

POLITECNICO DI TORINO

Facoltà di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

**Automatic processing of satellite acquisitions for
burnt area detection and damage estimation**



Relatori:

Prof. Paolo Garza
Dott. Alessandro Farasin

Correlatore:

Prof. Daniele Loiacono

Candidato:

Luca COLOMBA

Ringraziamenti

E così ci troviamo alla fine di un percorso durato cinque anni. Cinque anni che all'inizio sembravano lunghi e interminabili, cinque anni che guardandomi ora indietro sono volati. Un percorso pieno di sacrifici, progetti, impegni e tanta fatica, ma anche pieno di soddisfazioni, sorprese, sorrisi e risate. Cercare di racchiudere tutto in queste poche righe non è facile.

A tutta la mia famiglia, un grande grazie per avermi supportato e aiutato sempre, specialmente in questi anni di università. Per esserci sempre stati, in ogni momento in questi anni di crescita. Per avermi guidato, aiutato e consigliato quando ero in dubbio e non avevo idea di che strada percorrere.

A Cristina, amica dai tempi del liceo, per la compagnia e per le interminabili chiacchierate insieme. Per esserci stata nonostante tutti questi anni, nonostante i periodi difficili e gli impegni, per i tutti i discorsi che ci siamo fatti, specialmente in queste ultime settimane, e per tutti i momenti che verranno.

A Fabrizio e Gerardo per questi 5 anni passati insieme in aula e in aula studio. Per le ore passate a confrontarci sui corsi, sugli esami e sui progetti.

A Luca, Federico e Giuseppe per essere stati grandi compagni di studio e amici. Per esserci supportati e aiutati a vicenda nello studio, nei laboratori e per le infinite discussioni sul mondo tech.

A Alessandro, Marco, Celeste, Milica del Team DAHEC per aver passato momenti indimenticabili in un anno di progetto. Per il tanto lavoro fatto assieme, per le interminabili call, le grandi risate e le grandissime soddisfazioni del nostro progetto.

A Ilaria e Nicola per essere stati degli importanti compagni di avventure e amici, specialmente in questi ultimi due anni, per esserci ritrovati nel Team DAHEC, per tutto l'impegno che ci avete messo, sempre, in qualunque situazione. È stato un immenso piacere lavorare e scherzare con voi.

A Iacopo, compagno di stanza delle School e grande amico, con la A maiuscola, per tutti i momenti passati insieme, per le chiacchierate e la felicità che riesci a trasmettere al prossimo.

A Andrea e Nour, più volte compagni alle School, per la compagnia e le nottate passate in gruppo a lavorare e scherzare.

A tutti gli straordinari ragazzi dell'Alta Scuola Politecnica che ho conosciuto in questi due anni, alle interminabili lezioni e a tutte le ore passate insieme.

A Francesco, Giulia, Maria Laura e Massimiliano per la compagnia in tutti questi anni, per le serate e i bellissimi momenti.

Ad Alessandro per la pazienza durante questi mesi di tesi, per avermi aiutato e supportato nella fine di questo percorso, per avermi fatto apprezzare il mondo della ricerca. A tutti i ragazzi della Fondazione LINKS che mi hanno accompagnato in questo percorso di crescita e in questo periodo conclusivo del mio percorso di studi.

A tutti gli amici che ho conosciuto e che hanno reso unico questo percorso, che sono riusciti a rendere più leggeri e a far volare questi anni di università, tra risate, sorrisi e fatica. Il raggiungimento di questo traguardo è anche merito vostro.

Forse le parole non sono sufficienti a esprimere tutta la mia gratitudine e ciò che provo in questo momento, ma ci tenevo a fissare e a trasmettere queste parole in qualche modo.

Grazie,
Luca

Abstract

Every year hazardous events take place all around the globe, causing damages from many point of views: humanitarian losses, crisis, economical losses, environmental issues and damages to the vegetation in short terms. Considering a longer time span, areas involved by catastrophic events suffer from several drawbacks: cultural heritage loss, possible damages to flora and fauna and the need to rebuild structures and plan for vegetation regrowth. European Union and the European Space Agency are becoming more and more active in collecting information about natural disasters. Through the Copernicus Emergency Management System (EMS), data about past catastrophic events are collected and categorized according to the type of event, the state and the region in which they took place.

The goal of this thesis is to focus on wildfire events and exploit data provided by Copernicus EMS to train a machine learning algorithm to automatically determine damaged areas by a past wildfire and eventually provide a severity level estimation. In fact, Copernicus system monitors forest fires and after the fire is extinguished, human operators are in charge of providing delineation and grading maps, i.e. precisely identify which areas were affected by the forest fire and the entity of the damage, from slight damages to total destruction of the area of interest. Starting from Copernicus data, a dataset was generated by collecting satellite imagery from Sentinel2-L1C mission to train and feed CNN-based deep neural networks known in the state of the art of biomedical image segmentation. This project demonstrates the good learning capabilities of the implemented models in distinguishing damaged areas and areas not affected by the wildfires and underlines the limits of the considered approaches in providing a severity level estimation.

Contents

1	Introduction	9
2	State of the art	13
2.1	Related Works	13
2.2	Machine Learning & Deep Learning	14
2.2.1	Neural Networks (NN)	14
2.2.2	Convolutional Neural Networks (CNN)	16
2.2.3	Computer Vision & Image Segmentation	17
2.3	Satellite imagery analysis	21
3	Dataset and Tools	27
3.1	Data sources and services	27
3.1.1	Data sources	27
3.1.2	SentinelHub and Data collection	28
3.1.3	Mask generation	29
3.1.4	Data Cleaning	30
3.1.5	Label distribution	30
3.2	Tools & Frameworks	30
3.2.1	GDAL	31
3.2.2	QGis	31
3.2.3	Python	31
3.2.4	OpenCV & skimage	31
3.2.5	scikit-learn	32
3.2.6	PyTorch	32
4	Methodology	33
4.1	Problem interpretation and approaches	33
4.2	Preliminary analyses	35
4.2.1	Analysis on burnt area indexes and severity levels	35

4.2.2	Analysis on 13 bands	37
4.3	Data Preprocessing and input pipeline	38
4.4	BandsAggregator and training process	40
4.5	CUMedVision2 and contour generation	41
4.6	Input features - Incremental approach	41
4.7	Comparison with data before and after the activation event	42
4.8	Quality measures	44
4.8.1	Accuracy	44
4.8.2	Precision	45
4.8.3	Recall	45
4.8.4	F1 score	46
4.9	Validation procedure	46
4.10	Training system	47
5	Performance Evaluation	49
5.1	RGB input	50
5.1.1	UNet	50
5.1.2	CUMedVision1	55
5.1.3	CUMedVision2	59
5.1.4	Comparison with data before and after the activation event . .	60
5.2	Multi spectral bands input	62
5.2.1	UNet	64
5.2.2	CUMedVision1	69
5.2.3	CUMedVision2	73
5.2.4	Comparison with data before and after the activation event . .	73
5.3	Models comparison	77
6	Conclusions	83
6.1	Future developments	83
6.1.1	Feature engineering	84
6.1.2	Architectural and loss improvements	85
6.2	Final considerations	85

Chapter 1

Introduction

Catastrophic hazards are of different nature, ranging from floods, earthquakes, severe storms to humanitarian crisis. Such events represent potential threats from many point of views: humanitarian, economical and environmental. Forest fires are no exception: damage to properties, destruction of flora and fauna, economical loss, massive emission of CO₂, CO and air quality problems in the neighboring areas. Moreover, large emissions caused by fire activities have implications on climate change and global warming as stated in [1].

Global awareness about these types of events is constantly increasing, especially after the Siberia wildfires and the Amazon rainforest wildfires in 2019, where several millions of hectares are still burning.

Fires are caused either by human or natural causes. In the first case, incidents caused by machinery, petroleum leaks, electric faults or even people deliberately starting fires, while in the latter the main causes are lightning and volcanic activities. Especially in case of naturally-caused fires, the possibility to detect and intervene are lower due to difficulty in reaching certain areas.

Furthermore, after fire is fully extinguished, other negative aspects are present:

- economical loss: damage to properties and crops, thus resulting in the reduction of economical incomes and requires investments for post-fire restoration;
- local heritage loss;
- reduction of vegetation: the absorption of carbon-dioxide is reduced due to the reduction of vegetation in the area. A second consequence is the increase of soil erosion, which could possibly lead to other catastrophic events such as landslides;

- air pollution, caused by gas emissions during the event;

Evaluation for restoration must be made: depending on the area, the severity of the damages and the soil characteristics, a partial restoration of the original area may requires several decades or several hundreds years.

Climate change and global warming may increase the probability of having long lasting and more extended wildfires, thus underlining the importance of prevention and timely intervention during its occurrence and monitoring the burnt area after the fire extinguished.

For this reasons, real-time detection systems and the ability to provide a severity estimation is fundamental. The objective of this thesis is to evaluate, using a machine learning-based approach, the possibility to automate the identification of burnt areas and provide a severity estimation level based on satellite acquisitions and geospatial data. In fact, thanks to different sensors present on satellites and the possibility to collect data from locations around the globe, there are no difficulties in collecting post-fire data, unless in-situ information are required.

In particular, two different approaches were considered: the first exploits imagery acquired using cheaper sensors, i.e. RGB sensors, while the second approach exploits the data coming from all the available bandwidths from the satellites.

Identification of burnt areas and severity estimation can be both categorized as an image segmentation task. Starting from state of art techniques known in biomedical image segmentation, different models have been applied to satellite imagery to understand the tasks' feasibility and performance evaluation.

The thesis is structured as follows:

- State of the Art: a brief overview of Neural Networks is given together with the most recent techniques used in the context of Computer Vision and image segmentation;
- Dataset and Tools: the different data sources, the formats and the services used to obtain the dataset are described, together with the tools, frameworks and libraries used for the development of the system;
- Methodology: the data preprocessing, data splitting, training strategies, loss functions and evaluation metrics are described;
- Performance Evaluation: the performances of the models implemented are evaluated and compared, with some considerations about possible limitations of each of the approaches;

- Conclusions: a high level overview of the approaches is given together with some considerations about future developments;

Chapter 2

State of the art

This chapter provides a brief introduction to Machine Learning, with a particular focus on Neural Networks and Deep Learning and their main application in the field of Computer Vision and Image Segmentation.

2.1 Related Works

In the recent years, the field of Deep Learning and Computer Vision is becoming more and more subject of interest thanks to the increasing computing capabilities offered by hardware and GPUs acceleration. Modern Computer Vision problems are tackled with approaches entirely based on neural network models. Such models are extremely complex and difficult to interpret, but at the same time offer great possibility of improvements and learning thanks to the capabilities of features extraction and encoding, compared to other approaches based on classical machine learning models such as SVMs, random forests where models' improvements are entirely based on feature engineering and feature creation.

In particular, all the image segmentation models implemented in this work were created in the context of medical image segmentation, where research is active for segmentation in different tasks, such as [2], [3], [4], [5].

The application of machine learning-based approaches in the context of satellite imagery at the moment of writing is essentially limited to SVMs, decision trees, Random Forest models, K-means clustering and time series analysis ([6], [7], [8], [9]), with some research studies implementing Artificial Neural Networks (ANN) [6].

The adoption of Convolutional Neural Networks (CNN) for the analysis of satellite images could bring several benefits in the field: thanks to convolutions, deep neural networks are able to learn and classify pixels also considering spatial correlations and

morphological information.

2.2 Machine Learning & Deep Learning

Machine Learning is a subset of the Artificial Intelligence research field where algorithms and mathematical models are built to let a machine learn a task. In this section only Supervised Learning tasks are analysed.

Supervised machine learning algorithms require the input data and the labels/desired results. The training process consists in updating a set of learnable parameters such that the output of the model, i.e. the prediction, is as accurate as possible. The output of the training process is the model itself.

More formally, let $\{X_{tr}, y_{tr}\}$ be the training set and $\{X_{test}, y_{test}\}$ be the test set, where X and y are the input features and the labels respectively. The goal of the training process is to determine a function f such that the prediction \hat{y} (Equation 2.1) maximizes some evaluation metrics computed on the test set, i.e. the prediction is as close as possible to the ground truth y .

$$\hat{y} = f(X) \quad (2.1)$$

Thanks to the possibility of collecting and processing great amount of data at low cost due to the diffusion of IoT devices, smartphones, new sensors and the improvements in the computational capabilities, Machine Learning and more in general Data Analysis is becoming one of the hottest topics at the time of writing. Supervised Machine Learning models can be used for automatic fraud detection systems [10], human activity recognition [11] but also in Computer Vision for object identification or semantic segmentation of images. More specifically, modern machine learning techniques for object recognition and analysis of images are entirely based on CNNs models, which are described in the following paragraphs.

2.2.1 Neural Networks (NN)

Neural networks [12] are complex models composed of several interconnected neurons. Each neuron receives as input a vector x which is multiplied by the weight vector w . Then, a bias term b is added and a non-linear function, also known as the activation function f , is applied to the result (Equation 2.2). The training process consists in learning the parameters \vec{w} and b . The structure of a neuron is shown in Figure 2.1.

$$y = \vec{x} \cdot \vec{w} + b \quad (2.2)$$

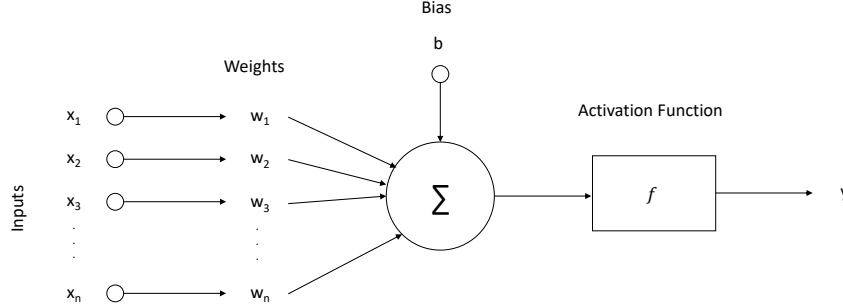


Figure 2.1: Structure of a neuron

Multiple neurons can be combined to create multiple layers, which can be stacked to obtain Artificial Neural Networks (ANN), such as in Figure 2.2.

Intermediate layers are called hidden layers. The input vector of a hidden neuron is obtained from the output of all the neurons in the previous layer. The number of learnable parameters in an ANN depends on the number of neurons: the higher the number, the higher the number of parameters.

Given the output of a neural network, the training process is performed by minimizing a loss function $l = l(y, \hat{y})$ through gradient descent, backpropagation algorithm and the chain rule [13]. The loss function is used to evaluate in a quantitative way the model's prediction on an input dataset with respect to the ground truth and update the model's parameters accordingly. In this thesis, two loss functions have been used: cross entropy (Equation 2.3) and dice loss (Equation 2.4).

$$CE = - \sum_{c=1}^C \sum_{i=1}^N y_{ci} \log \hat{y}_{ci} \quad (2.3)$$

$$DL = 1 - 2 \cdot \frac{|X \cap Y|}{|X| + |Y|} \quad (2.4)$$

Where:

- C and N represent the number of classes and number of data points respectively;

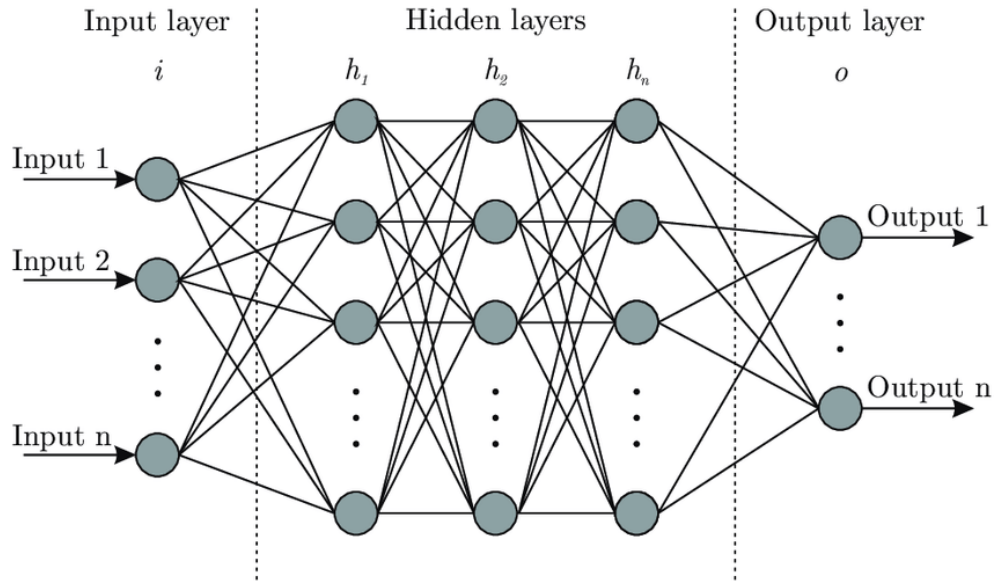


Figure 2.2: Structure of an ANN

- y_{ci} is a binary indicator (0 or 1): it is 1 if the point i belongs to class c , 0 otherwise;
- \hat{y}_{ci} is the probability of point i to belong to class c ;
- X is the prediction set;
- Y is the ground truth set;
- $|X|$ is the cardinality of set X ;

2.2.2 Convolutional Neural Networks (CNN)

Simple ANNs do not adapt very well to Computer Vision tasks due to the characteristics of simple, fully connected layers: each neuron is able to take as input information about one single pixel, without considering any spatial correlations and shapes among the pixel neighbourhood. For this reason, Convolutional Neural Networks represent the state of the art in modern Computer Vision.

CNNs are based on Convolutional Layer (Figure 2.3): a layer is defined by a kernel of size $n \times m$ which is multiplied element-wise by a portion of the input image. The new output feature is computed by summing up all the multiplied elements.

This operation is performed by sliding the kernel over the entire input image, thus generating as output a feature map.

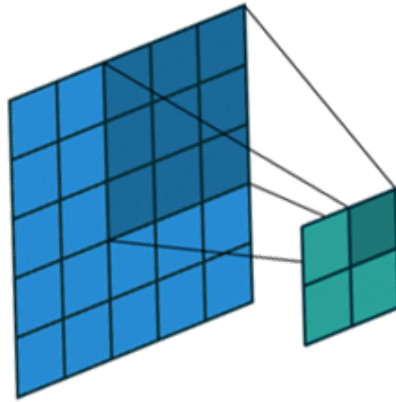


Figure 2.3: Example of a convolutional layer with a 3x3 kernel. The blue tensor is the input feature map, the green tensor is the output feature map

A convolutional layer is characterized by different parameters which determine the output shape: padding added to the input image, stride, kernel size and dilation.

2.2.3 Computer Vision & Image Segmentation

The goal of Computer Vision is to train computer systems to acquire the capabilities of human visual systems by analysing images and videos. Image analysis can be divided into the following categories:

- classification: given an input image, the system assigns a single label to the entire image, e.g. vehicle, person, dog;
- classification and localization: given an input image, the system assigns a single label to the entire image and localizes with a bounding box the object that lead the system to make that prediction;
- object detection: given an input image, the system identifies and localizes different objects in the scene, assigning a class label to each of them;
- semantic segmentation: given an input image, the system assigns a label to each pixel, thus understanding the content and the objects;

- instance segmentation: given an input image, the system assigns a label to each pixel, distinguishing objects with the same semantic meaning, e.g. two overlapped vehicles are identified as two different objects of class 'vehicle';

Image analysis and segmentation problems are tackled through CNNs and Deep Learning. Stacked convolutional layers act as feature extractors, starting from low level features such as shapes to higher level features, necessary for classification and segmentation purposes.

