POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Vision Transformers for burned area detection

Supervisors

Prof. Paolo GARZA

Dott. Luca COLOMBA

Candidate

Daniele REGE CAMBRIN

October 2022

Abstract

The automatic identification of burned areas is an important task that was mainly managed manually or semi-automatically in the past. In the last years, thanks to the availability of novel deep neural network architectures, automatic semantic segmentation solutions have been proposed also in the emergency management domain. The most recent works in burned area delineation make use of Convolutional Neural Networks (CNNs) to automatically identify regions that were previously affected by forest wildfires. A largely adopted segmentation model, U-Net, demonstrated good performances for the task under analysis, but in some cases a high overestimation of burned areas is given, leading to low precision scores. Given the recent advances in the field of NLP and the first successes also in the vision domain, in this thesis, we investigate the adoption of vision transformers for semantic segmentation to address the burned area identification task. In particular, we explore the SegFormer architecture with two of its variants: the smallest SegFormer-B0 and the intermediate one, SegFormer-B3. We exploited different loss functions to deal with the complex structures that can be found in satellite imagery. The experimental results show that SegFormer provides better predictions, with higher precision and F1 score, but also better performance in terms of the number of parameters with respect to CNNs.

Acknowledgements

I would thank professor Paolo Garza and Luca Colomba for all the technical help provided in creating this thesis, without them some parts of this work would never be created. I thank my family and friends for the support provided during these years, without them, I would not probably be here doing this write. I am especially glad to my parents Claudia and Riccardo for supporting me during my university experience in the good and bad parts. I have to thank HPC@POLITO and SMARTDATA@POLITO for providing computational resources to work on this project.

Table of Contents

List of Tables

List of Figures

Acronyms

CNN
: Convolutional Neural Network

ViT
: Vision Transformer

BCE
: Binary Cross-Entropy

Chapter 1

Introduction

In this chapter we are giving a context to our problem trying to describe why it is important, how the problem is currently addressed and how we tried to solve it.

1.1 Wildfires and their impact

Climate change is increasing the frequency of severe fire weather (conditions that favour the ignition of wildfires) and fire seasons around the globe are becoming longer and are spreading in larger areas [1]. According to the predictions of domain experts, the actual trend will intensify these environmental states [2]. Extreme weather conditions (strong wind, abnormal heat etc.) that increase aridity [3] or leave for example high fine fuel loadings [4] are not the only cause of wildfires, but human has a great role directly (smoking, playing with fires, using vehicles and equipment [5]) and also indirectly (through infrastructure failures such as electrical transmission lines and railways [6, 7]). We can easily understand in the near future we will have to deal with more wildfires.

The effects of wildfire can be seen in the short term, but also in the long term and they impact health, economies, livelihoods, infrastructure, and societies. Wildfires cause a release of high quantities of carbon [1] and reach long distances affecting the air quality for a long time [8]. Damages to the economy are of great importance: only in December 2020 in California 9600 wildfires were registered and they damaged around 10500 structures [9] and for the period 2019-2021 their damages are estimated at 25 billion dollars [10]. The human communities are not the only ones damaged by wildfires: ecological systems also feel the effects on many of their components such as birds [11], amphibians [12] and mammals [13], but for many of them we do not have any clear information on their reactions.

1.2 Burned area detection with satellite imagery

The availability of sensors with high resolution, in conjunction with the usage of aircraft and satellites, enables the acquisition of national and global-scale information in a short amount of time. Moreover, thanks to the recent advances in computer vision and the high availability of data in the remote sensing domain, Earth Observation represents an active field of research with strong community involvement. The Earth Observation domain involves several different tasks, ranging from land monitoring and land cover change characterization [14], change detection [15], damage estimation [16] and many others. Deep learning-based methodologies demonstrated state-of-the-art performances over a multitude of these tasks. Among the Earth Observation domain, the field of emergency management plays an important role for public authorities as well as governments in handling natural hazards, trying to limit societal and environmental damages as much as possible with timely intervention and proper restoration. Handling natural hazards also involves precise identification of affected areas, damage estimation and restoration process planning. Such mentioned operations are often performed reaching the area of interest, requiring human operators to spend time and risk their health in an unsafe environment. The availability of remote sensing data and satellite imagery enables the development of automatic recognition systems to delimit affected areas and provide initial damage assessments for operators and authorities. In this context, we concentrate our analyses on forest fires. More specifically, we propose our work in the field of semantic segmentation and automatic burned area identification from Copernicus Sentinel-2 L2A acquisitions, a European multi-spectral imaging mission with a resolution up to 10m (depending on the spectral band). Given a post-fire multispectral acquisition from Sentinel-2, the goal is to precisely identify the region affected by the already-extinguished forest fire. This thesis explores the application of one of the most recent advances in deep learning and computer vision: transformer-based architectures for semantic segmentation. In particular, we assess the performances of the SegFormer [17] architecture on an open dataset in comparison with a CNN-based state-of-the-art architecture, namely U-Net [18]. The considered model proved superior performance compared to both methods. The source code is available at `https://github.com/DarthReca/fire-detection`.

1.3 Thesis outline

The thesis is organized into chapters, each dealing with a specific topic:

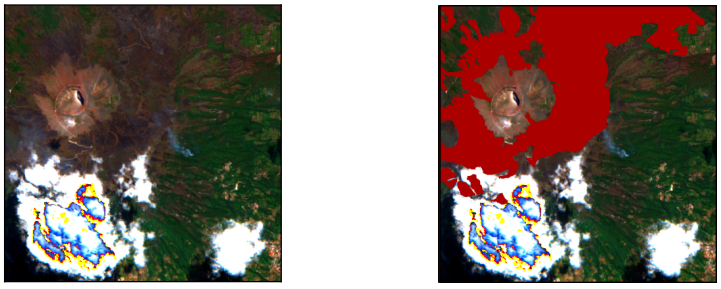
- Chapter 2 summarizes previous works in the computer vision and burned area detection fields;

- Chapter 3 formalizes the problem and presents the various methodologies applied to tackle the problem;
- Chapter 4 presents the results obtained using the previously explained methodologies and it tries to explain them;
- Chapter 5 summarizes the works done in the thesis and deals with possible future works.

