomous Driving: Model Evaluation, Dataset Generation, Perspective Comparison, and Real-Time Capability*. 2022. arXiv: 2207.12939 [cs.CV] (cit. on p. 9).

[10] Eric Guérin, Killian Oechslin, Christian Wolf, and Benoıt Martinez. «Satellite Image Semantic Segmentation». In: *CoRR* abs/2110.05812 (2021). arXiv: 2110.05812. URL: https://arxiv.org/abs/2110.05812 (cit. on p. 9).

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. «U-Net: Convolutional Networks for Biomedical Image Segmentation». In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597 (cit. on pp. 9, 26).

[12] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. «UNet++: A Nested U-Net Architecture for Medical Image Segmentation». In: *CoRR* abs/1807.10165 (2018). arXiv: 1807.10165. URL: http://arxiv.org/abs/1807.10165 (cit. on pp. 9, 26, 28).

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. «DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184 (cit. on p. 9).

[14] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. «Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation». In: *CoRR* abs/1802.02611 (2018). arXiv: 1802.02611. URL: http://arxiv.org/abs/1802.02611 (cit. on pp. 9, 27, 29).

[15] Mohammad Hosein Hamian, Ali Beikmohammadi, Ali Ahmadi, and Babak Nasersharif. *Semantic Segmentation of Autonomous Driving Images by the Combination of Deep Learning and Classical Segmentation*. 2021. DOI: 10.1109/CSICC52343.2021.9420573 (cit. on p. 9).

[16] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. «Pyramid Scene Parsing Network». In: *CoRR* abs/1612.01105 (2016). arXiv: `1612.01105`. URL: `http://arxiv.org/abs/1612.01105` (cit. on pp. 9, 29, 30).

[17] Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». In: *CoRR* abs/2010.11929 (2020). arXiv: `2010.11929`. URL: `https://arxiv.org/abs/2010.11929` (cit. on p. 9).

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. «Attention Is All You Need». In: *CoRR* abs/1706.03762 (2017). arXiv: `1706.03762`. URL: `http://arxiv.org/abs/1706.03762` (cit. on p. 9).

[19] Author's name or names are missing. «Normalized Burn Ratio Plus (NBR+): A New Index for Sentinel-2 Imagery». In: *Remote Sensing* 14.1727 (Apr. 2022). DOI: `10.3390/rs14071727`. URL: `https://doi.org/10.3390/rs14071727` (cit. on p. 10).

[20] C.H. Key, N.C. Benson, D.C. Lutes, R.E. Keane, J.F. Caratti, S. Steve, and L.J. Gangi. *Landscape assessment (LA). In FIREMON: Fire Effects Monitoring and Inventory System.* Tech. rep. Fort Collins, CO, USA: Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2006 (cit. on p. 10).

[21] G. Navarro, I. Caballero, G. Silva, P.C. Parra, Á. Vázquez, and R. Caldeira. «Evaluation of forest fire on Madeira Island using Sentinel-2A MSI imagery». In: *Int. J. Appl. Earth Obs. Geoinf.* 58 (2017), pp. 97–106 (cit. on p. 10).

[22] J.D. Miller and A.E. Thode. «Quantifying burn severity in a heterogeneous landscape with a relative version of the delta Normalized Burn Ratio (dNBR)». In: *Remote Sens. Environ.* 109 (2007), pp. 66–80 (cit. on p. 11).

[23] Alessandro Farasin, Luca Colomba, and Paolo Garza. «Double-Step U-Net: A Deep Learning-Based Approach for the Estimation of Wildfire Damage Severity through Sentinel-2 Satellite Data». In: *Applied Sciences* 10.12 (June 2020), p. 4332. ISSN: 2076-3417. DOI: `10.3390/app10124332`. URL: `http://dx.doi.org/10.3390/app10124332` (cit. on pp. 12, 40, 43).

[24] Daniele (Rege Cambrin, Luca Colomba, and Paolo Garza. «DVision Transformers for Burned Area Delineation». In: (2022). URL: `https://ceur-ws.org/Vol-3343/paper4.pdf` (cit. on p. 12).

[25] Seyd Teymoor Seydi, Mahdi Hasanlou, and Jocelyn Chanussot. «DSMNN-Net: A Deep Siamese Morphological Neural Network Model for Burned Area Mapping Using Multispectral Sentinel-2 and Hyperspectral PRISMA Images». In: *Remote Sensing* 13.24 (Dec. 2021), p. 5138. ISSN: 2072-4292. DOI: 10.3390/rs13245138. URL: http://dx.doi.org/10.3390/rs13245138 (cit. on p. 12).

[26] *Copernicus EMS Web Page.* URL: https://emergency.copernicus.eu/ (cit. on p. 13).

[27] *Copernicus activation product wiki.* URL: https://gis4schools.readthedocs.io/en/latest/part4/4_1.html (cit. on p. 13).

[28] *Copernicus damage assessment.* URL: https://emergency.copernicus.eu/mapping/ems/damage-assessment (cit. on p. 14).

[29] K. Karra, C. Kontgis, and et al. «Global land use/land cover with Sentinel-2 and deep learning». In: *IGARSS 2021-2021 IEEE International Geoscience and Remote Sensing Symposium.* IEEE. 2021 (cit. on p. 18).

[30] D. Zanaga et al. *ESA WorldCover 10 m 2020 v100.* https://doi.org/10.5281/zenodo.5571936. 2021 (cit. on p. 18).

[31] Justin D. Braaten, Warren B. Cohen, and Zhiqiang Yang. «Automated cloud and cloud shadow identification in Landsat MSS imagery for temperate ecosystems». In: *Remote Sensing of Environment* 169 (2015), pp. 128–138. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2015.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0034425715300948 (cit. on p. 19).

[32] C. Aybar, L. Ysuhuaylas, J. Loja, et al. «CloudSEN12, a global dataset for semantic understanding of cloud and cloud shadow in Sentinel-2». In: *Sci Data* 9 (2022). Received 05 September 2022; Accepted 29 November 2022; Published 24 December 2022, p. 782. DOI: 10.1038/s41597-022-01878-2. URL: https://doi.org/10.1038/s41597-022-01878-2 (cit. on p. 19).

[33] R. Caruana. «Multitask Learning». In: *Machine Learning* 28 (1997). Issue Date: July 1997, pp. 41–75. DOI: 10.1023/A:1007379606734. URL: https://doi.org/10.1023/A:1007379606734 (cit. on p. 24).

[34] Simone Monaco, Salvatore Greco, Alessandro Farasin, Luca Colomba, Daniele Apiletti, Paolo Garza, Tania Cerquitelli, and Elena Baralis. «Attention to Fires: Multi-Channel Deep Learning Models for Wildfire Severity Prediction». In: *Applied Sciences* 11.22 (2021), p. 11060. DOI: `https://doi.org/10.3390/app112211060` (cit. on p. 32).

[35] Leslie N. Smith. «No More Pesky Learning Rate Guessing Games». In: *CoRR* abs/1506.01186 (2015). arXiv: `1506.01186`. URL: `http://arxiv.org/abs/1506.01186` (cit. on p. 34).