In general, segmentation models are based on an encoder-decoder architecture, where:

- encoders/downsamplers reduce the input image's resolution, increasing the number of intermediate features and extracting high level relevant information. They are characterized by a series of convolutional layer, batch normalizations and ReLu;
- decoders/upsamplers process the output of encoders: through transposed convolutions or other interpolation techniques, they upsample the encoded image, reducing the number of features, thus localizing the most relevant features;

Through the gradient descent algorithm, neural networks are able to automatically learn the most relevant intermediate features for segmentation purposes. A final convolution with a kernel of size 1x1 is used to perform a pixel-wise classification over the reconstructed image. The number of output channels is equal to the number of classes.

In the following sections a brief description of the implemented neural networks is provided. All of them have been applied in the field of biomedical image segmentation.

UNet

UNet [3] is composed of 4 downsamplers and 4 upsamplers. Prior to each down-sample operation, the intermediate feature maps are processed by convolutions, batch-normalization and Rectified Linear Unit (ReLU). Then, a max-pooling layer with a kernel of size 2x2 is used to halve the image's resolution. The next convolutions double the number of intermediate features/channels.

Upsampling operations are performed by transposed convolutional layers to increase the feature maps' resolution. Figure 2.4 shows the UNet's architecture.

After each decoding operation, the next convolutional layer receives as input the concatenation along the channel axis of the upsampled and the downsampled feature map at the corresponding level.

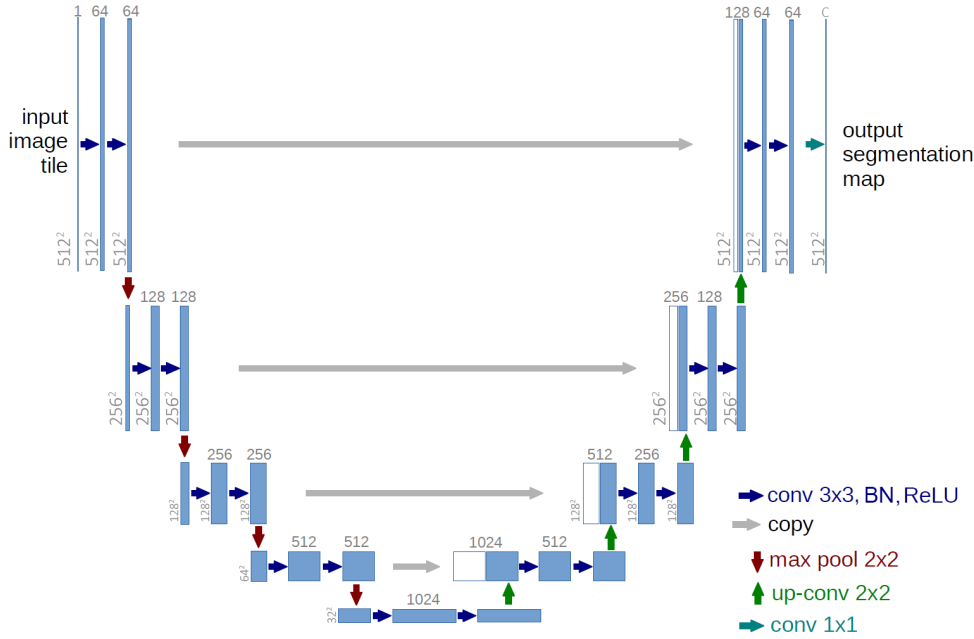


Figure 2.4: UNet architecture

CUMedVision1

CUMedVision1 [14] architecture (Figure 2.5) is composed of a single downsampling path, where 5 max-pooling layers halve the input image's resolution, and 3 up-sampling path. Each deconvolutional path is in charge of computing an output feature map of the same size as the input image where each single pixel is associated with a class, i.e. the upsampler is in charge of extracting the relevant features from the downsampled image. Each upsampling path is associated with a pixel-wise classifier to provide the output segmentation map and improve performances during the training process: through multi-task learning, better intermediate features are learnt such that the performances of the overall model are boosted.

The output feature maps of the decoders are merged together through a sum operation and a final Softmax non-linearity is applied.

CUMedVision2 - Deep Contour Aware Network

CUMedVision2 [14], also known as Deep Contour Aware Network, is an extension to CUMedVision1 architecture, as shown in Figure 2.6.

The green path corresponds to CUMedVision1 architecture. The blue path shares

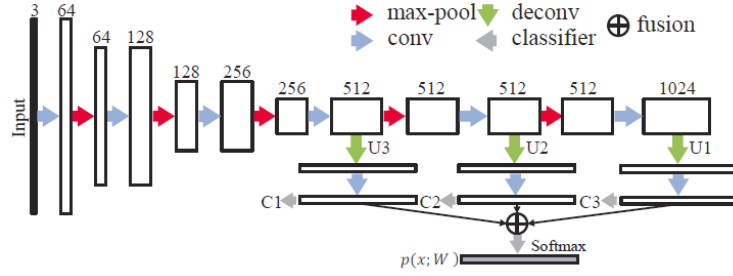


Figure 2.5: CUMedVision1 architecture. Source: [14]

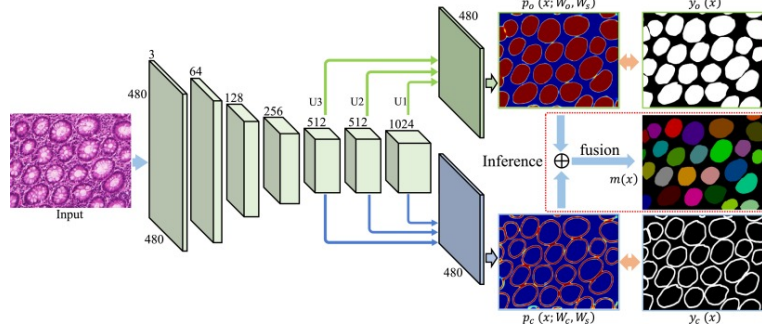


Figure 2.6: CUMedVision2 architecture. Source: [14]

the exact same architecture, with the only difference that the intermediate classifiers are trained to identify border pixels. Finally, information from both the paths are combined together in the output feature map $m(x)$ as described in [14] to generate the segmentation map.

Data Augmentation

The training process of Neural Networks is extremely computationally expensive and time consuming. Also, another big constraint for the development of Deep Learning models is the amount of input data required: in general, several hundreds or even thousands of images are required for the training process and to have the Neural Networks generalize fairly well. To introduce further variance and to avoid overfitting on the training dataset, data augmentation techniques are used in the context of image analysis and computer vision to dynamically generate new input images during the training phase. Some of the most common effects applied on input images are the following:

- random rotation;
- random vertical flip;
- random horizontal flip;
- random shear;
- random elastic distorsion;

2.3 Satellite imagery analysis

Satellite acquisitions acquire information not only among the visible bandwidths. In particular, Sentinel-2 mission (which is described more in depth in the following chapter) collects information at 13 different wavelengths. By combining them together, it is possible to obtain indexes useful for the identification of some specific areas, like burnt area or water presence. The definition of 4 burnt area indexes (BAI, BAIS2, NBR and NBR2) and 2 water presence indexes (WPI and MNDWI) are given with respect to Sentinel-2 bands, which are described in Table 2.1.

These indexes are computed by using spectral bands which are particularly sensible either to the presence of vegetation or water presence, as shown in Figure 2.7 [15]. For these reasons, B11 and B12 are used generally to compute water presence indexes, whereas a combination of B5, B6, B7 and B11/B12 are used generally to compute burnt area indexes. In case of burnt areas, the goal is to obtain an index which maximizes the reflectance in case of burnt vegetation (Figure 2.8) [16] [17]. Given an index, it is possible to identify a threshold value to generate a segmentation map which distinguishes between burnt and unburnt areas, such as [18]. One possible method to automatically identify the threshold value is the Otsu Method [19] used in image analysis, which minimizes the intra-class variance by maximizing the inter-class variance in a gray-scale image containing two classes.

Figure 2.9 shows the burnt area caused by a wildfire in Portugal with the comparisons of different indexes. The bottom-right corner of the image is absent due to missing satellite acquisition.

BAI - Burnt Area Index

$$\text{BAI} = \frac{1}{(0.1 - \text{B04})^2 + (0.06 - \text{B08})^2}$$

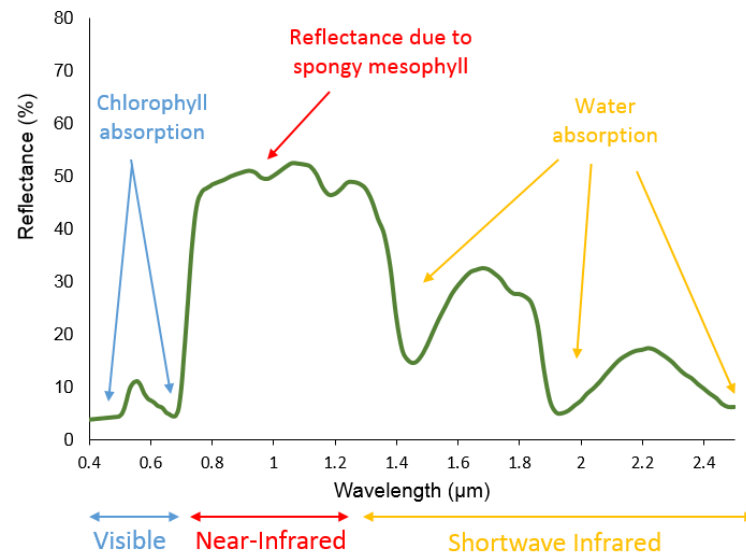


Figure 2.7: Bandwidths absorption. Source: [15]

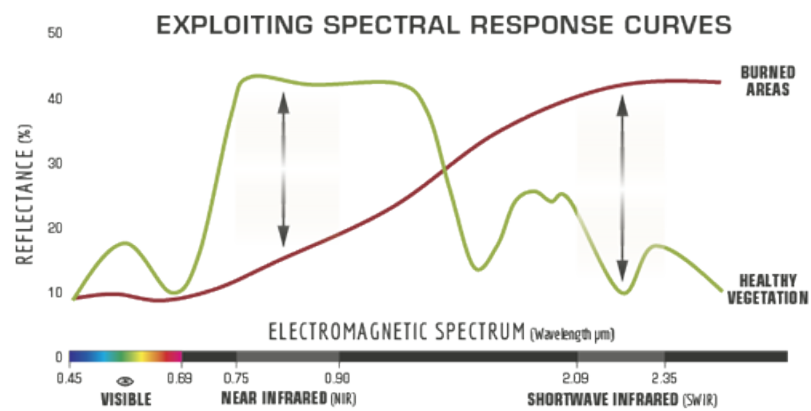


Figure 2.8: Healthy vegetation vs Burnt vegetation reflectance. Source: [17]

Band	Description	Central Wavelength (μm)	Spatial resolution
1	Coastal aerosol	0.443	20m
2	Blue	0.490	10m
3	Green	0.560	10m
4	Red	0.665	10m
5	Vegetation red edge	0.705	20m
6	Vegetation red edge	0.740	20m
7	Vegetation red edge	0.783	20m
8	Near Infrared (NIR)	0.842	10m
8A	Narrow NIR	0.865	10m
9	Water vapour	0.945	60m
10	Short wavelength infrared (SWIR)	1.375	60m
11	SWIR	1.610	20m
12	SWIR	2.190	20m

Table 2.1: Sentinel-2 bands description

BAIS2 - Burnt Area Index for Sentinel 2

Source: [20].

$$\text{BAIS2} = \left(1 - \sqrt{\frac{B06 \cdot B07 \cdot B8A}{(B04)}} \right) \cdot \left(\frac{B12 - B8A}{\sqrt{B12 + B8A}} + 1 \right)$$

NBR - Normalized Burnt Ratio

$$\text{NBR} = \frac{B08 - B12}{B08 + B12}$$

NBR2 - Normalized Burnt Ratio 2

Source: [21].

$$\text{NBR2} = \frac{B11 - B12}{B11 + B12}$$

WP - Water Presence Index

Source: [20].

$$WP = \frac{(B8A + B11 + B12) - (B01 + B02 + B03)}{(B8A + B11 + B12) + (B01 + B02 + B03)}$$

MNDWI - Modified Normalized Difference Water Index

Source: [22]

$$MNDWI = \frac{B03 - B11}{B03 + B11}$$

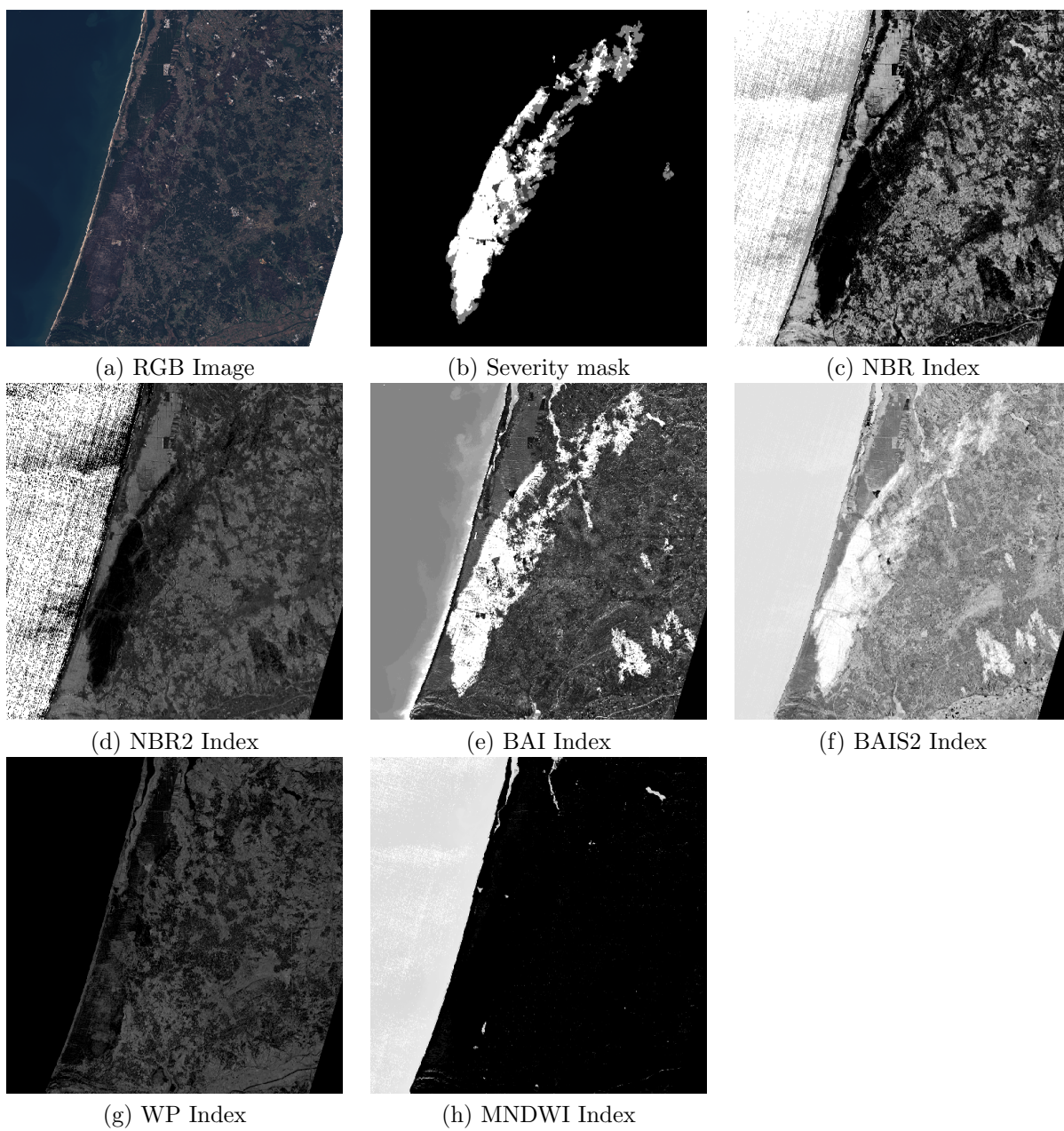


Figure 2.9: Wildfire in Mira, Portugal. Activation date: 15/10/2017

Chapter 3

Dataset and Tools

In the first part of this chapter a description on the different data sources and services used to download the dataset is given. In the second part, the set of tools and frameworks used for the development of the thesis is presented.

3.1 Data sources and services

3.1.1 Data sources

Information about catastrophic events and natural disasters are collected and processed by European Union and the European Space Agency (ESA) and made publicly available through the Copernicus Emergency Management Service (EMS) [23]. Different type of disasters are monitored: floods, volcanic eruptions, earthquakes, storms and fires. Each monitored event is identified through a code of this kind: EMSRXXX; each event may have one or more areas of interest, also known as bounding boxes. Events are also called activations and two different types of information may be available:

- delineation maps, which contain information about the burnt area;
- grading maps, which contain information about the burnt area and for each area a severity level is given from 0 (not burnt) to 4 (completely destroyed);

The activation maps from Copernicus are provided as shapefiles: a particular data format used for geospatial information. Geometrical information, such as points, lines and polygons, is contained in several files of different formats without any topological information. In particular, the following information are provided in the shapefile:

- identification of the area of interest with administrative boundaries and bounding box;
- delineation of hydrography of the entire area;
- delineation of settlements and populated areas;
- delineation of transportation maps and roads;
- delineation of the physiography of the area;
- delineation of the grading maps with the corresponding severity levels;

Among them, the bounding box and the polygons' definition of the grading maps were used: the first is defined with the geospatial coordinates of 4 vertices, which identifies the area of interest, and the second is defined with the array of the vertices composing the polygon. Each polygon or multipolygon is characterized with the same severity level. For a better understanding of the information contained in the shapefiles and to ease the automatic parsing of the information, shapefiles were converted into GeoJSON file format through the GDAL [24] translator library. The collected information represents the ground truth to be provided as input during the neural network training process, but it requires to be converted into a grayscale image. Also, the satellite data containing the topological information of the area of interest is not available from the shapefile but must be downloaded from SentinelHub services. The collection process is described in the following subsection.

3.1.2 SentinelHub and Data collection

SentinelHub [25] is an online service through which it is possible to visualize and download satellite acquisitions. Data collected from different missions and different satellites are available:

- Sentinel-1 mission includes two polar-orbiting satellites, operating day and night to collect information. Data collected from Sentinel-1 is used generally for land and ocean monitoring;
- Sentinel-2 mission provides high-resolution imagery in the visible and infrared bandwidths. Data collected by this mission is particularly used for soil and vegetation monitoring, as well as burnt area detection. The products provided by Sentinel-2 are divided in L1C, noisier and without any post-processing, and L2A, post-processed products to filter the atmosphere noise;

- Sentinel-3 mission’s goal is to measure sea topography, sea and land temperature and colours through the usage of 4 different sensors;
- Sentinel-5P mission collects data about air quality, ozone and UV radiation;
- Landsat mission (NASA), similar to Sentinel-2 systems both in terms of data collected and resolution;
- MODIS mission (NASA) collects data for land monitoring. MODIS mission does not differ too much from Sentinel-2 mission, except for the lower spatial resolution compared to the latter;

Sentinel-2 products were chosen for data collection as the bands collected by the satellites are useful for vegetation and burnt area monitoring. Furthermore, Sentinel-2 products offer higher resolution imagery compared to other missions such as MODIS and Landsat.