Chapter 2

Related works

The burned area identification problem, also named as burned area delineation problem, is a well-known and tackled challenge in remote sensing literature. The aforementioned issue consists in identifying, given a multispectral input acquisition, the areas previously affected by forest wildfire and currently damaged as in Figure 2.1. Such information is useful to (i) quantify damages, both environmental and economical, for public authorities and (ii) plan the restoration process. The field of this work is computer vision, in particular, we are doing semantic segmentation of images. The data are provided by Sentinel-2 satellites and they permit a good understanding of the vegetation status, which is of great importance to work on identifying areas burned by wildfires. Our solution uses the latest development in the deep learning field: Vision Transformers.



Satellite image with burned area Mask of burned area (red)

Figure 2.1: Example of an image from Sentinel-2 and highlight of burned area

2.1 Computer vision

Computer vision is a subfield of artificial intelligence where computers derive useful information from visual inputs (images, videos, etc) and they make decisions based on them. The basic idea is to help computers to see and interpret the world around them through some sensing devices. In this context, some of the most common tasks are (Figure 2.2):

(a) Classification of images or 3D objects. The task consists in assigning to each image (2D or 3D) a class label, based on its content. [19]

(b) Object detection. The main focus is to detect and localize objects using bounding boxes, which are contour boxes containing the searched object. [20]

(c) Pose estimation. The objective is to estimate the position and the orientation of the human body. It founds applications in augmented and virtual reality, gaming and sports. [21]

(d) Image and video generation. This task involves the creation of new media from an existing pool of data that can seem real. The main application is in art and animation. [22]

(e) De-noising. This task consists in removing noise from a media and predicting the original one without noise. This task is useful for the restoration process and other computer vision tasks that require high-quality media. [23]

(f) Activity recognition. This task regards the prediction of the movement in a video. [24]

(g) Semantic segmentation. The objective is to assign a class label to each pixel: it can be thought of as image classification at the pixel level. [17]

(h) Instance segmentation. The output is the contour (or mask) of a searched object: this is similar to object detection, but in this case, we are not looking at bounding boxes. Differently from semantic segmentation, we are not labelling each pixel, because we are searching for specific objects. [25]

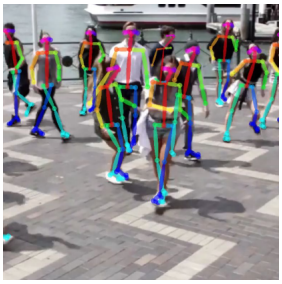
(i) Panoptic segmentation. We combine instance and semantic segmentation to label every pixel of the image. The main difference is that if there are multiple instances of a class, we know which pixel belongs to which instance. [26]

In this thesis, the main topic is semantic segmentation. An input image is given to the network, which generates an output with each pixel assigned to a class. This is compared to the expectation to update the network weights.

(a) classification [19]

(b) object detection [20]

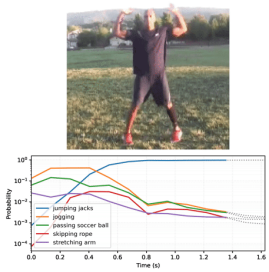
(c) pose estimation [21]



(d) image generation [22]

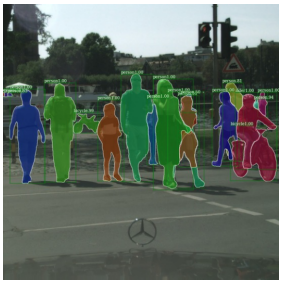
(e) de-noising [23]

(f) activity recognition [24]



(g) semantic segmentation [17]

(h) instance segmentation [25]



(i) panoptic segmentation [26]

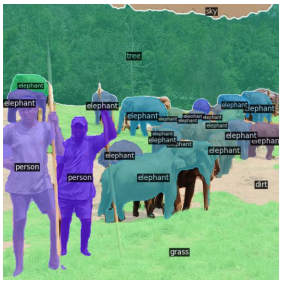


Figure 2.2: Different computer vision tasks

2.2 Deep learning in computer vision

Computer vision only exploded recently, but it is not a young scientific field. During 1960s some influential papers were published giving the ideas for deep learning birth [27] and there was a first unsuccessful attempt of creating a vision system [28]. Studies of 1970s formed the early foundations for many of the computer vision algorithms that exist today [29] and the first robust Optical Character Recognition (OCR) system was developed [30]. In 1980s it was established vision works hierarchically [31] and Neocognitron was developed, which is a network of cells with convolution operations that can recognize patterns [32]. At the end of the decade, the first modern convolutional neural network was developed and it was

called LeNet [33]. In 1990s the improvement in the progress of the cameras helps to feed the field. After 2000, the focus of computer vision was object recognition and the first recognition system was created without the help of convolutional layers [34]. The advancements in computer vision led to the necessity of standard datasets with many images as benchmarks and in 2010 ImageNet Large Scale Visual Recognition Competition (ILSVRC) was created [19]. Until 2012 the error rates in the image classification competition were around 26%, but AlexNet, a convolutional neural network inspired by LeNet, changed everything reaching 16%. From that moment the winners of the ILSVCR were convolutional neural networks. The nourishment was favoured by the higher computational availability and better access to large datasets of images for many different tasks.

2.2.1 Loss functions for semantic segmentation

Loss functions represent the cost (or risk) associated with a prediction. This means the objective is intuitively to minimize it to have a low prediction cost (i.e., accurate predictions). They generally take the form of expected risk:

$$L = E(G, S) \tag{2.1}$$

where L is the loss, G is the ground truth (the expected result), S is the prediction and E is the function that computes the expected risk. They are also called *objective functions* because they lead the system towards a goal. In Figure 2.3 it is possible to see some common functions for image segmentation, but many others can be found. They can be grouped in clusters based on their optimization goal: distribution-based, region-based, boundary-based and compound [35].