Sentinel-2 collects a total number of 13 bands, each with a different spatial resolution and characteristic.

SentinelHub service offers Cloud API keys to retrieve data, given the upper-left and bottom-right vertices of the bounding box, the time interval and the layer which determines which bands to retrieve. For each activation, satellite acquisitions 1 month prior the activation date and 1 month after the activation date have been collected by exploiting the information contained in the GeoJSON files. Each image is composed of 13 channels previously described. All the images were downloaded from SentinelHub service by forcing the smallest dimension between height and width to be 512 pixels, as the second dimension is computed automatically by the service.

For each activation event, Copernicus EMS provides the activation date, i.e. the date in which the forest wildfire started. Given the activation date, Sentinel-Hub was configured to retrieve only satellite imagery with a cloud coverage lower than 10% to limit the amount of noise in the downloaded images. All the available data 1 month prior and 1 month after the activation date were downloaded from the service.

3.1.3 Mask generation

Neural networks used for segmentation require as input satellite images and segmentation masks which associate per each pixel of the input image a class, ranging from 0 to 4. Images and masks must be of the same resolution. The severity mask generation process has been automatized through a python script, exploiting the information contained in the GeoJSON files and the resolution of the downloaded images. In fact, by parsing the GeoJSON files containing the crisis information, polygons and

Severity level	Pixel value	Class label
Completely Destroyed	255	4
Highly Damaged	192	3
Moderately Damaged	128	2
Negligible to slight damage	64	1
No damage	0	0

Table 3.1: Severity level to pixel color translation

multipolygons structures define burnt area and the corresponding severity levels. In particular, vertices are defined as (longitude, latitude) tuples with the corresponding severity level.

To generate severity masks as grayscale images, all the vertices coordinates were translated into pixels and the OpenCV library was used to draw polygons on a black image. Table 3.1 shows how severity levels were translated to pixel colors in range $[0, 255]$.

Given the mask, stored as a grayscale png image, a 2D matrix is obtained by analysing pixels' values. Each single pixel is classified with a label ranging in $[0, 4]$.

3.1.4 Data Cleaning

After mask generation and data retrieval through Sentinel-Hub online service, for each single activation, all the downloaded products were verified and checked for consistency together with the severity mask. Some wildfires events were discarded due to high presence of clouds, unavailable data or mask mismatch.

After the data cleaning process, a total number of 147 acquisitions were selected for the training and test procedures of neural network models.

3.1.5 Label distribution

Table 3.2 shows the percentage of pixels divided per each class. The segmentation problem is an imbalanced task: the majority of pixels belongs to unburnt areas, whereas among the 4 different severity levels, class 4 and class 3 are the most frequent.

3.2 Tools & Frameworks

In this section the different tools and frameworks used for the development of the algorithms of this thesis are introduced and described.

Class	Percentage
Class 0	85.23%
Class 1	1.78%
Class 2	3.46%
Class 3	4.53%
Class 4	5.00%

Table 3.2: Label distribution

3.2.1 GDAL

Geospatial Data Abstraction Library (GDAL) is an open-source library to read, parse and analyse different geospatial data file formats, such as shapefiles. GDAL software was used to convert automatically shapefiles into GeoJSON files through python scripts.

3.2.2 QGis

QGis is a Geographic Information System (GIS) application with a Graphical User Interface. It can open different geospatial data formats, visualize and manipulate vectorial and geospatial data format. Information is shown with different informative layers and delineation maps and it is particularly useful to comprehend the content of all the different shapefiles provided by Copernicus EMS.

3.2.3 Python

python is a high level programming language which supports both the object-oriented and functional programming paradigm. It is becoming one of the most popular and diffused programming language thanks to its simplicity and flexibility. It is also supported by a large variety of external libraries for web development, data analysis, machine learning, numerical computation and data visualization libraries.

3.2.4 OpenCV & skimage

OpenCV [26] and *skimage* [27] are open-source libraries for image processing. OpenCV is a computer vision library which offers APIs in C++, Java and python. It contains several algorithms to process images, identify objects and faces, eye-tracking algorithms, filters, possibilities to draw polygons but also functions to process images such as equalization tools and color models conversions.

skimage is another image processing library with similar functionalities as OpenCV. skimage was used for its contour-detection algorithms and for the implementation of data augmentation algorithms, such as vertical and horizontal flip, rotation and shear effects.

3.2.5 scikit-learn

scikit-learn [28] is an open-source library for machine learning. It contains several classes and tools useful for data analysis, supervised and unsupervised learning mechanisms, such as Decision Trees, Random Forests, Logistic and Linear Regressions, Support Vector Machines, normalization and dimensionality reduction techniques. scikit-learn also provides useful functions for computing class distribution and class weights in case of imbalanced classification problems, classes to perform train and test split and cross-validation.

3.2.6 PyTorch

PyTorch [29] is an open-source library for machine learning and numerical computation based on the Torch library. It offers a python and a C++ interface for development. The basic data type in PyTorch is the tensor which stores numerical information or model parameters. As the operations are performed on different tensors, a computational graph is dynamically built by the framework to keep track of all the functions applied to the input data. Then, the *Autograd* module is responsible for the automatic differentiation to compute the gradients in the backward pass during the training process of the neural network. Tensor operations can be handled by CPU or GPUs, thanks to the support of CUDA library and GPUs' hardware acceleration.

PyTorch is provided with the *Optim* module which implements several optimization algorithms to train neural networks. Models can be easily built using the *nn* module, which contains user-friendly interfaces and classes to easily build complex neural network models and concatenate different layers together.

Furthermore, PyTorch offers great flexibility for dataset loading operation and preprocess: it is possible to define an input pipeline to alter, augment and normalize the data.

Chapter 4

Methodology

In this chapter the description of the approaches and the analysis performed are given, starting from a preliminary study on the input data. Then, the input pipeline for data preprocessing and data augmentation is described to enhance the capabilities of neural networks to generalize and avoid overfitting issues. Finally, possible improvements to the networks' architecture and the quality measures used in the Performance Evaluation chapters are explained.

4.1 Problem interpretation and approaches

Each satellite acquisition is coupled with a grayscale mask, generated as explained in Chapter 3. In particular, 4 qualitative severity levels are available. Thus, the segmentation problem can be interpreted in two different ways:

- a simpler binary problem, distinguishing between burnt or not burnt area;
- a more complex multiclass problem, with an increasing severity level: the higher it is, more severe the damage is;

Masks can be easily converted to the binary case by simply grouping together classes 1, 2, 3, 4.

It is obvious that passing from the binary case to the multiclass problem, the complexity of the problem increases. In fact, the dataset presents an evident problem of class imbalance considering the burnt/unburnt area ratio. Class imbalance becomes even more evident when considering the severity level, as can be seen in Table 4.1 and Table 4.2.

	Class 0	Class 1
Occurrence	53,699,522	9,255,486
Percentage	85.30%	14.70%

Table 4.1: Class imbalance in binary case

	Class 0	Class 1	Class 2	Class 3	Class 4
Occurrence	53,699,522	1,323,767	2,485,435	2,830,252	2,616,032
Percentage	85.30%	2.10%	3.95%	4.50%	4.15%

Table 4.2: Class imbalance in multi-class case

Imbalanced class issues can be relieved by using proper loss functions, such as Dice Loss or a Weighted Cross Entropy Loss, where each class is inversely proportionally weighted to its occurrence in the dataset, thus penalizing more misclassification errors performed on the minority classes.

It is worth noticing that all the neural network architectures considered in this work were created in the context of biomedical image segmentation, where cells ([14], [3], [4]) and organs ([5], [30], [31]) or other structures ([32]) are segmented. Two main differences are that:

- cells and organs are characterized by well-defined shapes and their contours are defined by cell membrane or tissues, while burnt area have undefined shapes;
- cells and organs are not interpenetrating each other, while burnt area of different severity levels may be overlapped together within the same region (Figure 4.1);

It is worth noticing that the original UNet implementation used a CrossEntropy loss combined with a pixel-wise weight map, which penalized more errors made in contour pixels, i.e. cell membranes. Such approach is not suitable in this context due to the extremely variability of burnt areas: some regions present great amount of damaged areas, some other monitored activations present smaller regions represented with few pixels. Computing weight maps in these latter cases would lead to weight maps with all values near zero, thus making training neural networks more difficult. Since Sentinel-2 satellite acquisitions are obtained by analysing 13 different bandwidths, it is possible to perform the prediction of burnt areas by considering a subset of the original feature space. In this thesis, two evaluations are performed:

- a first analysis based on low-cost sensors, i.e. RGB sensors;

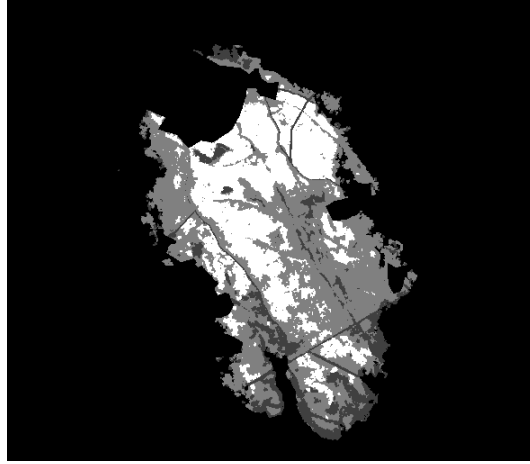


Figure 4.1: Severity mask with overlapping regions of different severity levels

- a second analysis based on the entire feature set;

So, neural networks were trained considering the two different input feature sets.

4.2 Preliminary analyses

Given the premises about class imbalance and overlapping regions previously introduced, it is important to perform a preliminary analysis on the grade of separability between the 5 different classes to evaluate possible difficulties during the training process of neural networks.

4.2.1 Analysis on burnt area indexes and severity levels

Satellite imagery analysis and in particular burnt area detection relies on Burnt Area Index (BAI) or Normalized Burnt Ratio (NBR) and their variants, particular combinations of different bandwidths such that the resulting greyscale image enhances the presence of burnt areas. The quality of an index must be evaluated through the Separability Index (Equation 4.1), which quantifies how well two classes are separated.

$$SI = \frac{|\mu_a - \mu_b|}{\sigma_a + \sigma_b} \quad (4.1)$$

The higher the Separability Index is, more separable are the classes. The indexes were computed for both the binary case and the multiclass case for all the burnt area

indexes (NBR, NBR2, BAI, BAIS2). The separability indexes between burnt and unburnt area pixels are shown in Table 4.3.

	NBR	NBR2	BAI	BAIS2
SI	0.553	0.928	0.773	0.862

Table 4.3: Separability indexes in the binary scenario

The results show that state of the art's indexes for burnt area detection are effective since SI value is almost 1. Thus, it is expected to see good performances for neural network models in the binary segmentation scenario.

This analysis was extended to the multiclass scenario by computing the SI between all possible combinations of classes. The results are shown in Figure 4.2.

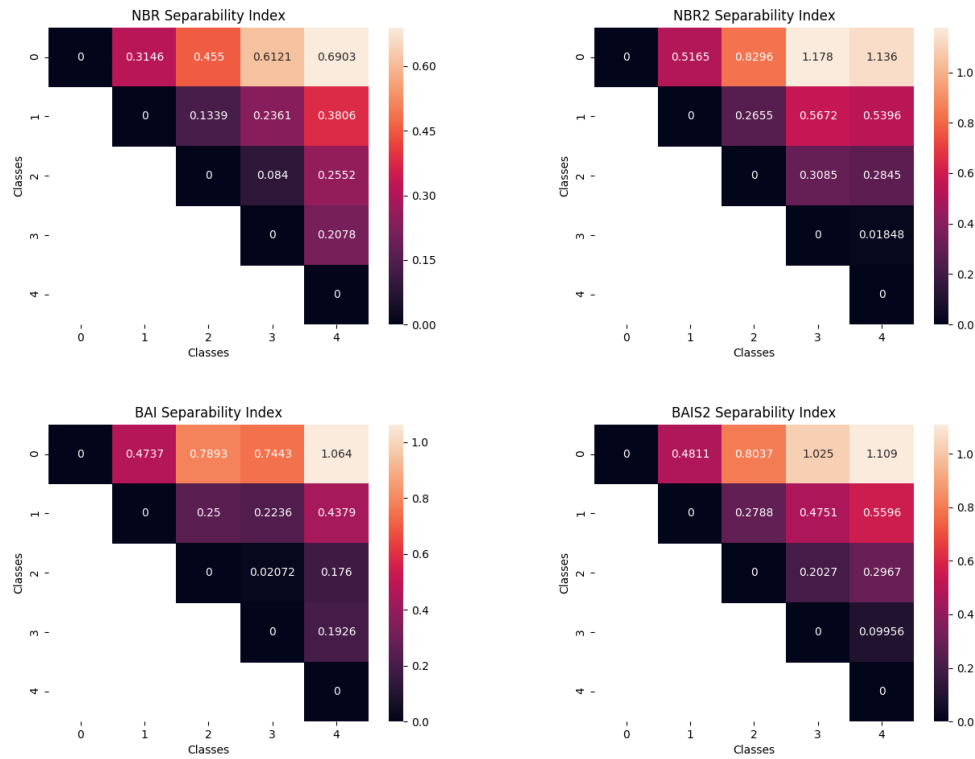


Figure 4.2: Separability indexes in multiclass scenario

All the 4 indexes lead to the same results: a good separability is obtained by considering class 2 vs class 0, class 3 vs class 0 and class 4 vs class 0. In contrast,

the separability index between unburnt areas and slightly damaged areas is not very high, thus underlining the difficulty in automating this task.

The most relevant outcome of this analysis is the extremely low degree of inter-class separability within burnt regions: the highest index value is 0.25 and is observed between class 1 and class 2 for BAIS2 and BAI and 0.3 between class 2 and class 3 for NBR2 and the lowest value is lower than 0.1 between class 3 and class 4.

This analysis underlines the impossibility to detect and to estimate with a reasonable confidence the severity level using the state of art indexes for burnt area detection. It is important to notice that the procedures and the data used for the severity level estimation by Copernicus EMS and the companies in charge are not known: it is possible that in-situ information are collected for this purpose.

Furthermore, it is worth noticing that in NBR2 index, class 3 presents higher separability indexes than class 4 with respect to unburnt areas. Such index emphasizes more burnt areas with severity level 3 than severity level 4, thus underlining the possibility that the distinction from "Highly Damaged" and "Completely Destroyed" may not be so clear, at least solely using information obtained from the 13 bands.

4.2.2 Analysis on 13 bands

The previous analysis considered pixel values obtained by computing indexes known in state of the art for burnt area detection. It is possible that burnt area inter-class overlapping issues may be relieved by consider the problem in 13 dimensions, i.e. all the available bands of Sentinel-2 L1C mission. To perform this evaluation, the Principal Component Analysis (PCA) [33] has been used.

First, all the images with the corresponding masks were read to generate a $n \times p$ matrix, where n is the number of pixels and p is the number of features per pixels, i.e. 13 bands. Given the matrix, PCA is computed with all the 13 components. From Figure 4.3, it can be seen that with the first two and three principal components the explained variance ratio reaches 82% and 91% respectively, so it is possible to plot the pixels with the respective class in a 2D and 3D chart (Figure 4.4). Scatterplots represent a sample of 150.000 pixels per class for all the 5 classes due to rendering issues.

Given the scatterplots and the results from the PCA analysis, it is likely that the 5 severity classes are overlapped and hardly separable in the feature space of the 13 bands.

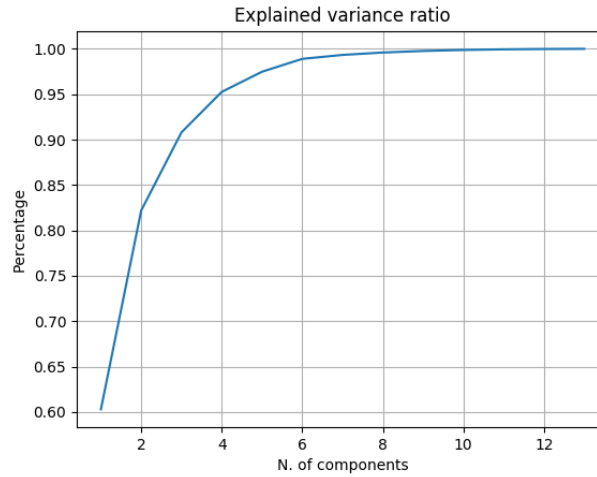


Figure 4.3: Explained variance ratio

4.3 Data Preprocessing and input pipeline

Before providing data to neural networks, it is important to build an input pipeline to preprocess and alter images such that models receive the input in the correct format and with values within proper ranges. Figure 4.5 represents the entire input pipeline.

The training input pipeline operates as follows:

1. all training images are read as numpy arrays and resized: each activation is an image composed of 3 or 13 channels and normalized with a min-max normalization. Each image was resized to 512x512;
2. for each training image, the corresponding severity mask is loaded from a greyscale png image;
3. a class transformer component is in charge of converting greyscale images into multi-class masks, where each pixel is associated with a class in range $[0 - 4]$ or $[0 - 1]$;
4. the tuple (image, mask) is fed to a transformation pipeline so defined (Figure 4.6):
 - (a) ToBandsMapper and BandsAggregator are two classes used to easily select certain spectral bands and eventually combine them. The aggregator

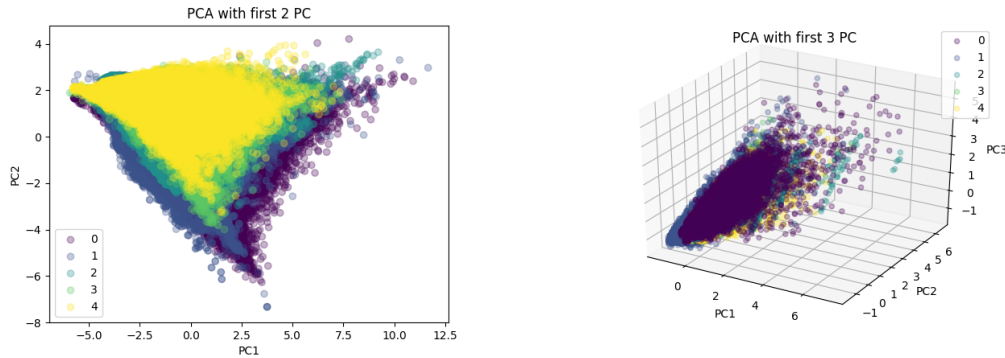


Figure 4.4: Scatterplots of pixels with PCA

class returns a new multi-channel image, generated by executing a lambda expression. These modules are present only for multi spectral bands input;

- (b) Random Rotate randomly applies a rotation within a specified range to both the input image and mask;
- (c) Random Horizontal/Vertical Flip randomly applies a flip on the corresponding axis to both the input image and mask;
- (d) Random Shear randomly applies a shear mapping with an angle within a specified range to both the image and mask;
- (e) ToTensor converts a batch of images with axes $[N, H, W, C]$ to PyTorch's convention $[N, C, H, W]$;
- (f) Normalize maps the input image from range $[0, 1]$ to range $[-1, 1]$;

5. the input images are used for training the model;

6. the mask prediction obtained by neural network model is compared to the ground truth via the loss function, which is used to update the model's weights and biases;

The transformation pipeline corresponds to the data augmentation process [34] to generate each time a different input set to lead neural networks to generalize better. The parameters for the transformation pipeline are reported in Table 4.4.

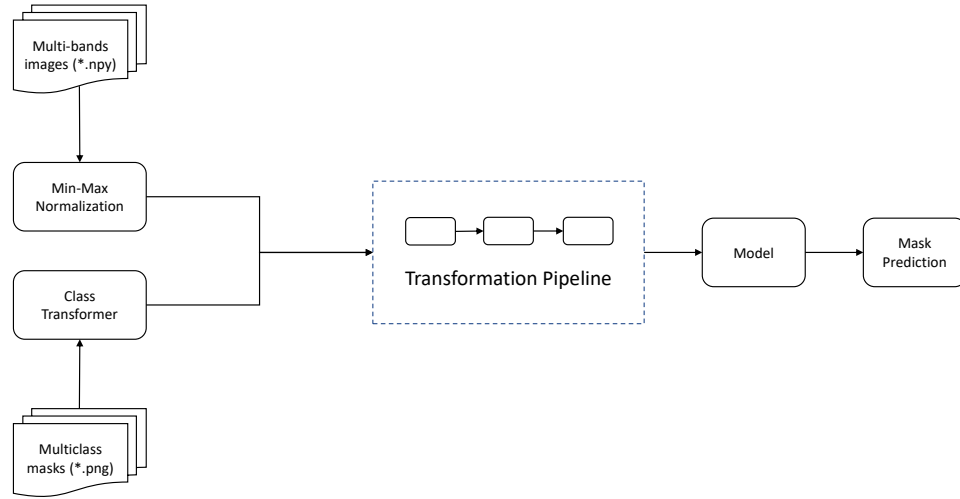


Figure 4.5: Input pipeline during training process

	Probability	Angle (degree)
RandomRotate	0.5	$[-50, +50]$
RandomHorizontalFlip	0.5	/
RandomVerticalFlip	0.5	/
RandomShear	0.5	$[-20, +20]$

Table 4.4: Data augmentation parameters

4.4 BandsAggregator and training process

The BandsAggregator class was designed to easily change the number of input features and thus the number of activation's input channels. Through PyTorch's transformers, this process is done on-the-fly as data is provided to the network.

The greatest advantage of this mechanism is obtained when the neural network is trained by considering all the input feature space: instead of providing all 13 spectral bands to the model, an incremental approach was used. In particular, first the spectral bands used in the different burnt area indexes were provided and then, one by one each of the spectral bands was added till all the input feature space was used. This process was performed to understand if and to which extent each single

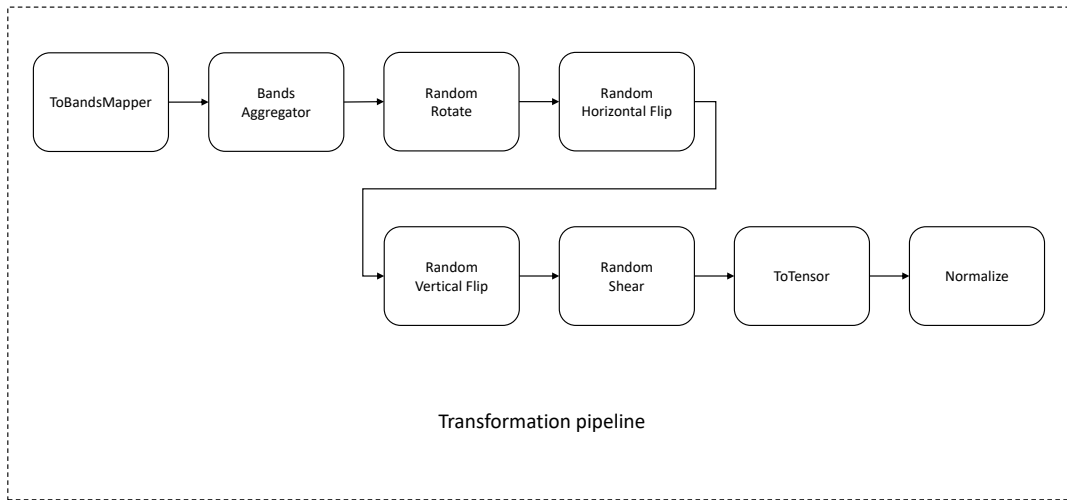


Figure 4.6: Transformation pipeline

feature contributed to the prediction in case significant improvements were observed.

4.5 CUMedVision2 and contour generation

CUMedVision2 model has two different branches, which are fused together to get the prediction outcome: the first is in charge of segmenting the input image, whereas the second branch is trained to determine the contour pixels of burnt area regions. Thus, the training process requires, together with the input image and the ground truth, a third image which represents all the contour pixels. This image is dynamically generated using the ContourGenerator class, which uses scikit-image library to determine the boundaries of all the burnt regions and applies a disk filter of radius 1. The pipeline used to feed the CUMedVision2 network is shown in Figure 4.7.

4.6 Input features - Incremental approach

As described in the previous section, an incremental approach was chosen to evaluate the benefits of adding more input features to the neural networks. In particular, two

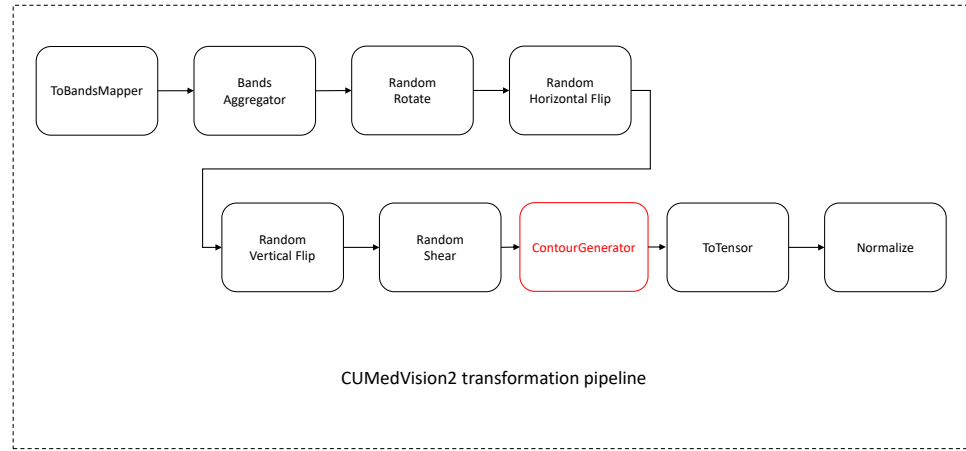


Figure 4.7: CUMedVision2 input pipeline

neural networks were trained using the RGB input and the single-channel BAIS2 image. Then, multiclass UNet models with Dice Loss were trained. By incrementally adding more input bands, the network is able to compensate the absence of data (example in Figure 4.8) and properly recognize water as unburnt areas. Unfortunately, regions with water presence or absent data are extremely limited in the dataset and thus, the ability of the network to properly learn how to distinguish these elements in the satellite image is strictly related to the train/test set partitioning.

In all the training procedures class 1 and class 3 never or rarely got activated in case of UNet models. In terms of shape of the burnt area, results were comparable and thus the 13-channel multi spectral input has been chosen for the multiclass-multibands training procedure.

4.7 Comparison with data before and after the activation event

A final approach was considered by providing as input to the neural network two multi spectral bands input: one image before the activation event and one image after the conclusion of the catastrophic event. This operation was performed to let the neural network learn by comparing the satellite imagery prior and after the

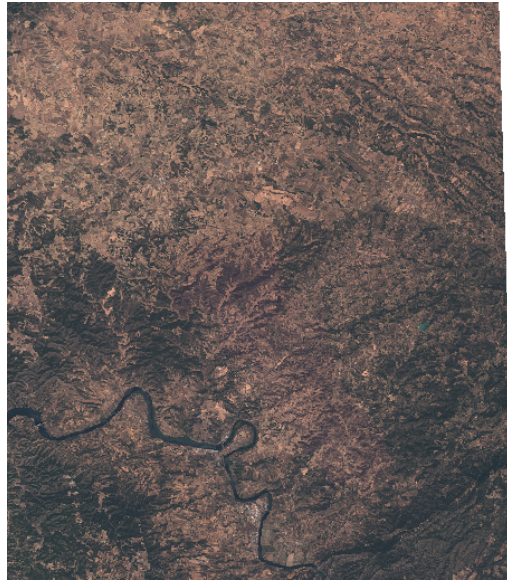


Figure 4.8: Example of an image with missing information obtained through SentinelHub service

wildfire. The benefits of this approach could be the followings:

- increased robustness towards mountainous areas: the presence of shadows and the noise introduced by the morphology of the considered area should be reduced when considering a differential approach, in which two images are compared;
- increased robustness towards already-burnt areas: in few cases, images with evident burnt regions prior the activation are present in the dataset, which severity level was not monitored Copernicus EMS, thus introducing noise and further difficulties when training the system;
- increased performances in the multiclass case, due to the possibility to compare information about vegetation presence before and after the event;

It is important to notice that in this thesis, no approaches for interpreting neural network models were considered and thus it is not known which are the most relevant features which led the model to take a certain decision. By providing two multispectral bands images to the network, it is expected to achieve better performances for the reasons previously considered.

Due to the time required for training and UNet achieving better performances over CUMedVision1 when considering the shape of the burnt area regions, only multiclass UNet model was extended to receive as input the two images. Two UNet models were trained: one for the RGB case (6 channels) and one for the multi spectral input case (26 channels).

4.8 Quality measures

In Machine Learning it is fundamental to assess the developed model and its output through different metrics, according to the type of problem and the class distribution. In this section, different evaluation metrics which were used for models' assessment are described, both in binary classification problems and in the multi-class case. Table 4.5 and Table 4.6 represents the confusion matrices for both the binary classification and multi-class classification problems. Conventionally, in the binary case the positive class is attributed to burnt areas.

		Ground truth	
		True	False
Predicted	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Table 4.5: Binary confusion matrix

4.8.1 Accuracy

Accuracy measures the percentage of correctly classified data points with respect to the total number of samples. In the binary case it is computed as follows:

		Ground truth				
		Class 0	Class 1	Class 2	Class 3	Class 4
Predicted	Class 0	m_{00}	m_{01}	m_{04}
	Class 1	m_{10}	m_{11}			m_{14}
	Class 2	...		m_{22}		...
	Class 3	...			m_{33}	...
	Class 4	m_{40}	m_{44}

Table 4.6: Multi-class confusion matrix

$$Acc = \frac{TP + TN}{TP + FP + FN + TN}$$

It is generalized as follows in the multi-class case:

$$Acc = \frac{\sum_{i=0}^4 m_{ii}}{\sum_{i=0}^4 \sum_{j=0}^4 m_{ij}}$$

Accuracy evaluation metric cannot be used as a reliable evaluation metric for this thesis due to class imbalance: in fact, a simple classifier that always predicts class 0 would reach an accuracy of 85.30% on the entire dataset. For this reason, it is important to introduce other evaluation metrics.

4.8.2 Precision

Precision for class i measures the amount of correctly classified data points in class i with respect to total number of predicted samples in i . In the binary case, precision is measured on the positive class as follows:

$$precision = \frac{TP}{TP + FP}$$

For multi-class problems, precision can be computed for each of the class as:

$$precision_i = \frac{m_{ii}}{\sum_{j=0}^4 m_{ij}}$$

4.8.3 Recall

Recall for class i measures the amount of correctly classified samples in class i with respect to number of samples actually belonging to class i . For binary problems, recall is computed as follows:

$$recall = \frac{TP}{TP + FN}$$

The recall metrics can be extended to multi-class problems by computing it for each single class.

$$recall_i = \frac{m_{ii}}{\sum_{j=0}^4 m_{ji}}$$

4.8.4 F1 score

F1 score combines together recall and precision through a harmonic mean so defined:

$$F1_i = 2 \cdot \frac{precision_i \cdot recall_i}{precision_i + recall_i} \quad (4.2)$$

In binary classification problems, F1 score is generally computed for the positive class, as for multi-class problems, the metric is computed for each class. In the latter case, a macro-F1 score can be computed as the (weighted) mean of per-class F1 score (Equation 4.3 and 4.4, where N is the total number of data points and n_i is the number of points belonging to class i).

$$\text{macro-F1} = \frac{1}{4} \cdot \sum_{i=0}^4 F1_i \quad (4.3)$$

$$\text{weighted-F1} = \frac{1}{N} \cdot \sum_{i=0}^4 n_i \cdot F1_i \quad (4.4)$$

4.9 Validation procedure

For validation purposes, it is important to provide neural network models test images which do not overlap with the bounding boxes of training images. This is to overcome overfitting issues: if the neural network is overfitting and it is learning the morphology of each of the training input images and overlapping regions with test images are present, then validation losses and metrics are not valid. Thus, to validate the developed models a cross-validation approach has been implemented, which is shown in Figure 4.9.

Given the entire dataset composed of 51 activations with a total number of 147 input images, the 7-fold cross validation is performed as follows: at each iteration, a group of activation events is chosen as test set with all its images, as the remaining 6 groups of activations are used for the training process. Activations are grouped together such that each fold contains (almost) the same amount of test images and there are not overlapping regions between the train and test set. The training set is enriched with synthetic data through data augmentation, whereas test set is not. As the training process is completed, evaluation metrics are computed on the test set: accuracy, per-class recall, precision and f1-score are computed. Then, the mean value is computed for all the metrics. The groups are defined in Table 4.7.

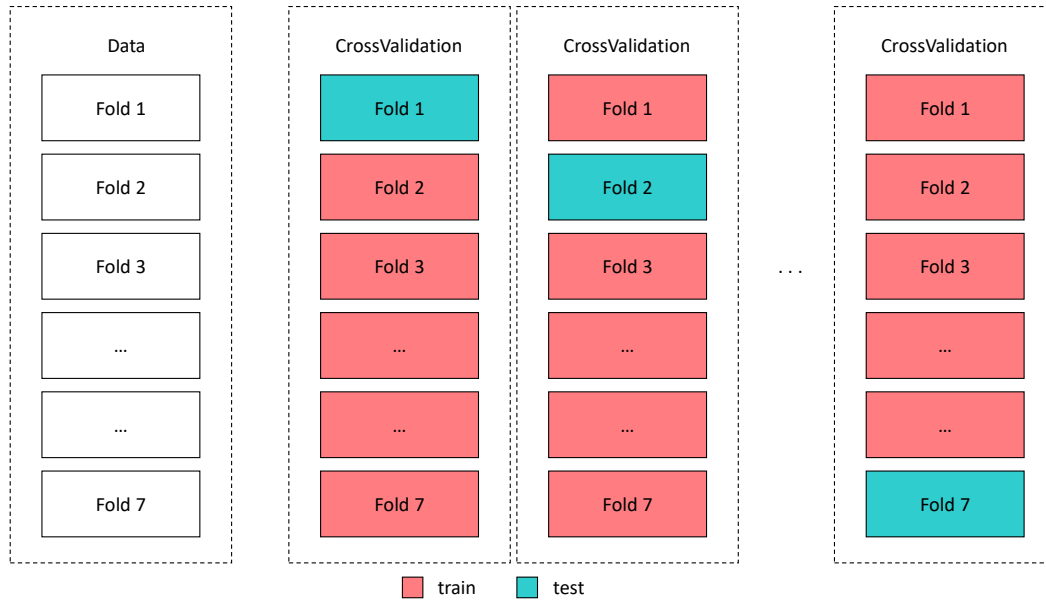


Figure 4.9: CrossValidation procedure

4.10 Training system

Due to the high amount of parameters and images to analyse, it was not possible to perform the analysis on a simple local machine. Furthermore, to train neural networks in reasonable amount of time, a CUDA-accelerated system is fundamental. Table 4.8 describes the main hardware characteristics and software packages.

Fold #	Activation codes	Test set size (images)
1	EMSR370, EMSR299, EMSR252, EMSR288, EMSR316, EMSR365, EMSR344, EMSR363, EMSR303, EMSR362, EMSR278, EMSR373, EMSR371, EMSR247, EMSR221, EMSR173, EMSR360, EMSR331, EMSR307, EMSR367, EMSR230	21
2	EMSR207, EMSR133	21
3	EMSR169, EMSR209, EMSR290	21
4	EMSR213, EMSR217, EMSR302	20
5	EMSR239, EMSR237, EMSR214, EMSR219, EMSR368, EMSR305, EMSR353, EMSR175, EMSR295	21
6	EMSR369, EMSR171, EMSR250, EMSR248, EMSR210, EMSR254, EMSR211	21
7	EMSR132, EMSR216, EMSR227, EMSR298, EMSR291, EMSR300	22

Table 4.7: Definition of folds for cross-validation

Component	Model/Version
CPU	Intel Core i9-7940X @ 3.10GHz
RAM	128GB
GPU	4x GTX 1080Ti 12GB
Nvidia Drivers	390.116
CUDA	release 9.0, V9.0.176
Python	3.6.7
PyTorch	1.1.0
torchvision	0.3.0

Table 4.8: System configuration

Chapter 5

Performance Evaluation

In this chapter the experimental results of the developed neural network models are presented. This chapter is structured as follows: first, UNet, CUMedVision1 and CUMedVision2 models are trained and tested on RGB images by considering both the binary and the multi-class case. Then, the same training and testing procedures are performed on the entire dataset with the 13 input spectral bands. An input example with the corresponding ground truth is shown in Figure 5.1. Then, two comparison tables are shown in Section 5.3 to compare the different models prediction, both in binary and multiclass case.

For binary classification problems, accuracy, recall, precision and F1-score are presented with mean value obtained across the 7 folds, best and worst value obtained for each metric among the 7 folds. Best and worst values of two different metrics may be obtained from two different folds.

The same analysis is performed in the multiclass case: recall, precision and F1-score are computed per each class and mean, best and worst values obtained from the 7 different folds are reported.

In some experiments, a higher number of epochs was chosen with a variable learning rate to stabilize the models prediction. In particular, a MultiStepLR Scheduler has been used with the steps and gamma parameters specified in the following tables.

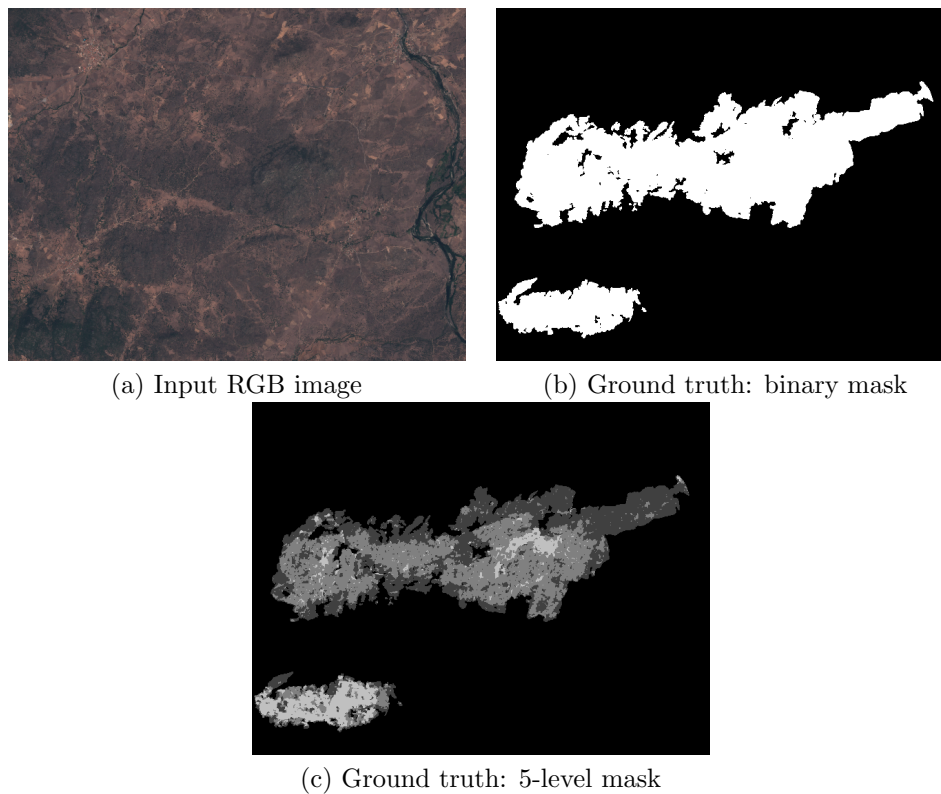


Figure 5.1: Input example for the RGB case

5.1 RGB input

5.1.1 UNet

Binary case

Parameters used for the training process on each fold are reported in Table 5.1.