- Distribution-based losses aim to minimize the differences between two distributions.
- Region-based losses maximize the overlap regions between prediction and ground truth.
- Boundary-based losses minimize the distance between ground truth and predictions.
- Compound losses combine in a single equation more losses, creating a multi-objective function.

In this thesis, we tested cross entropy, dice, focal, DiceFocal and unified focal losses.

- Binary cross entropy loss is distribution-based and its objective is to minimize the difference between two distributions (in this case prediction and target)

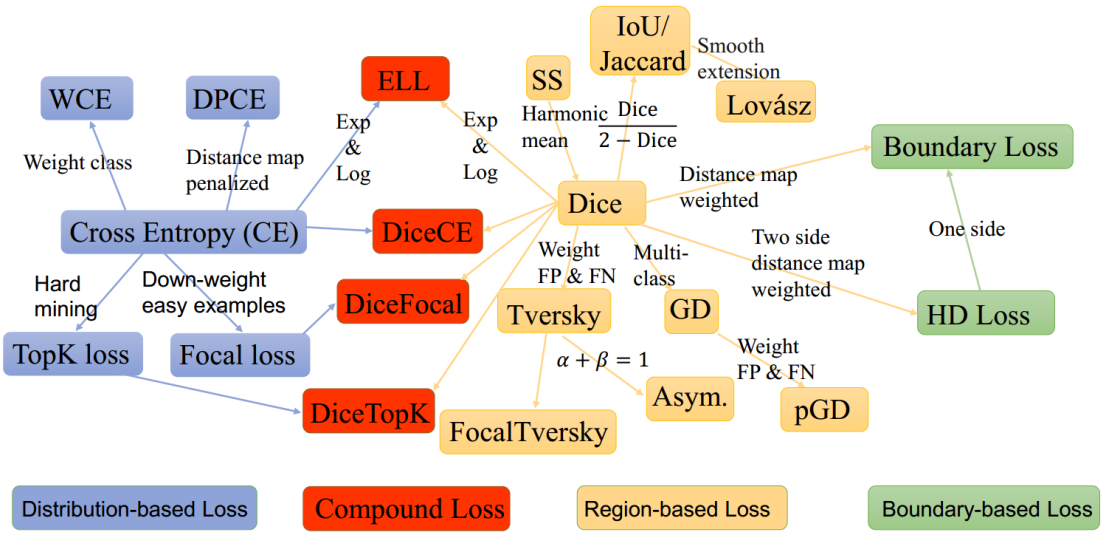


Figure 2.3: Common loss functions for segmentation [35]

giving a penalty to wrong predictions. The following is the equation for mean BCE:

$$L_{BCE} = \frac{\sum_{n=0}^{N} -y_n \cdot \log x_n - (1 - y_n) \cdot \log(1 - x_n)}{N} \tag{2.2}$$

where N is the sample count, x_n is the prediction and y_n the ground truth for n^{th} sample

- Generalized dice loss [36] is region-based and its objective is to make two areas perfectly overlapping. The function is:

$$L_{dice} = 1 - 2\frac{\sum_{l=0}^{1} w_l \sum_{n=0}^{N} y_{ln} x_{ln}}{\sum_{l=0}^{1} w_l \sum_{n=0}^{N} y_{ln} + x_{ln}} \tag{2.3}$$

where N is the sample count, l is the class label, w_l is the weight to provide label invariance to the different label set properties and x_{ln} is the prediction and y_{ln} is the ground truth for n^{th} sample for class l.

- Focal loss [37] is distribution-based and it has the same objective of binary cross entropy, but with some differences as can be seen in the equation for mean focal loss:

$$L_{focal} = \frac{\sum_{n=0}^{N} -\alpha(1 - x_n)^{\gamma} \log x_n}{N} \tag{2.4}$$

where N is the sample count, α is the weight for positive vs negative samples, γ is giving more importance to hard sample with high loss and x_n is the prediction and y_n the ground truth for n^{th} sample

- DiceFocal loss [38] is compound and it combines in a more robust function the objective of focal loss and dice loss in this way:

$$L_{DiceFocal} = \alpha L_{focal} + (1 - \alpha) L_{dice} \tag{2.5}$$

  where α is the weight given to focal loss.

- Unified Focal loss [39] is compound and it combines focal and focal twersky loss with some modifications in this way:

$$L_{AsymmetricUnifiedFocal} = \lambda L_{maFocal} + (1 - \lambda) L_{maFocalTwersky} \tag{2.6}$$

  where $L_{maFocal}$ is the Modified Asymmetric Focal loss [39], $L_{maFocalTwersky}$ is the Modified Asymmetric Focal Twersky loss [39] and λ is the weight of modified focal. The modifications allow to use less hyper-parameters.

2.2.2 CNN for semantic segmentation

For years convolutional neural networks helped solve semantic segmentation in various applications: facial recognition [40], autonomous vehicles [41], medical imaging [18] and diagnostic [42] and many others [43]. The main problem the first models faced was that using the input image size throughout the network is computationally expensive and the solution was found using an encoder/decoder structure as in Figure 2.4. In this way, the encoder makes a concise and meaningful representation of the input, especially if they are large, while the decoder elaborates this information to create a human understandable output. The evolution of this idea is U-Net, which is one of the most popular architectures in semantic segmentation for its simplicity combined with improved results and many variations were created afterwards [44]. The encoder-decoder architecture was kept with some important changes: a contracting encoding path, a symmetric expanding path and the encoded feature concatenation with decoded one (Figure 2.5). These approaches lack global contextual information and different techniques were adopted to do better scene parsing as in ParseNet[45], with the addition of global features, and PSP-Net [46], where a pyramid module harvests different sub-region representations. The last model we have to mention is surely DeepLab [47], because of three main contributions in the work: (1) they reduced the feature resolution without losing information with atrous convolution (Figure 2.9), (2) the proposed autrous spatial pyramid pooling permit to segment objects at multiple scales (Figure 2.7) and (3) they improve localization accuracy by adding a fully connected conditional random field (CRF) to capture fine details (Figure 2.8).

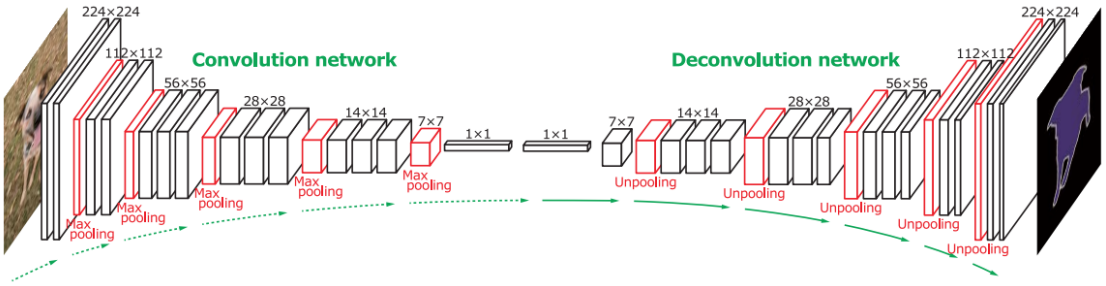


Figure 2.4: Deconvolution network [48]

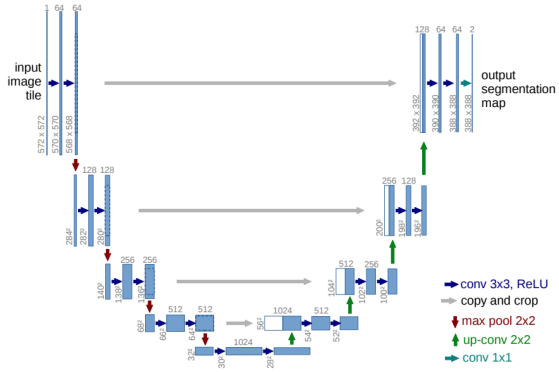


Figure 2.5: U-Net architecture [18]

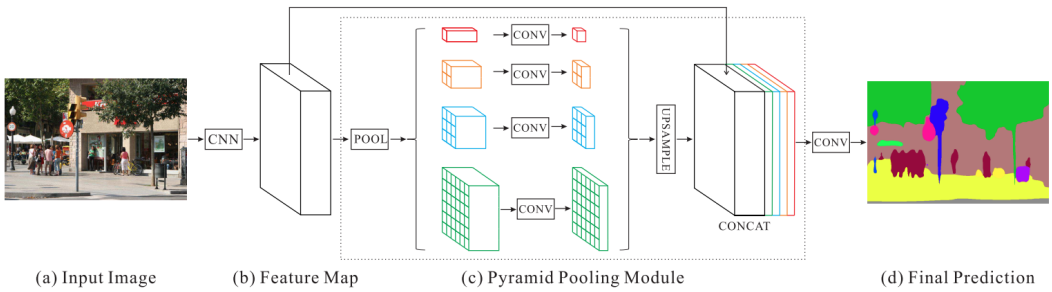


Figure 2.6: PSP-Net architecture [46]

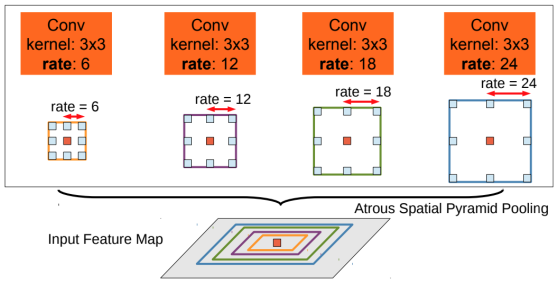


Figure 2.7: Autrous spatial pyramid pooling (ASPP) [47]. To classify the center pixel (orange) multiple filters are exploited.

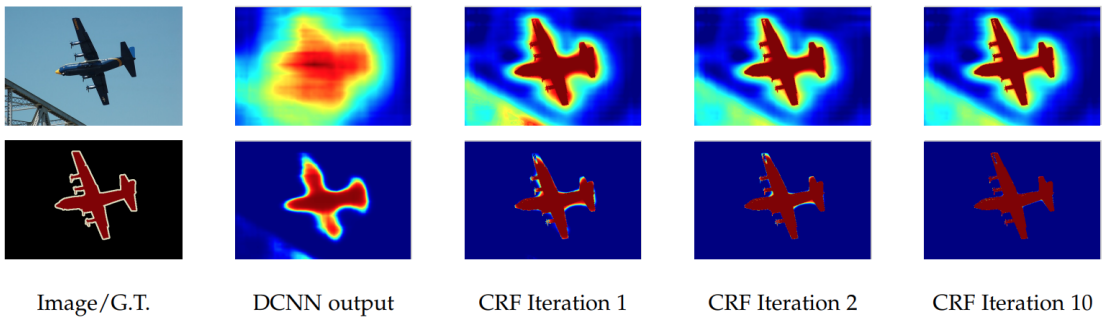


Figure 2.8: Effect of CRF on prediction heatmap (first row) and mask (second row) [47]. The successive iterations help improve the small details of the object.

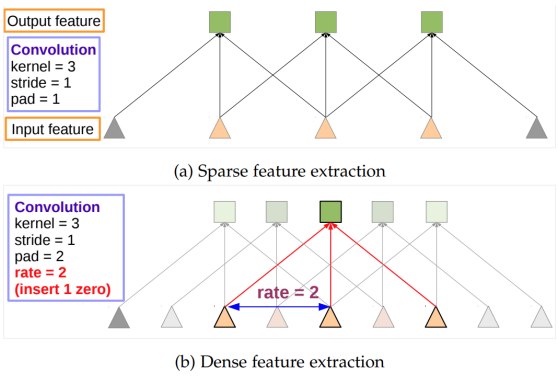


Figure 2.9: Standard convolution (first) and atrous convolution (second) [47]. The rate controls the spacing between kernel points (in this case 1 space between two points).