Parameter	Value
Batch size	8
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Binary Cross-Entropy

Table 5.1: UNet training parameters in binary case, RGB input

	Accuracy	Recall	Precision	F1-score
Mean	90.43%	0.698	0.706	0.665
Best	96.30%	0.934	0.899	0.886
Worst	82.38%	0.390	0.492	0.518

Table 5.2: UNet model evaluation metrics in binary case, RGB input

The basic UNet model in the binary case achieved a mean F1-score of 0.665, with a best score of **0.886**. Figure 5.2 shows model outputs obtained from the best and the worst performing folds respectively.

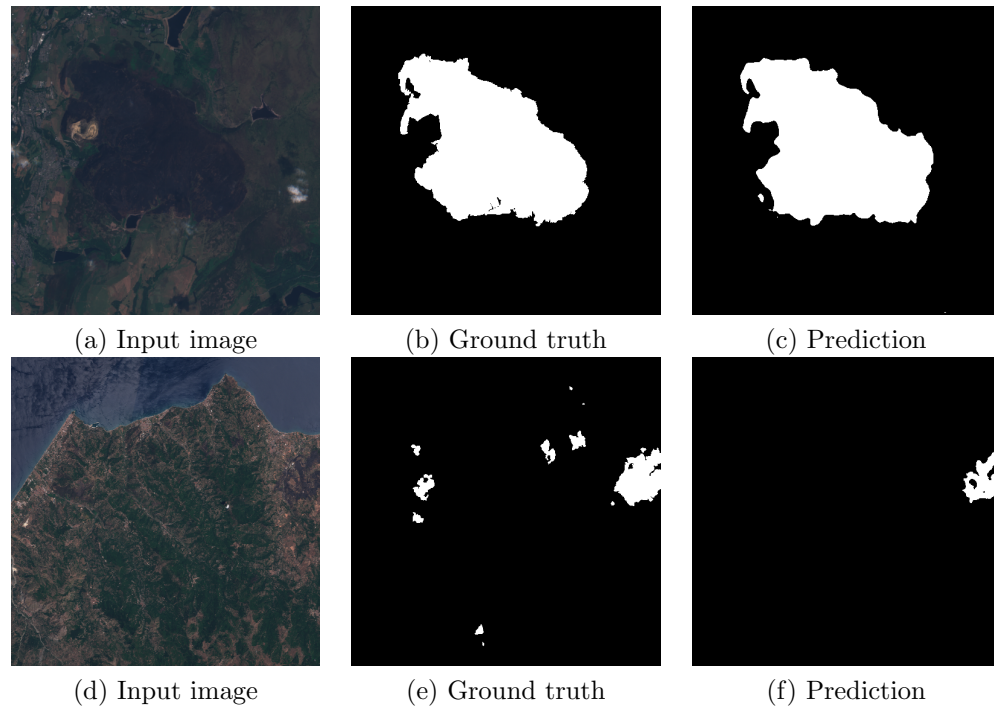


Figure 5.2: Binary UNet RGB predictions from best and worst performing folds

Most errors are related to small burnt areas that the UNet is not able to detect. Minor misclassification errors are seen in the contour areas, while burnt areas with greater extension are detected more precisely. Also, minor errors are performed within small unburnt areas surrounded by major burnt regions, presence of clouds and volcanic areas.

Light conditions affect prediction outcomes and training process: depending on the time of the day at which the satellite acquisition is made and the morphology of the area, the presence of shadows caused by mountainous areas may lead the neural network to label undamaged areas as burnt.

Multi-class case

Parameter	Value
Batch size	8
Epochs	50
Optimizer	Adam
Learning rate	0.01
Weight decay	0
Scheduler	MultiStep
Steps	Epochs: 25, 35, 45
Gamma	0.1
Loss function(s)	Dice Loss

Table 5.3: UNet training parameters in multiclass case, RGB input

		Recall	Precision	F1-score
Class 0	Mean	0.842	0.954	0.889
	Best	0.951	0.986	0.949
	Worst	0.632	0.901	0.769
Class 1	Mean	0	/	/
	Best	0	/	/
	Worst	0	/	/
Class 2	Mean	0.350	0.136	0.177
	Best	0.676	0.262	0.277
	Worst	0	0.043	0.075
Class 3	Mean	0	/	/
	Best	0	/	/
	Worst	0	/	/
Class 4	Mean	0.496	0.207	0.187
	Best	0.769	0.694	0.295
	Worst	0.054	0.056	0.100

Table 5.4: UNet model evaluation metrics in multiclass case, RGB input

The extension of UNet to the multiclass case is performing poorly in terms of F1-score with respect to the different burnt severity levels considering the RGB input. Class 1 and Class 3 layers never activated and thus recall, precision and F1-score values are not valid.

Two prediction outcomes are shown in Figure 5.3: Example 1 shows a case where UNet was able to determine the general shape of the burnt area, but it was unable to correctly estimate the severity level; Example 2 shows the prediction outcome influenced by the presence of crops.

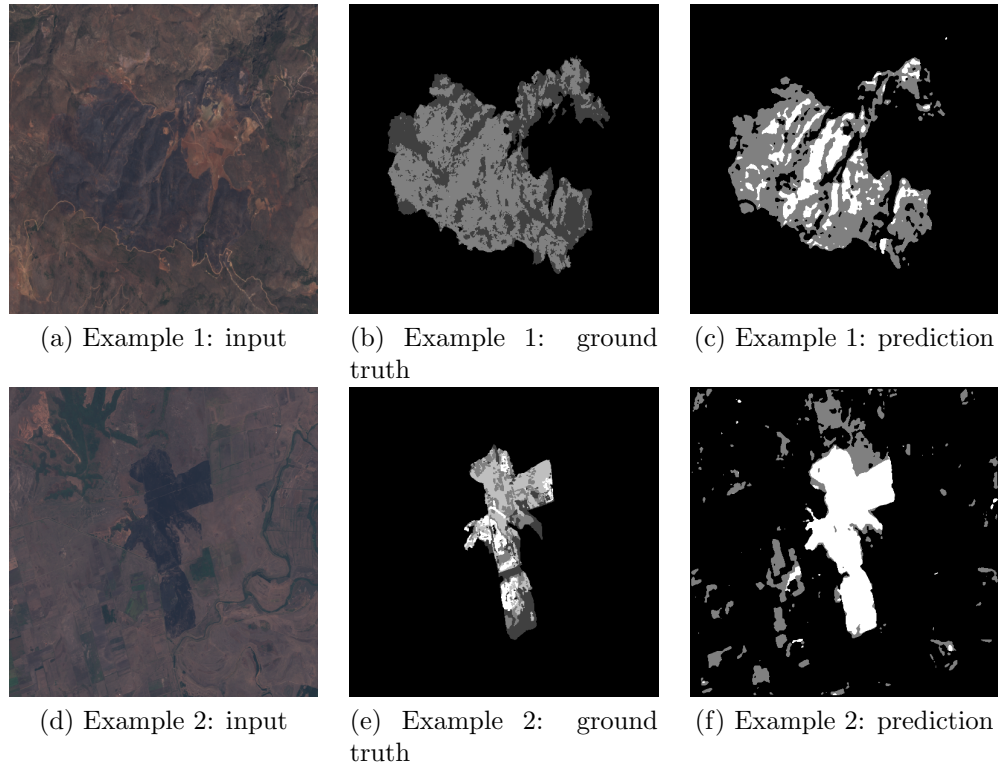


Figure 5.3: Examples obtained from multi-class UNet with RGB input

In Example 2 case, the presence of cultivated areas and light conditions negatively influence the prediction outcomes, due to high presence of areas with dark brown color. In this case, UNet was not able to detect severity levels and overestimated the burnt area region.

5.1.2 CUMedVision1

Binary case

Parameter	Value
Batch size	16
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Binary CrossEntropy

Table 5.5: CUMedVision1 training parameters in binary case, RGB input

	Accuracy	Recall	Precision	F1-score
Mean	90.40%	0.688	0.678	0.647
Best	94.56%	0.944	0.955	0.813
Worst	0.874	0.432	0.398	0.536

Table 5.6: CUMedVision1 evaluation metrics in binary case, RGB input

CUMedVision1 model performances are in general lower compared to binary UNet. In particular, the network is able to detect the burnt area inside the image, but its shape and the contour areas are not as precise as UNet, with major errors obtained in the worst case, as shown in Figure 5.4 Example 2. The absence of concatenation layers in the network architecture is providing a lower amount of information to determine the burnt area more precisely.

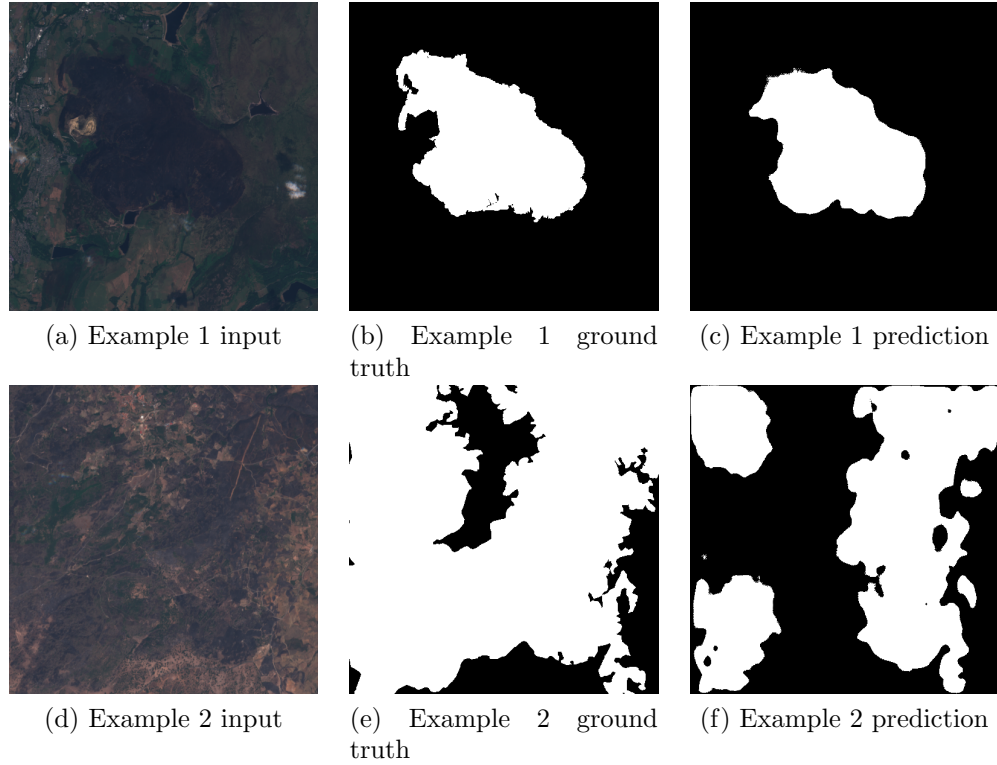


Figure 5.4: Examples obtained from binary CUMedvision1 with RGB input

Multi-class case

Parameter	Value
Batch size	8
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	CrossEntropy

Table 5.7: CUMedVision1 training parameters in multiclass case, RGB input

		Recall	Precision	F1-score
Class 0	Mean	0.938	0.910	0.921
	Best	0.997	0.971	0.968
	Worst	0.794	0.797	0.874
Class 1	Mean	0.026	0.033	0.016
	Best	0.145	0.138	0.077
	Worst	0	0	0
Class 2	Mean	0.039	0.078	0.041
	Best	0.177	0.179	0.155
	Worst	0	0	0
Class 3	Mean	0.112	0.145	0.075
	Best	0.557	0.285	0.196
	Worst	0.011	0.028	0.016
Class 4	Mean	0.382	0.257	0.245
	Best	0.590	0.797	0.469
	Worst	0.185	0.049	0.090

Table 5.8: CUMedVision1 evaluation metrics in multiclass case, RGB input

Differently from the multiclass UNet, all the classes were present in the prediction performed on the test set with CUMedvision1, though the recall, precision and f1-score are fairly low on the different burnt severity levels. Figure 5.5 shows 3 different examples on prediction outcomes.

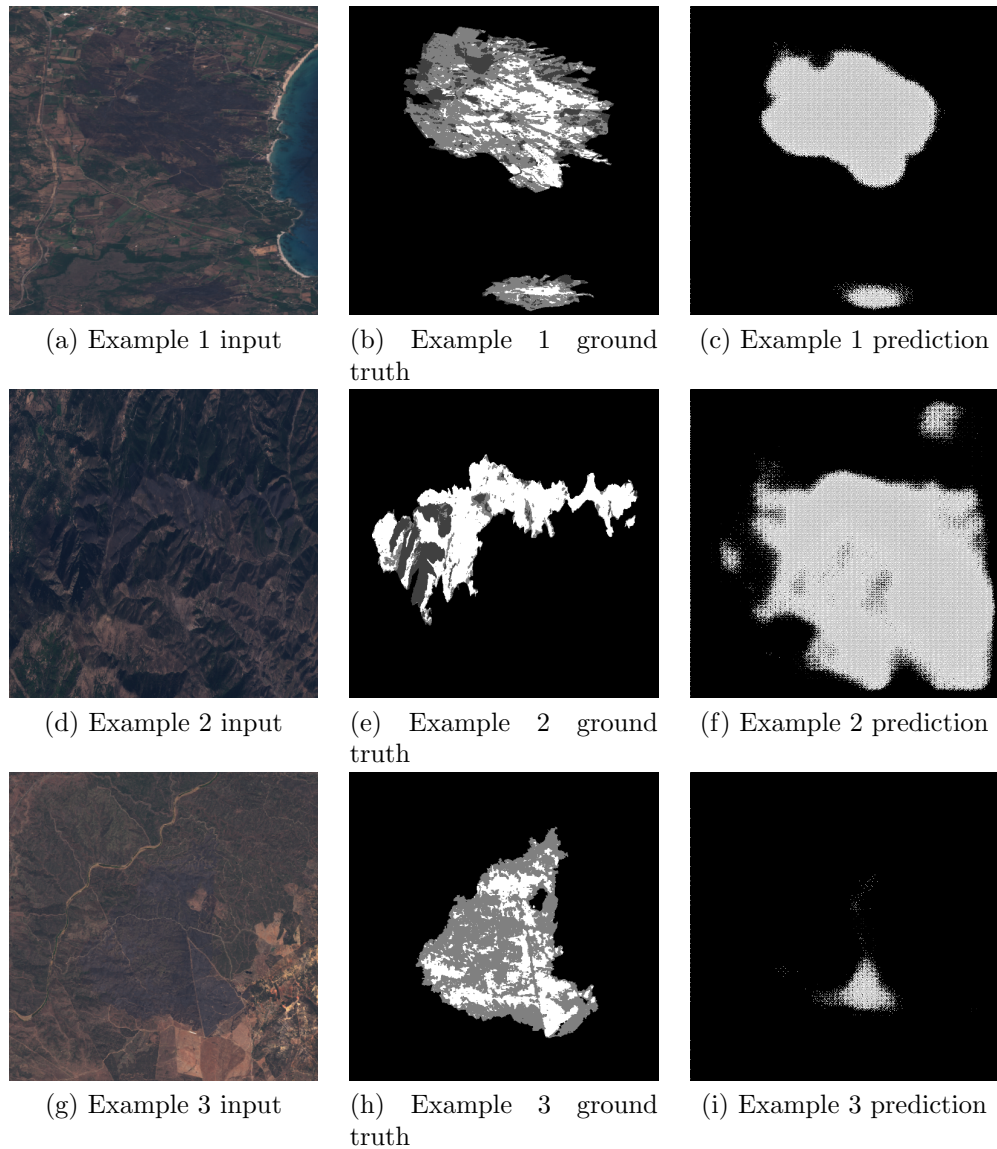


Figure 5.5: Examples obtained from multiclass CUMedvision1 with RGB input

Example 1 shows a case where the network is able to detect the burnt region general shape, with great errors made in contour areas. Example 2 and Example 3 show two different cases in which CUMedVision1 is making major mistakes:

- Example 2: the network is greatly overestimating the burnt area region due to light conditions, shadows and mountainous areas;

- Example 3: burnt areas are almost absent in prediction, even though they are clearly visible from the RGB image;

Furthermore, in all the three prediction outcomes, areas with different severity levels are not clearly identified. Instead, the network is computing a grid pattern in which different severity levels are present, thus leading to a slight increase in F1-score with respect to the results obtained from multiclass UNet.

Further attempts were performed by increasing the learning rate value up to 0.01. The obtained models performed worse since all pixels were classified as unburnt regions.

5.1.3 CUMedVision2

Binary case

Parameter	Value
Batch size	4
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Binary CrossEntropy

Table 5.9: CUMedVision2 training parameters in binary case, RGB input

	Accuracy	Recall	Precision	F1-score
Mean	86.66%	0.563	0.707	0.526
Best	95.92%	0.965	0.963	0.849
Worst	67.63%	0.103	0.203	0.186

Table 5.10: CUMedVision2 evaluation metrics in binary case, RGB input

CUMedVision2 models were trained using the same parameters as CUMedVision1, with the only difference that Binary CrossEntropy Loss was applied also to the secondary branch in charge of contour prediction. As shown in Figure 5.6, in Example 1 the network is predicting the burnt area precisely, whereas in Example 2 the network

is unable to correctly determine the burnt area and thus overestimates it, due to light conditions and high presence of shadows. Furthermore, in all the cases the contour prediction provided a completely empty image, thus underlining the difficulty in determining precisely the shapes of burnt area regions. Due to this reason, CUMed-Vision2 reached comparable performances with respect to CUMedVision1 and thus was not considered in the multiclass scenario.

5.1.4 Comparison with data before and after the activation event

In this subsection, the performances of multiclass UNet model with 6-channels images are reported.