2.2.3 Transformer and Vision Transformer

When transformer architecture was proposed [49], CNNs were modified to exploit attention [50, 51]. With attention, the network self-learns to give different weights to various parts of the input, highlighting the most important ones. Soon, transformer-based architectures became the most used and successful ones [52] in the natural language processing (NLP) field and models such as BERT [53] and GPT-3 [54] became state-of-art. This success pushed the researcher in trying to apply attention to computer vision and many successful applications with CNNs were developed. The creation of vision transformer (ViT) [55], a pure transformer-based architecture (Figure 2.10), demonstrated convolutions are not necessary to achieve good results. It requires fewer computational resources than CNNs, too. Next to it, many approaches came out and proved their effectiveness in many different tasks: image classification [56], object detection [57], colorization [58], super-resolution for images [59] and videos [60], panoptic segmentation [26] and also semantic segmentation [17]. The various architectures focused not only on improving the results but also on solving the bottlenecks of ViT (efficiency and the need for a large amount of data [55]). Swin Transformer [57] and Twins [61] focus on improving the results, DeiT [62] tried to reduce the needed amount of data with token distillation, SegFormer [17] and LeViT [63] tried to reduce the computational cost. In this thesis, we choose SegFormer because it was designed to have a variable number of parameters based on the desired computational cost and to be more noise resistant than other vision transformer architectures [17]. Looking at Figure 2.11, it is possible to understand the architecture is different from other vision transformers such as Swin (Figure 2.12) because of (1) the hierarchical encoder which outputs multiscale features and (2) the absence of positional encoding. It was designed specifically for semantic segmentation, optimizing the computationally expensive parts: the self-attention module was optimized according to [64] and the MLP decoder does not contain any convolution, greatly reducing their costs. The network accepts an image of size $W \times H$ with C channels reducing the resolution while going deeper creating more abstract representations of the input. Each temporary result is kept because they are concatenated in the decoder to make the final prediction.

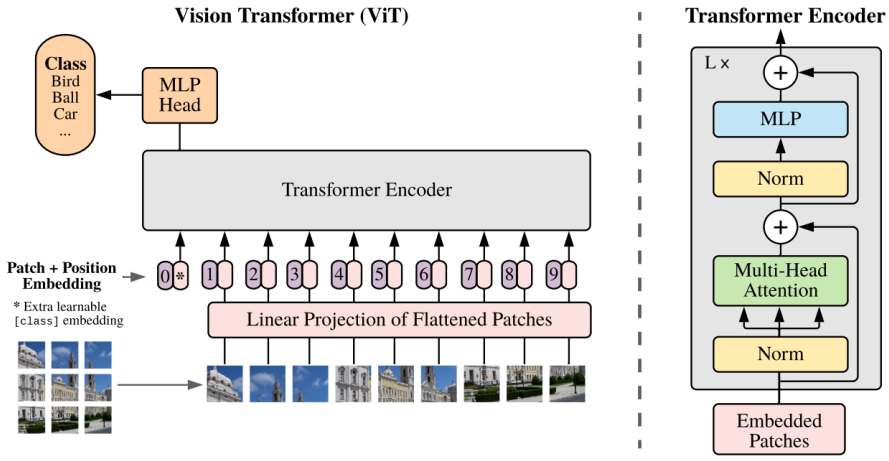


Figure 2.10: Vision transformer architecture [55]

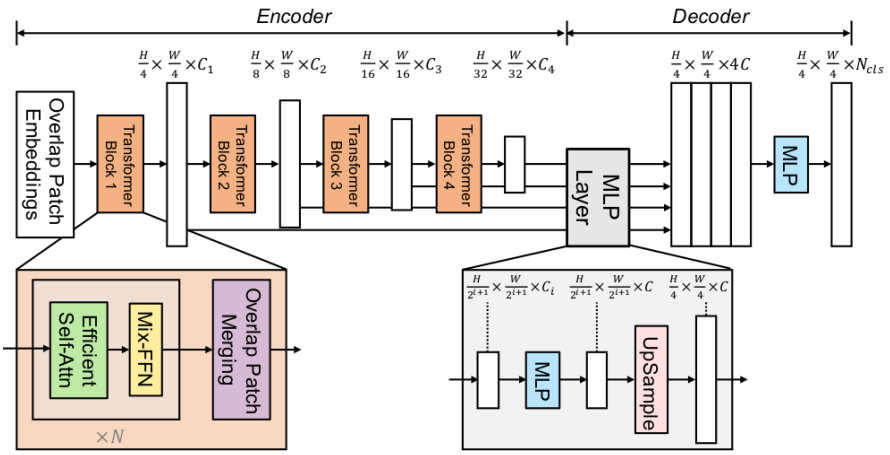


Figure 2.11: Segformer architecture [17]. Mix-FFN is Mix Feed Forward Network.

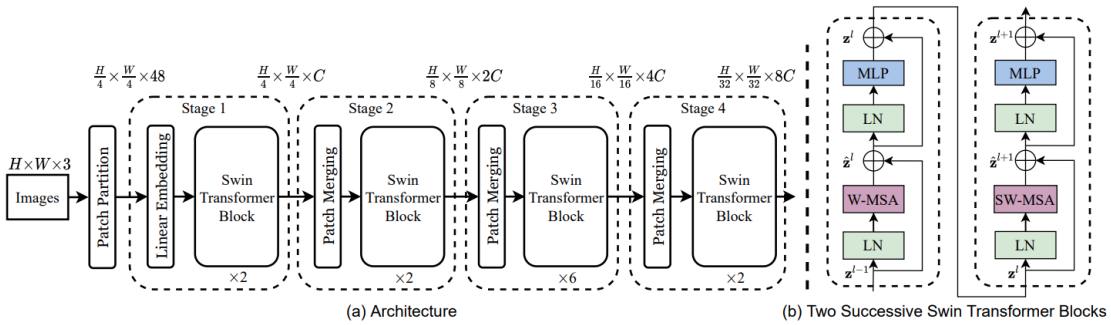


Figure 2.12: Swin architecture [57]. W-MSA and SW-MSA are multi-head self attention modules.

2.3 Burned area identification with Sentinel-2

Sentinel-2 mission monitors the green areas of the planet and give support in natural disaster management exploiting not only visible spectral band, but also infrared. The satellites provide 13 spectral bands, with most of them that are in SWIR (ShortWave InfraRed) and VNIR (Visible and Near InfraRed), because they proved to be effective in evaluating water properties, moisture content and plants health [65]. Before the advent of modern computer vision methodologies, researchers tackled the problem with the analysis of burned area indexes. Specifically, by gathering and combining information from several spectral bands which are sensitive to humidity and vegetation, it is possible to highlight regions affected by the hazardous event. Some examples and their equations are:

- Normalized Burn Ratio (NBR) [66]

$$NBR = \frac{B08 - B12}{B08 + B12} \tag{2.7}$$

- Normalized Burn Ratio 2 (NBR2) [67]

$$NBR2 = \frac{B11 - B12}{B11 + B12} \tag{2.8}$$

- Burned Area Index for Sentinel-2 (BAIS2) [68]

$$BAIS2 = \left(1 - \sqrt{\frac{B06 * B07 * B8a}{B4}}\right)\left(\frac{B12 - B8a}{\sqrt{B8a + B12}} + 1\right) \tag{2.9}$$

- relative delta Normalized Burn Ratio 2 (dNBR2) [69]

$$RdNBR = \frac{NBR_{pre} - NBR_{post}}{\sqrt{|NBR_{pre}/1000|}} \tag{2.10}$$

  where NBR_{pre} and NBR_{post} are respectively the NBR before and after the wildfire.

where Bx is a spectral band of Table 2.1. Some of them, such as the latter, perform the comparison of the burned area index before and after the wildfire to improve performances and detect drastic changes in vegetation but are heavily sensible to the presence of agricultural areas and crops. Index-based methodologies for burned area delineation are often coupled with automatic or semi-automatic [70, 71] thresholding algorithms, such as the Otsu method [72]. One of the main complications of threshold-based techniques is the choice of the most adequate

threshold, varying the vegetation type, environmental and lighting condition, making it difficult to determine a unique, universal value [73] for every region worldwide.

The latest deep learning solutions for burned area delineation focused on CNN architectures in particular on U-Net [74, 75, 76] and Siamese networks [77, 78]. Only a few attempts tried to exploit the power of the new vision transformers [79], so, in this thesis, we explore the adoption of SegFormer architecture and some variations for burned area delineation, comparing the achieved performances with U-Net and threshold-based techniques.

Band	Resolution	Central wavelength	Description
B1	60 m	443 nm	Ultra Blue (Coastal and Aerosol)
B2	10 m	490 nm	Blue
B3	10 m	560 nm	Green
B4	10 m	665 nm	Red
B5	20 m	705 nm	Visible and Near Infrared (VNIR)
B6	20 m	740 nm	Visible and Near Infrared (VNIR)
B7	20 m	783 nm	Visible and Near Infrared (VNIR)
B8	10 m	842 nm	Visible and Near Infrared (VNIR)
B8a	20 m	865 nm	Visible and Near Infrared (VNIR)
B9	60 m	940 nm	Short Wave Infrared (SWIR)
B10	60 m	1375 nm	Short Wave Infrared (SWIR)
B11	20 m	1610 nm	Short Wave Infrared (SWIR)
B12	20 m	2190 nm	Short Wave Infrared (SWIR)

Table 2.1: Sentinel-2 spectral bands [80]

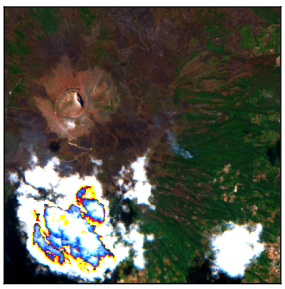
Chapter 3

Methodology

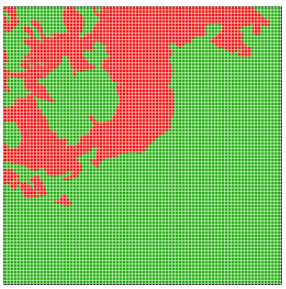
In this chapter we are going to present the data used to train and test the model and the details of the architectures and the losses we used to solve the task. Finally, we provide the experimental results.

3.1 Problem statement

Given a set of labelled satellite images of size $W \times H$, each one associated with a binary mask representing the information about the burned/unburned pixels, the goal consists in training a classification model that can then be used to predict the class label (burned/unburned) for all pixels of new images, i.e., we are interested in training a model that solves the semantic segmentation task.



Satellite image with burned area



Mask with per pixel class label

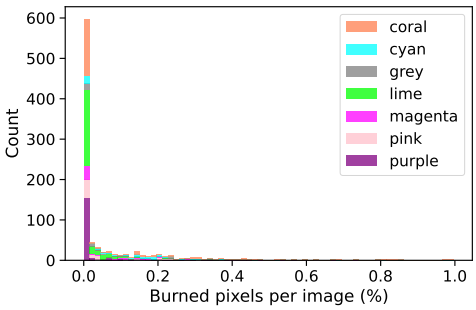
Figure 3.1: Example of an image from Sentinel-2 and mask with a label for each pixel. Red pixels are assigned to the burned class and green ones to the unburned class.