Parameter	Value
Batch size	8
Epochs	50
Optimizer	Adam
Learning rate	0.001
Weight decay	0
Scheduler	MultiStepLR
Steps	Epochs: 25, 35, 45
Gamma	0.1
Loss function(s)	Dice Loss

Table 5.11: UNet training parameters in multiclass case, 6-channels input

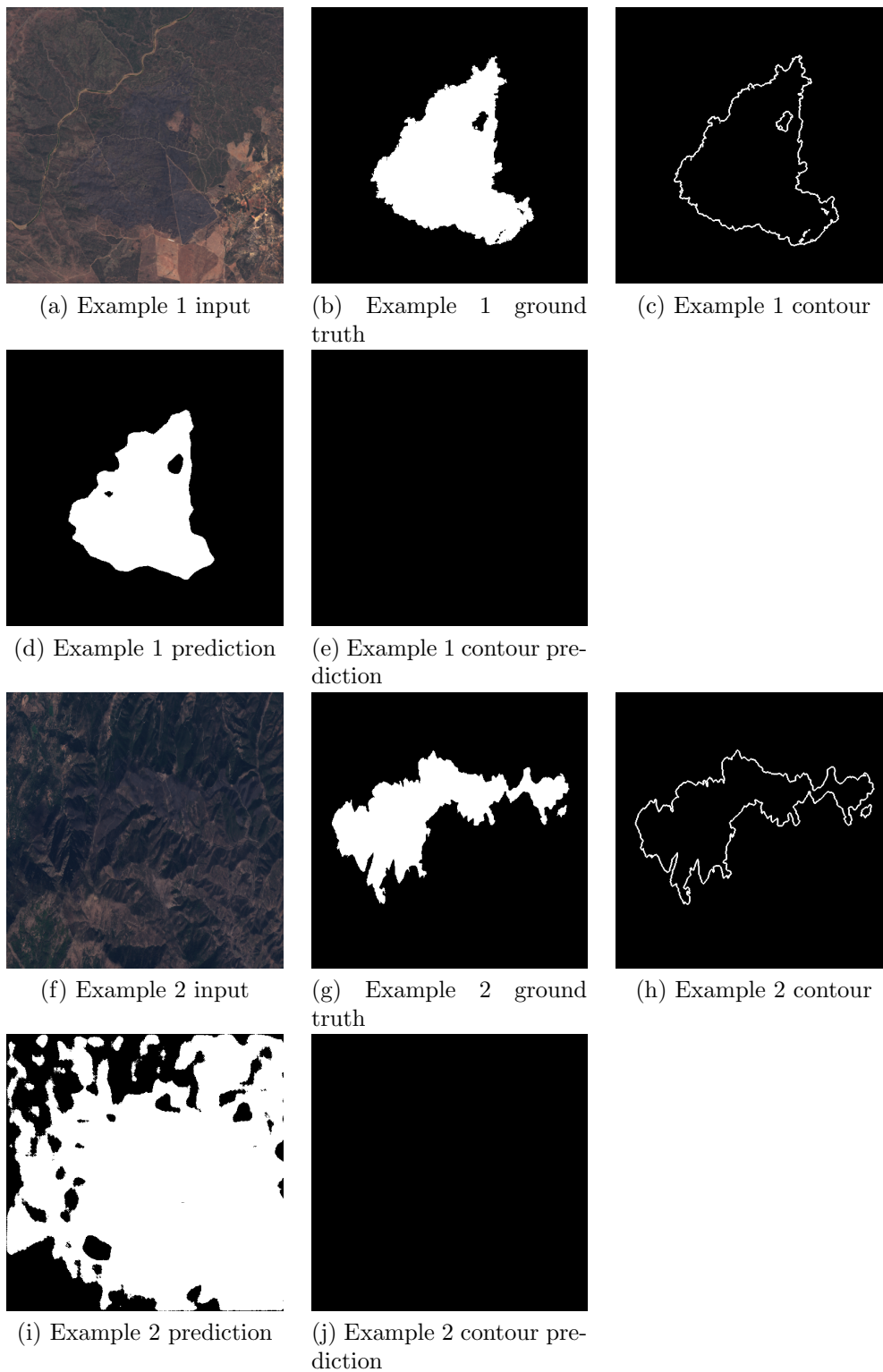


Figure 5.6: Examples from binary CUMedVision2, RGB input

		Recall	Precision	F1-score
Class 0	Mean	0.878	0.954	0.908
	Best	0.991	0.995	0.952
	Worst	0.672	0.892	0.801
Class 1	Mean	0.098	0.078	0.078
	Best	0.234	0.151	0.183
	Worst	0	0.016	0.028
Class 2	Mean	0.378	0.116	0.201
	Best	0.890	0.204	0.332
	Worst	0	0	0.080
Class 3	Mean	0.102	0.248	0.132
	Best	0.392	0.763	0.249
	Worst	0	0.016	0.025
Class 4	Mean	0.364	0.398	0.347
	Best	0.824	0.792	0.560
	Worst	0	0.098	0.148

Table 5.12: UNet evaluation metrics in multiclass case, 6-channels input

5.2 Multi spectral bands input

In this section the same experiments of the previous section are performed, with the only change on the input data. In particular, all the 13 inputs bands available from Sentinel-2 L1C mission are used.

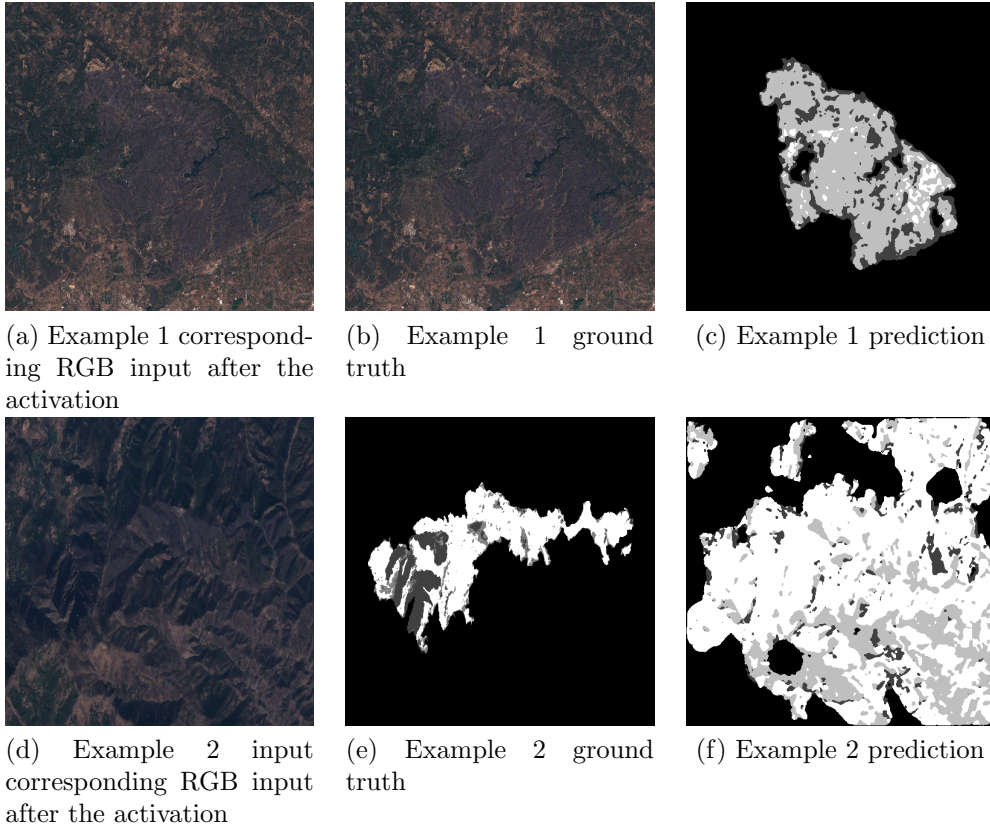


Figure 5.7: Examples from multiclass UNet, 6-channels input

5.2.1 UNet

Binary case

Parameter	Value
Batch size	8
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Binary CrossEntropy

Table 5.13: UNet training parameters in binary case, 13-channels input

	Accuracy	Recall	Precision	F1-score
Mean	95.64%	0.771	0.885	0.811
Best	97.62%	0.964	0.953	0.914
Worst	93.41%	0.444	0.796	0.597

Table 5.14: UNet evaluation metrics in binary case, 13-channels input

By analysing all the 13 input spectral bands available from Sentinel-Hub service, binary UNet model with a threshold of 0.5 achieved an overall mean F1-score over all the folds of **0.811**, compared to 0.665 obtained in the RGB case. Considering the best performance obtained from a fold, F1-score incremented from 0.886 to **0.914**. In particular, by training on 13 features, the model was able to:

- determine burnt areas more precisely;
- identify clearly the presence of water in the image and do not classify it as a burnt area;
- become more robust with respect to atmospheric noise and light conditions, which were limiting factors when considering performances in the RGB scenario;

Two examples are shown in Figure 5.8.

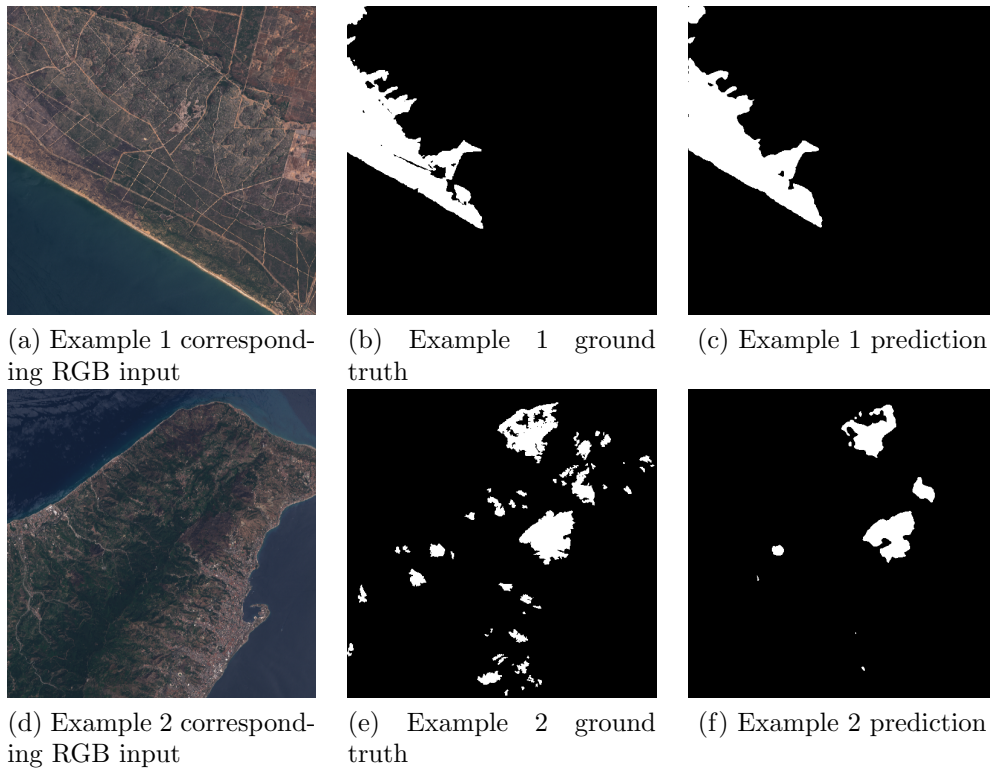


Figure 5.8: Examples from binary UNet with multi spectral bands input

In Example 1, the region was clearly determined by UNet with some minor errors in identifying small unburnt regions surrounded by destroyed areas, whereas in Example 2 a greater number of mistakes were made.

In general, the possibility to use the multi spectral input bands severely improved performances and burnt area delineation in countour regions, but UNet model still lacks the ability to precisely determine small burnt areas.

Multi-class case

Parameter	Value
Batch size	8
Epochs	50
Optimizer	Adam
Learning rate	0.01
Weight decay	0
Scheduler	MultiStep
Steps	Epochs: 25, 35, 45
Gamma	0.1
Loss function(s)	Dice Loss

Table 5.15: UNet training parameters in multiclass case, 13-channels input

		Recall	Precision	F1-score
Class 0	Mean	0.839	0.980	0.902
	Best	0.948	0.991	0.965
	Worst	0.732	0.965	0.842
Class 1	Mean	0	/	/
	Best	0	/	/
	Worst	0	/	/
Class 2	Mean	0.257	0.085	0.119
	Best	0.632	0.158	0.252
	Worst	0.048	0.046	0.050
Class 3	Mean	0.257	0.125	0.199
	Best	0.934	0.343	0.502
	Worst	0.0	0.0	0.0
Class 4	Mean	0.526	0.332	0.411
	Best	0.943	0.830	0.765
	Worst	0.0	0.105	0.177

Table 5.16: UNet evaluation metrics in multiclass case, 13-channels input

As in the RGB input case, Class 1 and Class 3 were never (or rarely) predicted. Only some pixels of Class 2 and Class 4 were correctly classified, with a F1-score of 0.177 and 0.187 as mean values and 0.277, 0.295 as best values respectively. Figure 5.10 shows 3 examples, where:

- Example 1 shows a case in which UNet model is able to determine precisely the shape of the burnt area region, with minor misclassification errors in the contours areas;
- Example 2 shows a case in which the model is severely underestimating the burnt area;
- Example 3 is a case in which the burnt region is overestimated;

In general, the model is not able to correctly detect the severity level of the area: the predominant predicted class is class 4, which in general determines the overall shape of the burnt area, whereas the contour areas are always labelled as class 2. Class 1 is never predicted. Low values of F1-score are justified by the fact that, even though the shape in most of the cases is well determined, the severity level prediction within the identified region is wrong. Compared to RGB input, the trained UNet models were able to partially activate severity level 3 in few cases, which lead to a significant drop of F1-score for class 4 or total absence of it (Figure 5.9).

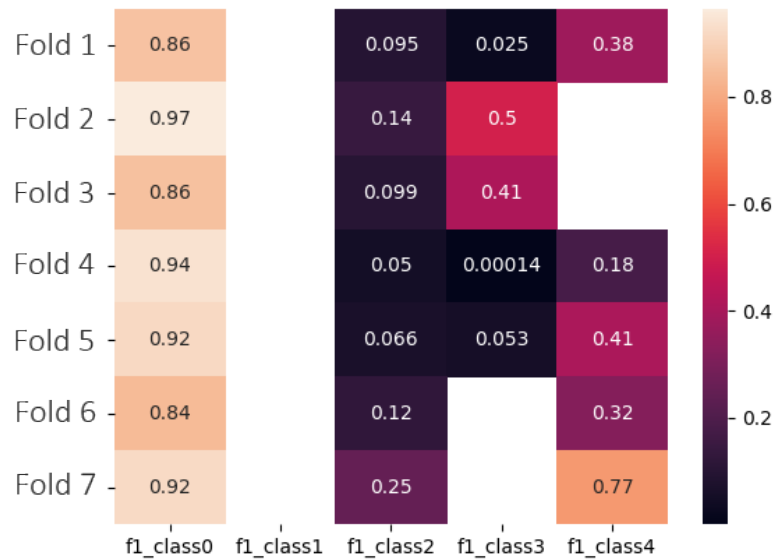


Figure 5.9: F1-score values in multiclass UNet, multi spectral input

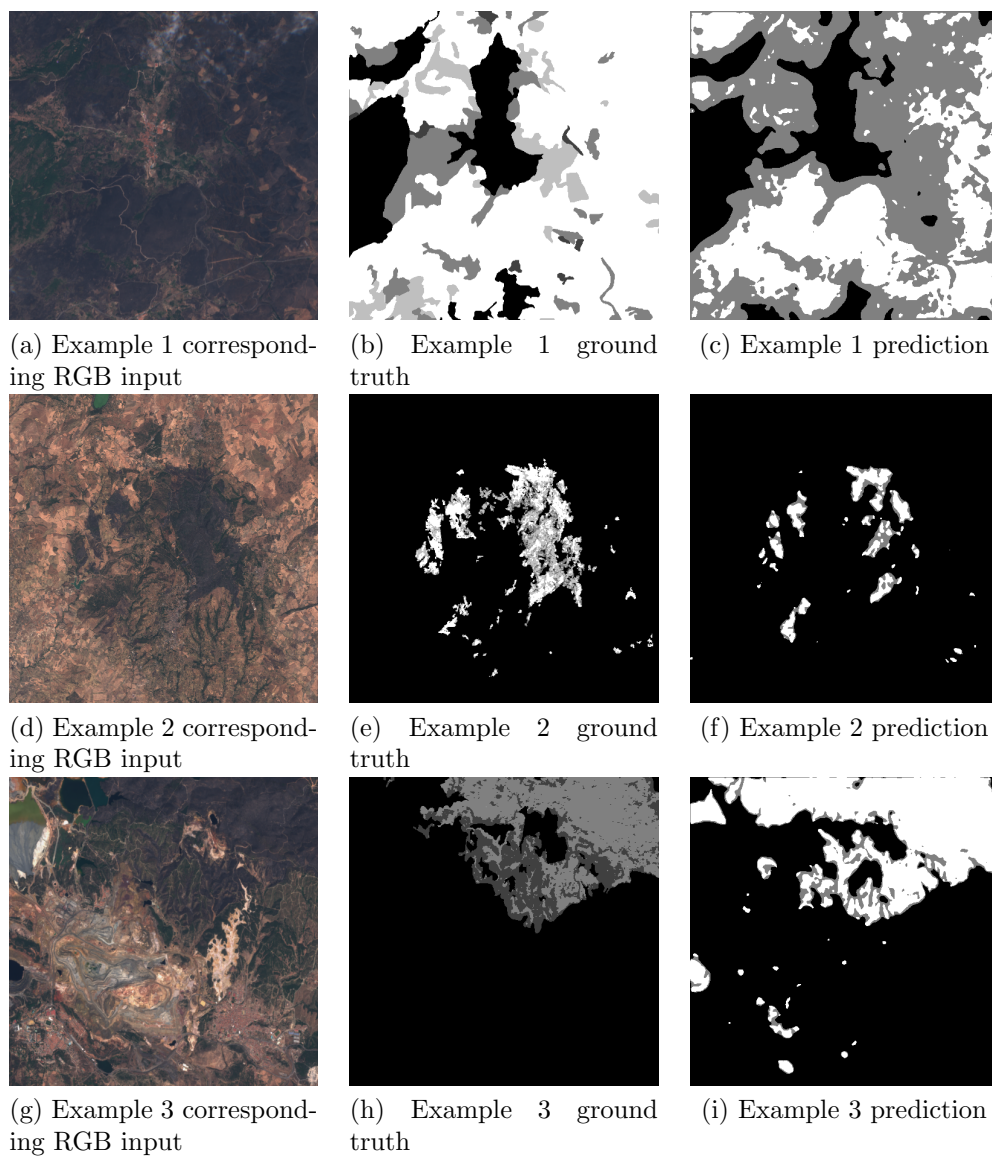


Figure 5.10: Examples from multiclass prediction with UNet, multi spectral bands input

5.2.2 CUMedVision1

Binary case

Parameter	Value
Batch size	16
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Binary CrossEntropy

Table 5.17: CUMedVision1 training parameters in binary case, 13-channels input

	Accuracy	Recall	Precision	F1-score
Mean	94.25%	0.609	0.930	0.719
Best	96.30%	0.805	0.966	0.873
Worst	91.93%	0.295	0.864	0.452

Table 5.18: CUMedVision1 evaluation metrics in binary case, 13-channels input

CUMedVision1 models trained on multi spectral input demonstrated slight improvements in terms of F1-score with respect to the corresponding models trained on the RGB input. Figure 5.11 shows two examples in which the model performs well and underestimates the burnt region respectively. In particular, the neural network learnt to distinguish the presence of water and not classify it as burnt areas. In Example 2, it is likely that the conformation of the land and the low presence of vegetation negatively influenced the prediction outcome.