3.2 Dataset

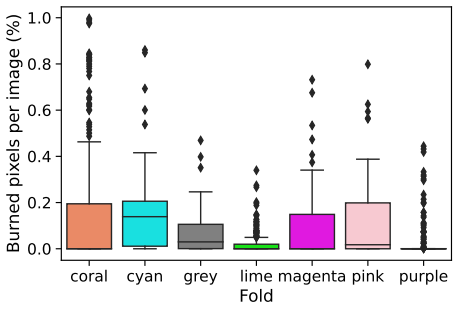
The dataset is composed of images taken from Copernicus Sentinel-2 in combination with data provided by Copernicus Emergency Management Service (EMS), which contains manually generated damage severity maps of burned regions hit by past wildfires. The EMS damage severity maps were used as ground truth, but because the quality of the information can vary a lot, some constraints were applied to select the images: (i) the satellite acquisition date must be equal to the date of the severity map, (ii) data must be available for at least the 90% of the AoI (Area of Interest), and (iii) cloud coverage must not exceed the 10% of the AoI [76]. The dataset contains images pre and post fire of 73 different AoI around Europe and they were aggregated in folds according to their geographical position [75] (see Figure 3.1). We used these folds to generalize the model using cross-validation. Data have variable resolution, up to 5000x5000 pixels. The dataset indicates burned areas with a discrete severity level, ranging from 0 (undamaged) to 4 (completely destroyed). In this thesis, we explored the burned area delineation problem and consequently, we binarize the target labels into unburned/burned classes, accordingly to our problem statement. As such, all values in range $[1, 4]$ were encoded into the burned class. We set the reference image resolution of 512×512 ($W \times H$) pixels, cropping bigger acquisitions into several images due to hardware limitations. Sentinel-2 data have 13 channels as shown in Table 2.1, but Level-2A products do not contain band 10 so we used only 12 of them. Some corrections were already applied to avoid noise related to (i) natural conditions(air turbulence, fog etc) and (ii) the influence of aerosols [76].

Original size images contain some burned areas, but after the cropping, some patches do not contain any burned pixels. Figure 3.2 shows the highly imbalanced distribution of target labels. Many images contain few pixels assigned to the burned class. Looking at the box plot we can see folds suffering from high imbalance. Thus, we exclude the cropped images without any burned pixels from the dataset, mitigating the class asymmetry. In the ablated dataset the *coral* fold is the most complete one with percentages from 0 to 1, while the others have the majority of the samples below 0.6 and *lime* even below 0.2. The assumption is reasonable because we expect our system will be applied to areas we know there have been wildfires (several public services usually provide this information).

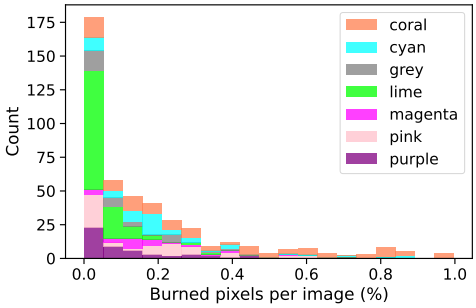
For each series of experiments, we are exploiting cross-validation and so we report for each test set, the corresponding validation and training set in Table 3.1.



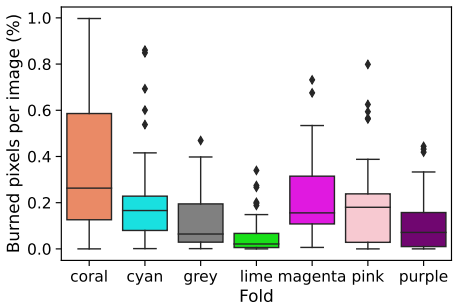
(a) Complete dataset: distribution of burned pixels per image (%)



(b) Complete dataset: percentage of burned pixels per image for each fold



(c) Images with at least one burned pixel: distribution of burned pixels per image (%)



(d) Images with at least one burned pixel: percentage of burned pixels per image for each fold

Figure 3.2: Left: distribution of the percentage of burned pixels per image. Right: distribution of the percentage of burned pixels per image for each fold.

Test set	Validation set	Training set
purple	coral	pink, grey, cyan, lime, magenta
coral	cyan	pink, grey, purple, lime, magenta
pink	coral	purple, grey, cyan, lime, magenta
grey	coral	pink, purple, cyan, lime, magenta
cyan	coral	pink, grey, purple, lime, magenta
lime	coral	pink, grey, cyan, purple, magenta
magenta	coral	pink, grey, cyan, lime, purple

Table 3.1: Sets table

3.3 Model

In this section we explain all the solution adopted to tackle the presented problem. We decided to use SegFormer [17] because it can have fewer parameters, be computationally lighter and more noise resistant than U-Net [18] and other vision transformer architectures [17]. U-Net divides perfectly the most complex models of SegFormer from the simplest ones considering the number of parameters (Table 3.2).

	SegFormer-B0	SegFormer-B1	SegFormer-B2	**U-Net**	SegFormer-B3	SegFormer-B4	SegFormer-B5
# parameters	3.8M	15.9M	27.5M	**31.0M**	47.3M	64.1M	81.4M

Table 3.2: U-Net and SegFormer versions by number of parameters

3.3.1 SegFormer

The first approach we used to address the burned area identification problem consists in finetuning a pre-trained SegFormer on our task providing as input $W \times H$ labelled images of burned/unburned areas. Then, we apply the trained model to new images to perform predictions. Since the output size is not equal to the input size, it is necessary to upsample the output image. We choose to use bilinear interpolation according to the original implementation. The chosen model is SegFormer-B3 because it has a similar but higher number of parameters than U-Net and it is the simplest of the complex versions.