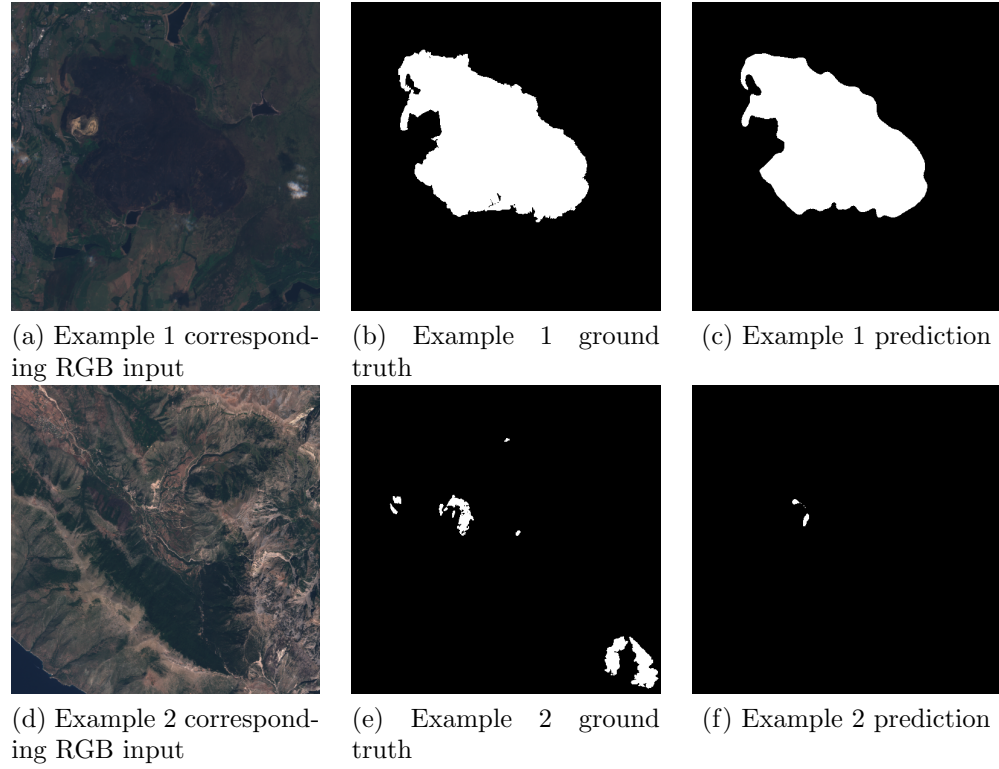


Figure 5.11: Examples from binary prediction with CUMedVision1, multi spectral bands input

Multi-class case

Parameter	Value
Batch size	8
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Cross Entropy

Table 5.19: CUMedVision1 training parameters in multiclass case, 13-channels input

		Recall	Precision	F1-score
Class 0	Mean	0.958	0.973	0.965
	Best	0.989	0.992	0.979
	Worst	0.930	0.942	0.952
Class 1	Mean	0.013	0.083	0.024
	Best	0.029	0.254	0.047
	Worst	0	0	0
Class 2	Mean	0.176	0.138	0.126
	Best	0.378	0.269	0.222
	Worst	0.021	0.052	0.029
Class 3	Mean	0.136	0.231	0.132
	Best	0.390	0.355	0.272
	Worst	0.011	0.074	0.021
Class 4	Mean	0.524	0.253	0.283
	Best	0.851	0.699	0.514
	Worst	0.206	0.041	0.078

Table 5.20: CUMedVision1 evaluation metrics in multiclass case, 13-channels input

Figure 5.12 shows that in general the developed and trained model is able to detect the burnt area, providing as output all the possible 4 severity levels. The main issue is the same encountered in the RGB input case: the network did not learn how to distinguish the different severity levels, thus providing as output a grid of pixels in which different classes are present. Among the 4 severity levels, the best F1-score is achieved in identifying class 4, with a mean value of 0.283 and best value of **0.514**. Class 1 is the worst-performing, with a mean F1-score of 0.024 and best F1-score of 0.047. Considering instead the evaluation metrics of class 0, a mean F1-score of 0.965 is achieved, thus indicating that the neural network is able to correctly distinguish between burnt and unburnt areas, i.e. there are high learning capabilities in the binary setting.

Example 1 shows a case in which few mistakes were made in identifying internal unburnt regions, whereas in Example 2 the burnt region was underestimated.

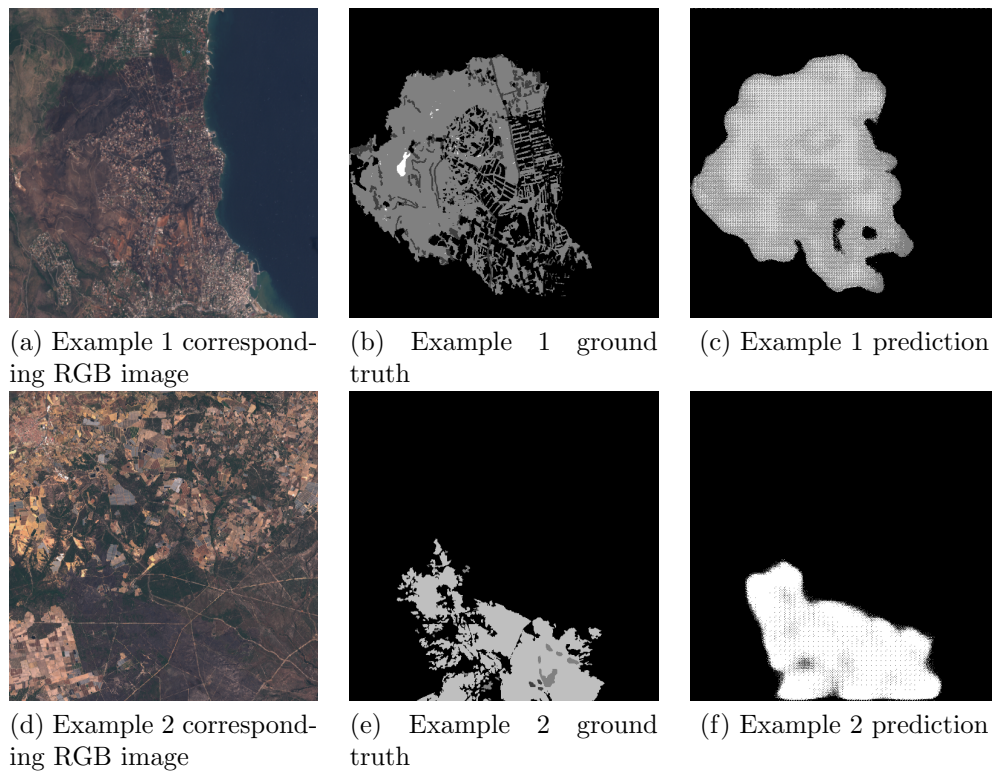


Figure 5.12: Examples from multiclass prediction with CUMedVision1, multi spectral bands input

5.2.3 CUMedVision2

Binary case

Parameter	Value
Batch size	4
Epochs	25
Optimizer	Adam
Learning rate	0.0001
Weight decay	0
Scheduler	/
Loss function(s)	Binary CrossEntropy

Table 5.21: CUMedVision2 training parameters in binary case, 13-channels input

	Accuracy	Recall	Precision	F1-score
Mean	92.91%	0.702	0.852	0.743
Best	97.67%	0.8981	0.996	0.852
Worst	87.97%	0.432	0.560	0.603

Table 5.22: CUMedVision2 evaluation metrics in binary case, 13-channels input

CUMedVision2 models were trained in the multibands case to determine whether or not the considered architecture is able to determine contour areas in case of multi spectral band input. Figure 5.13 shows two examples. In all the cases, the models were not able to detect contours of the burnt area region. As in the RGB input case, performances are comparable with CUMedVision1 and thus this architecture has not been considered in the multiclass case.

5.2.4 Comparison with data before and after the activation event

In this subsection, the performances of multiclass UNet model with 26-channels images are reported.

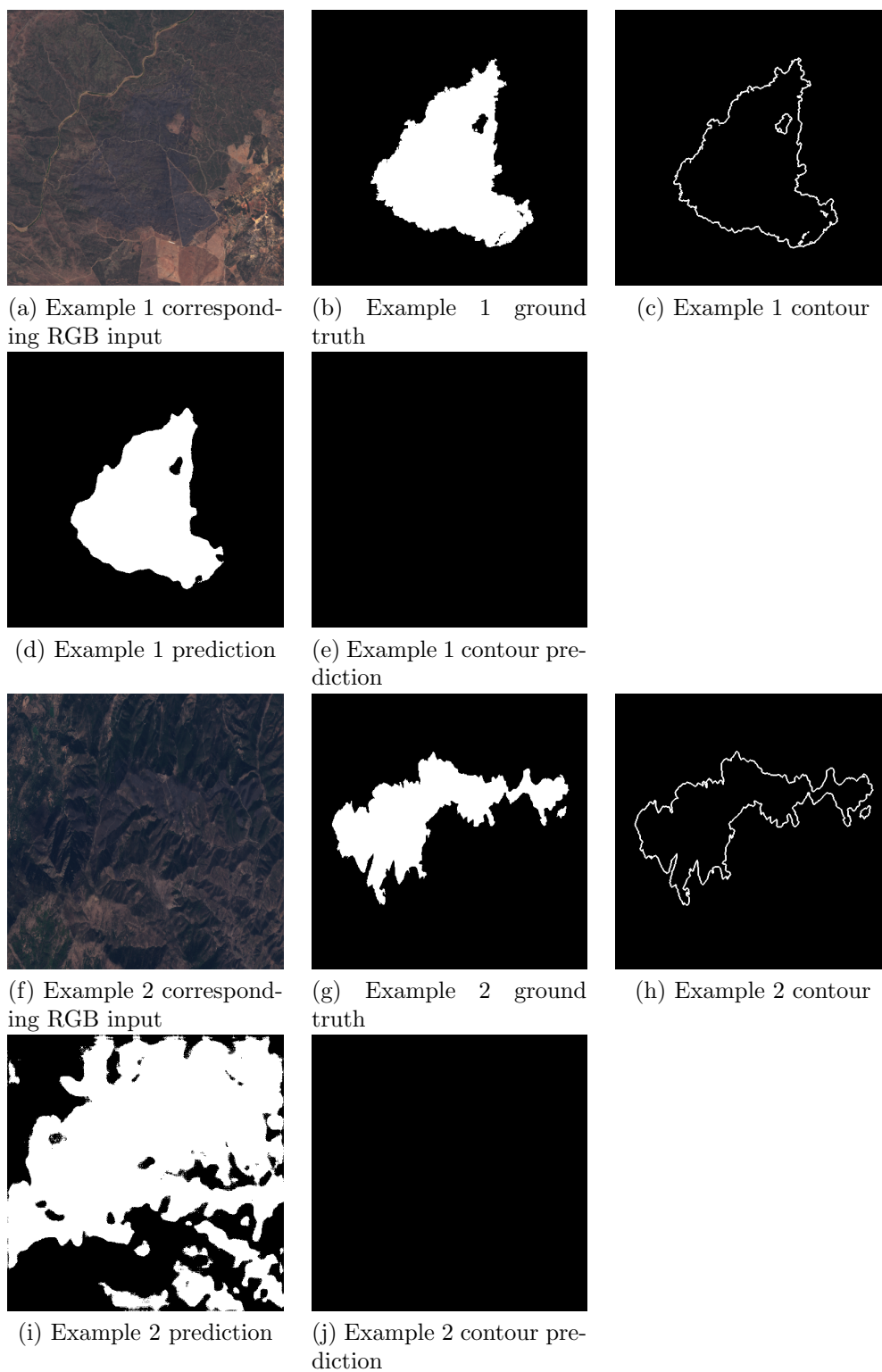


Figure 5.13: Examples from binary CUMedVision2, multi spectral input

Parameter	Value
Batch size	16
Epochs	50
Optimizer	Adam
Learning rate	0.01
Weight decay	0
Scheduler	MultiStepLR
Steps	Epochs: 25, 35, 45
Gamma	0.1
Loss function(s)	Dice Loss

Table 5.23: UNet training parameters in multiclass case, 26-channels input

		Recall	Precision	F1-score
Class 0	Mean	0.975	0.963	0.969
	Best	0.998	0.990	0.994
	Worst	0.895	0.913	0.932
Class 1	Mean	0	/	/
	Best	0	/	/
	Worst	0	/	/
Class 2	Mean	0.374	0.184	0.206
	Best	0.685	0.449	0.354
	Worst	0.004	0.082	0.008
Class 3	Mean	0.047	0.178	0.063
	Best	0.128	0.660	0.114
	Worst	0	0.007	0.010
Class 4	Mean	0.391	0.393	0.356
	Best	0.755	0.861	0.542
	Worst	0.035	0.038	0.037

Table 5.24: UNet evaluation metrics in multiclass case, 26-channels input

Figure 5.14 shows 3 examples of the developed model. Example 1 shows a case in which UNet was able to precisely determine the burnt areas, with major errors made in determining the class. Example 2 is a case with extremely difficult light conditions, in which previously trained models severely overestimated the burnt area. Thanks to the differential approach, in which two satellite images acquired before and after the event are compared, the neural network was able to correctly identify the burnt

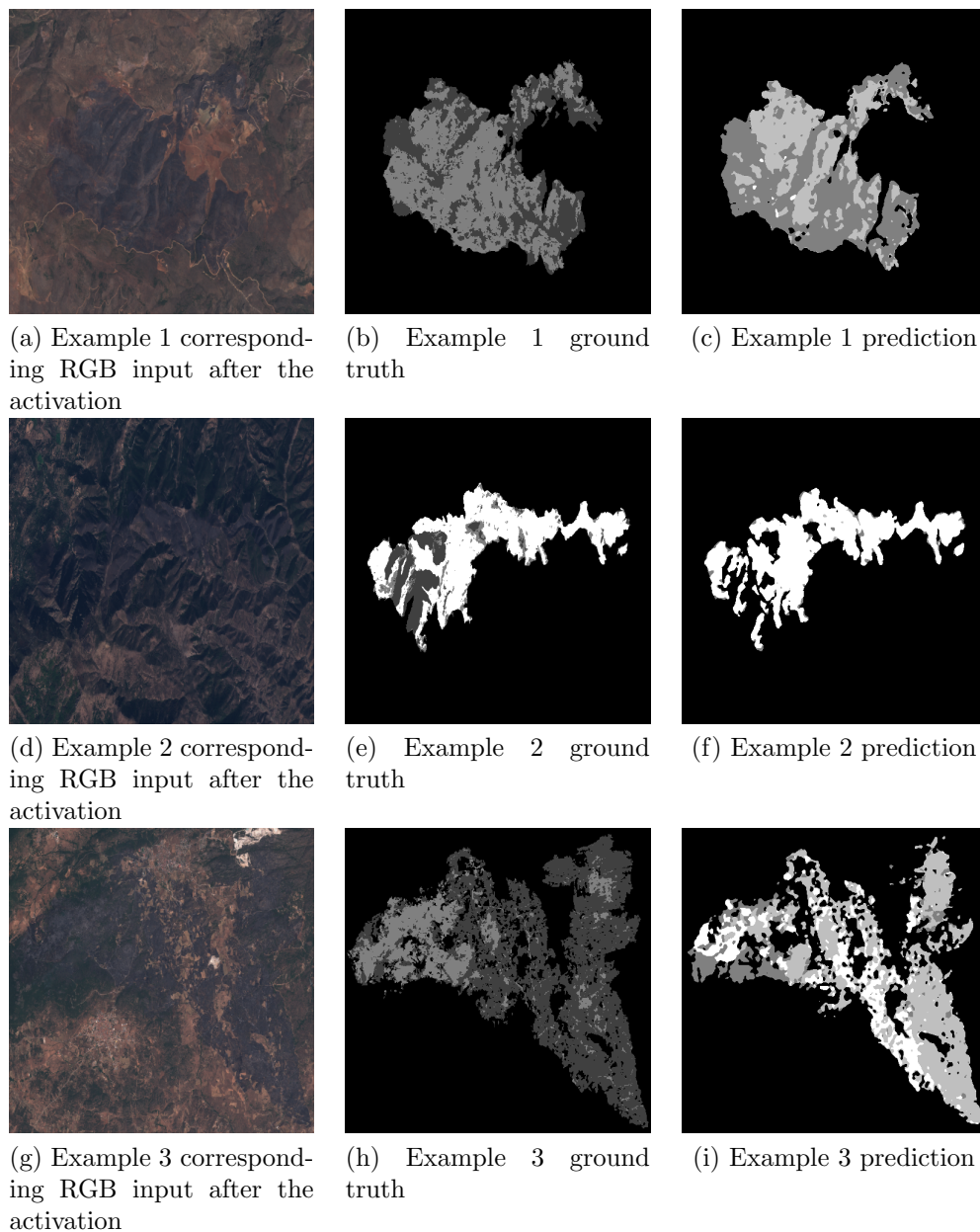


Figure 5.14: Examples from multiclass UNet, 26-channels input

area. Example 3 shows a case in which class 1 was entirely misclassified as class 2. Furthermore, providing images before and after the activation event may lead to misclassification errors when considering a longer time span, especially in case the vegetation changes or when considering cultivated areas.

5.3 Models comparison

In this section, all the different models are compared in terms of accuracy, recall, precision and F1-score, considering both the RGB input and the multi spectral input. The metrics reported in the tables are mean values, computed considering the results from all the 7 folds. The networks are compared in the binary case, as well as in the multiclass case.

Table 5.25 shows the performance comparisons between the two considered models. CUMedVision2 model was discarded due to poor performances and the inability to detect contour areas. Overall, UNet model performed better than CUMedVision1 model. In particular, in the RGB case the differences in terms of F1-score are fairly limited. As the number of input features is incremented to all the bandwidths available from Sentinel-2 L1C products, UNet demonstrated greater learning capabilities by better identifying burnt areas with respect to CUMedVision1. In particular, as shown previously, major misclassification errors are made in contour regions for both the models, with UNet performing better and CUMedVision1 greatly underestimating the burnt region.

		UNet	CUMedVision1
RGB	Accuracy	90.43%	90.40%
	Recall	0.698	0.688
	Precision	0.706	0.678
	F1-score	0.665	0.647
Multi Spectral input	Accuracy	95.64%	94.25%
	Recall	0.771	0.609
	Precision	0.885	0.930
	F1-score	0.811	0.719

Table 5.25: Performance comparison in Binary case

Segmentation maps computed by thresholding the NBR2 index were used as an evaluation metrics to assess the deep learning models. The threshold values

were computed using the Otsu method on each single image, similarly to [18]. The considered segmentation technique achieved the following scores:

- accuracy: 72.97%;
- precision: 0.340;
- recall: 0.884;
- F1-score: 0.491;

The worst-performing model achieved higher performances in all the 4 evaluation metrics, with further improvements in the multi spectral input case.

A comparison of different outputs is shown in Figure 5.15, where the input image was obtained from Fold 7.

In the examples shown, it is noticeable the difference between RGB models and multibands models, especially in case of CUMedVision1: since more significant features are available, the neural networks are able to better predict burnt areas in the considered region. However, the prediction outcomes severely depends on the time at which the satellite imagery were taken: the more the time passes after the activation date, the less significant the available data become. Unfortunately, this issue is out of the control of the developer/data scientist since the frequency at which the satellite imagery is taken strictly depends on Sentinel-2 orbits and availability of the service.

Models performances by considering the extension to the multiclass case are shown in Table 5.26. By considering the evaluation metrics, CUMedVision1 was able to recall all the 5 classes, while UNet never activated class 1 and class 3 in the RGB case and class 1 in the multibands input case. Furthermore, F1-score of CUMedVision1 in class 0 is higher in both the cases. This is due to the fact that CUMedVision1 underestimates the amount of burnt pixels, thus leading to better results in class 0. Considering class 2 and class 4, UNet reached comparable or even better results than CUMedVision1, especially in the case of multi spectral input bands.

Among the different burnt severity levels, the best F1-score of **0.411** is achieved by UNet model with multi spectral bands input on class 4, compared to CUMedVision1 which achieved **0.283**. This result is justified by the fact that the majority of activations monitored by Copernicus EMS present a greater percentage of burnt areas of class 4.

Figure 5.16 shows an example of multiclass prediction. In general, both models do not correctly predict class 1, 2 and 3 and identify the entire burnt area as class 4,

with some minor regions marked as Moderately Damaged (class 2) near the contours. UNet outputs are sharper than CUMedVision1, which tends to identify the burnt area with a grid pattern, where different severity levels are repeated, thus justifying extremely low values of F1-score for class 1 and class 3. Finally, two UNet models were trained by taking as input two images: an image prior the activation event and one image after the wildfire was stopped. One model was trained on RGB images, whereas the second was trained on multi spectral band inputs. The main differences are the following:

- RGB input: the 6-channels UNet achieved a higher F1-score with respect to RGB UNet. The developed UNet was also able to classify pixels belonging to class 1 and class 3;
- multi spectral input: the 26-channels UNet achieved higher values of mean precision compared to the traditional UNet, with a better mean F1-score on class 2 and class 0. Furthermore, this models demonstrated more robustness in mountainous areas;

These experiments underline the limitation of the implemented models: both UNet and CUMedVision1 were originally implemented in the context of biomedical image segmentation. Their adaptations in the analysis of satellite imagery evidence their limitations, due to the absence of cell membranes and stacked areas with different severity levels: the difficulty in the identification of small areas and the ability to precisely determine contour regions.

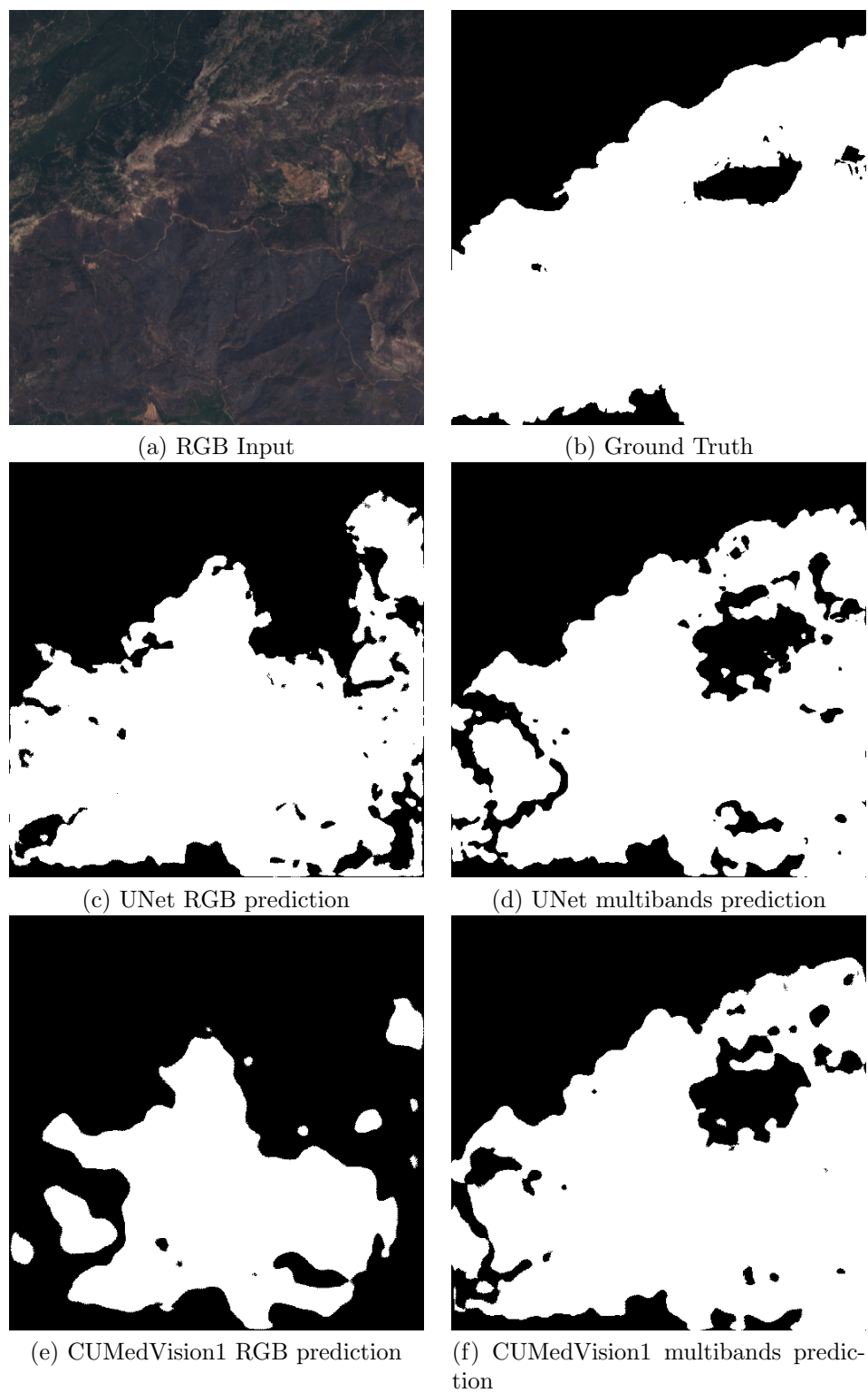


Figure 5.15: Model predictions comparison in binary case

			UNet	UNet 6/26-channels	CUMedVision1
RGB	Class 0	Recall	0.842	0.878	0.938
		Precision	0.954	0.954	0.910
		F1-score	0.889	0.908	0.921
	Class 1	Recall	0	0.098	0.026
		Precision	/	0.078	0.033
		F1-score	/	0.078	0.016
	Class 2	Recall	0.350	0.378	0.039
		Precision	0.136	0.116	0.078
		F1-score	0.177	0.201	0.041
	Class 3	Recall	0	0.102	0.112
		Precision	/	0.248	0.145
		F1-score	/	0.132	0.075
	Class 4	Recall	0.496	0.364	0.382
		Precision	0.207	0.398	0.257
		F1-score	0.187	0.347	0.245
Multibands	Class 0	Recall	0.839	0.975	0.958
		Precision	0.980	0.963	0.973
		F1-score	0.902	0.969	0.965
	Class 1	Recall	0	0	0.013
		Precision	/	/	0.083
		F1-score	/	/	0.024
	Class 2	Recall	0.257	0.374	0.176
		Precision	0.085	0.184	0.138
		F1-score	0.119	0.206	0.126
	Class 3	Recall	0.257	0.047	0.136
		Precision	0.125	0.178	0.231
		F1-score	0.199	0.063	0.132
	Class 4	Recall	0.526	0.391	0.524
		Precision	0.332	0.393	0.253
		F1-score	0.411	0.356	0.283

Table 5.26: Performance comparison in multiclass case

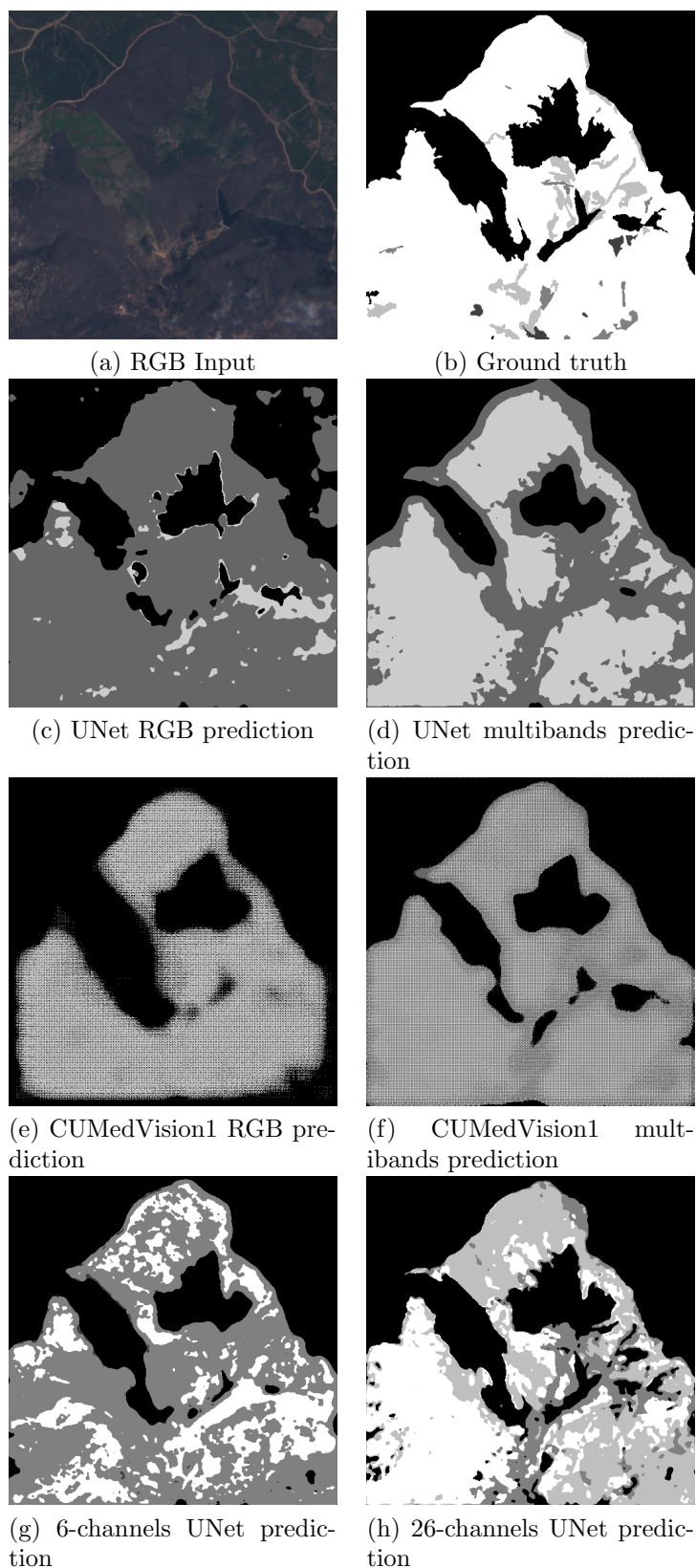


Figure 5.16: Model predictions comparison in multiclass case

Chapter 6

Conclusions

In this chapter possible future developments in the field of automatic satellite image processing through deep learning techniques are presented. Then, some final considerations are given.

6.1 Future developments

As seen from the previous chapter and due to the nature of the phenomenon, models performances, especially in a multi-class context, offer margin of improvements. This is due to two main reasons:

- nature of data: natural disasters and forest wildfires are unpredictable events. They depend on a great amount of factors, from human activities to meteorological information and the morphological aspect of the involved areas. Also, the time component is extremely important: as time passes, the effect of a forest wildfire may reduce due to growing vegetation, thus making it extremely difficult to provide a prediction on the severity level;
- nature of the model: both UNet and CUMedVision architectures were created in the context of biomedical image segmentation, where cells and organs are defined by specific membranes or tissues, whereas burnt areas with different severity levels are of different shapes and land extensions;

To overcome performances issues, it is possible to work either on input data or on the network architecture.

6.1.1 Feature engineering

First of all, it is important to remember that all the input data used in this thesis are obtained from Sentinel2-L1C mission, which provides raw data obtained from satellite acquisitions. Sentinel2-L1C products are affected by atmospheric noise which can negatively influence the training process and the outcome. To overcome this issue, it could be possible to use Sentinel2-L2A products if available, which provide instead bottom of atmosphere reflectance and thus limit the noise introduced by the atmosphere.

Another possibility is to train a neural network to remove noise from Sentinel2-L1C products: if both L1C and L2A data are available, it is possible to train a custom neural network which learns how to clean the input data. Then, the cleaned data is used to feed and train a UNet model both in binary and multiclass case. This automated cleaning process could provide benefits, especially in case the acquisition is performed by an aircraft and not by a satellite. Eventually, the input feature space could be enriched with data from Sentinel-1 mission.

To enhance model performances, it is also possible to preprocess and eventually transform the input data using image and signal processing techniques. In particular, considering the RGB case, it is possible to analyse different color theory techniques, different color spaces and equalization techniques to manipulate input data. As an example, Figure 6.1 shows a possible image altered using the HSV color space by increasing saturation and value.

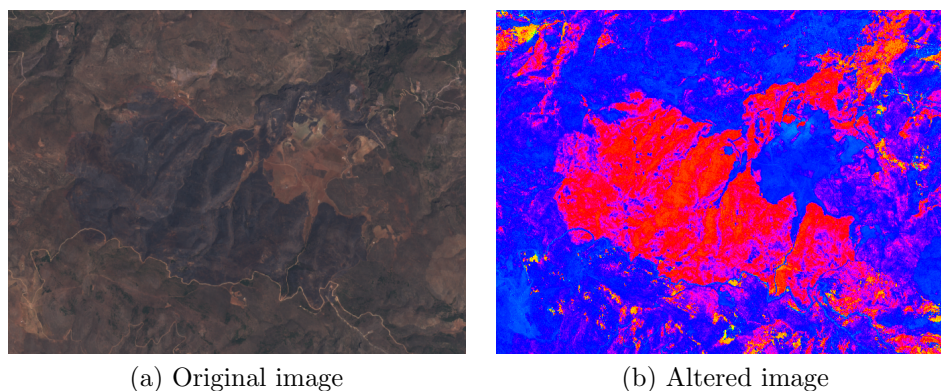


Figure 6.1: Example of image alteration using HSV color space

This exact same concept can be extended to multi spectral bands input but requires proper domain knowledge for data manipulation and feature engineering, where a new feature map is identified such that different severity levels are more

separable with respect to the original feature space.

6.1.2 Architectural and loss improvements

To reduce misclassification errors due to light conditions and to enhance the identification of small burnt areas, it is also possible to modify the current neural network architecture, either by changing the encoder-decoder structure or by revising the entire network and develop ad-hoc solutions for satellite imagery context. Indeed, the absence of a great amount of data covering different types of geographic areas in different periods of the year is a great issue that cannot be easily overcome and underlines the need of models which are able to learn with a limited amount of samples per each type of morphological areas. In particular, many misclassification errors were made in mountainous area and cultivated areas due to high presence of shadows and great changes during summer and winter season respectively.

The development of a custom loss function may also provide benefits to the training process, both in terms of time required and model performances.

6.2 Final considerations

Analysis of satellite imagery with machine learning and especially deep learning techniques is still at early stages of development and many improvements are possible. The development of machine learning algorithms for emergency management systems is extremely important, especially nowadays, due to the high number of forest wildfires which are endangering the planet environment. Such a system could be for example installed on aircrafts to automatically identify burnt regions and automatize the mask generation process, which is entirely done manually by operators at the time of writing. This thesis showed the feasibility of automatic burnt area detection and the limits of the considered architectures in this context.

Bibliography

- [1] M. D. Flannigan, B. D. Amiro, K. A. Logan, B. J. Stocks, and B. M. Wotton, “Forest fires and climate change in the 21 st century,” *Mitigation and adaptation strategies for global change*, vol. 11, no. 4, pp. 847–859, 2006.
- [2] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, *et al.*, “Attention u-net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*, 2018.
- [3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [4] W. Shen, B. Wang, Y. Jiang, Y. Wang, and A. Yuille, “Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2391–2400, 2017.
- [5] M. Chung, J. Lee, M. Lee, J. Lee, and Y.-G. Shin, “Deeply self-supervising edge-to-contour neural network applied to liver segmentation,” *arXiv preprint arXiv:1808.00739*, 2018.
- [6] R. Ramo, M. García, D. Rodríguez, and E. Chuvieco, “A data mining approach for global burned area mapping,” *International journal of applied earth observation and geoinformation*, vol. 73, pp. 39–51, 2018.
- [7] R. Ramo and E. Chuvieco, “Developing a random forest algorithm for modis global burned area classification,” *Remote Sensing*, vol. 9, no. 11, p. 1193, 2017.
- [8] T. Kumar and K. Reddy, “A technique for burning area identification using ihs transformation and image segmentation,” *International Arab Journal of Information Technology (IAJIT)*, vol. 12, no. 6A, pp. 764–771, 2015.

- [9] L. A. Hardtke, P. D. Blanco, H. F. del Valle, G. I. Metternicht, and W. F. Sione, "Semi-automated mapping of burned areas in semi-arid ecosystems using modis time-series imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 38, pp. 25–35, 2015.
- [10] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bon-tempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, 05 2019.
- [11] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*, pp. 216–223, Springer, 2012.
- [12] K. Gurney, *An introduction to neural networks*. CRC press, 2014.
- [13] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.
- [14] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "Dcan: Deep contour-aware networks for object instance segmentation from histology images," *Medical image analysis*, vol. 36, pp. 135–146, 2017.
- [15] H. S. University, "Vegetation spectral reflectance curves." http://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson2-1/vegetation.html.
- [16] H. S. University, "Normalized burn ratio." http://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson5-1/NBR.html.
- [17] U. F. Service, "Geospatial technology and applications center (gtac)." <https://www.fs.fed.us/eng/rsac/baer/barc.html>.
- [18] W. Bin, L. Ming, J. Dan, L. Suju, C. Qiang, W. Chao, Z. Yang, Y. Huan, and Z. Jun, "A method of automatically extracting forest fire burned areas using gf-1 remote sensing images," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 9953–9955, IEEE, 2019.
- [19] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [20] F. Filipponi, "Bais2: Burned area index for sentinel-2," in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 2, p. 364, 2018.

- [21] E. Roteta, A. Bastarrika, M. Padilla, T. Storm, and E. Chuvieco, “Development of a sentinel-2 burned area algorithm: Generation of a small fire database for sub-saharan africa,” *Remote sensing of environment*, vol. 222, pp. 1–17, 2019.
- [22] G. Donchyts, J. Schellekens, H. Winsemius, E. Eisemann, and N. van de Giesen, “A 30 m resolution surface water mask including estimation of positional and thematic differences using landsat 8, srtm and openstreetmap: a case study in the murray-darling basin, australia,” *Remote Sensing*, vol. 8, no. 5, p. 386, 2016.
- [23] “Copernicus emergency management service.” <https://emergency.copernicus.eu/>.
- [24] “Gdal translator library.” <https://gdal.org/>.
- [25] “Sentinel hub.” <https://www.sentinel-hub.com/>.
- [26] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. ” O’Reilly Media, Inc.”, 2008.
- [27] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in python,” *PeerJ*, vol. 2, p. e453, 2014.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.
- [30] C. Wang, T. MacGillivray, G. Macnaught, G. Yang, and D. Newby, “A two-stage 3d unet framework for multi-class segmentation on full resolution image,” *arXiv preprint arXiv:1804.04341*, 2018.
- [31] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, “Attention gated networks: Learning to leverage salient regions in medical images,” *Medical image analysis*, vol. 53, pp. 197–207, 2019.

- [32] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in *International conference on medical image computing and computer-assisted intervention*, pp. 424–432, Springer, 2016.
- [33] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [34] L. Taylor and G. Nitschke, “Improving deep learning using generic data augmentation,” *arXiv preprint arXiv:1708.06020*, 2017.