3.3.2 Crop&Recompose

Furthermore, we explored a second approach which we called *Crop&Recompose*, in which the training phase was done on images of size $N \times N$, being N smaller than the reference size $W \times H$, i.e., $N \leq W$ and $N \leq H$ (to be more comfortable with calculations we choose N submultiple of W and H). In this case, we used SegFormer-B3, too. The second solution was proposed to verify the positive or negative impact of smaller patches during the training phase of SegFormer model in terms of precision of the predictions. We have smaller crops, and hence less context, but more images (in terms of images analyzed by the network at training time). However, the final goal consists in segmenting the original images of size $W \times H$, thus requiring recomposing the output to match the original input. The model is trained on smaller images of size $N \times N$ using the same architecture discussed before. Then, we apply the following approach to segment the new images (Figure 3.3), which are of size $W \times H$:

1. The original image of size $W \times H$ is cropped into M patches of size $N \times N$;

2. The M new images are passed through the model to perform the predictions;
3. The output composed of the predictions for the M images is recomposed into a single prediction/image of size $W \times H$.

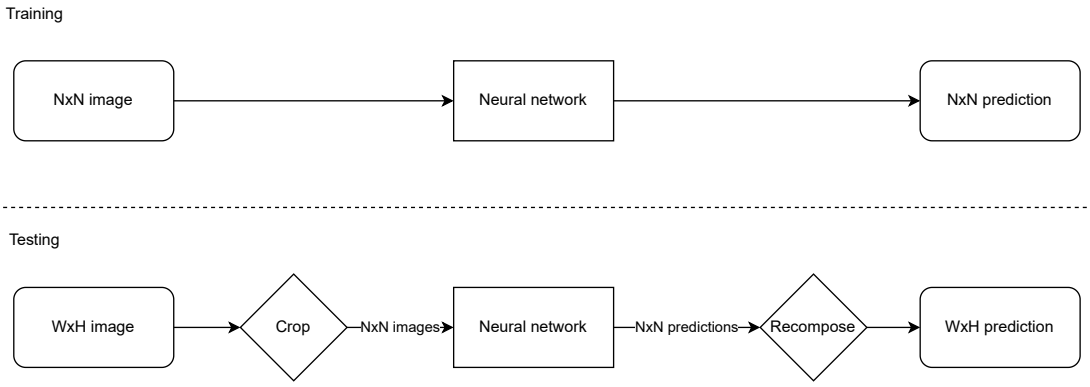


Figure 3.3: *Crop&Recompose* training and testing phases

3.3.3 Cloud coverage channel

In the third approach we added a 13th channel which indicates the cloud presence according to the formula [81]:

$$CH_{cloud} = (B03 > 0.175 \wedge \frac{B03 - B04}{B03 + B04} > 0) \vee (B03 > 0.39) \tag{3.1}$$

where CH_{cloud} is the value in the cloud channel and Bx refers to a band of Table 2.1. The motivation for this expedient can be seen in Figure 3.4, where there are some examples of images with clouds and their effects on the prediction heatmap. The cloudy areas have generally less certain predictions and in some cases, the burned area is not detected at all. The addition of this aggregation channel could help the network to focus on the cloud noise problem, although it is a redundant channel. The chosen version is SegFormer-B3.

3.3.4 Magnifier Net

The fourth approach we adopted consists of a network with two backbones working at different resolutions (Figure 3.5) as suggested in other papers [82] exploiting a well-known technique called "early fusion" [83]:

1. The original image of size $W \times H$ is cropped into N patches of size $W' \times H'$

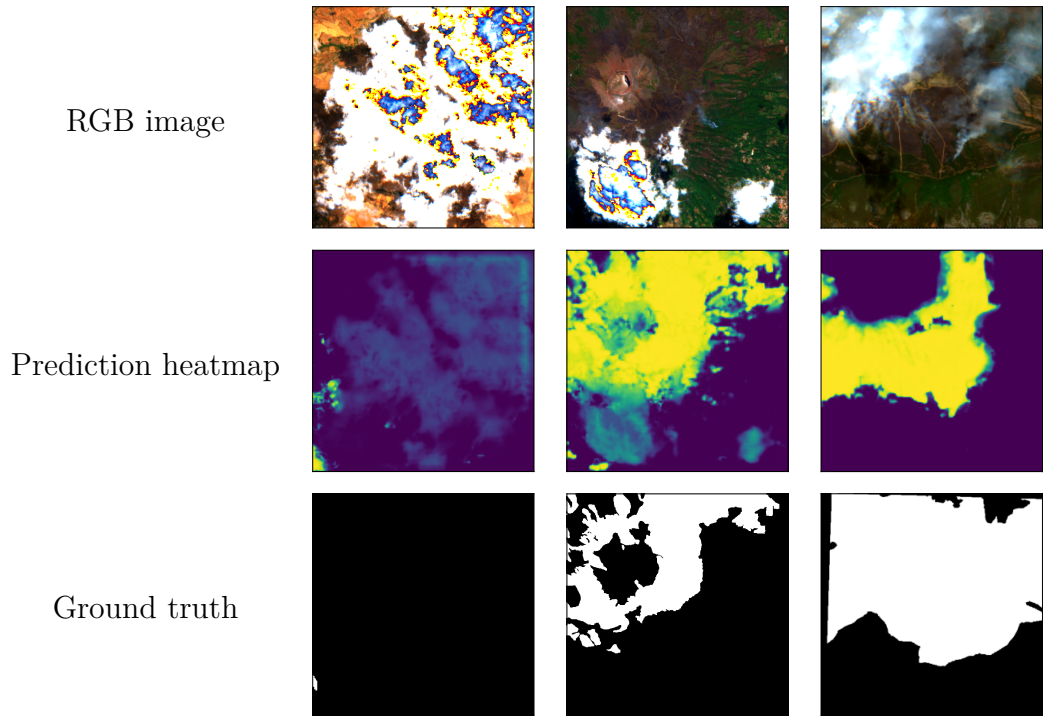


Figure 3.4: RGB images with cloud presence, heatmap of SegFormer-B3 in viridis colormap and related ground truth masks

2. The original image is passed through an encoder (called *big encoder* for simplicity)
3. The crops are passed through another encoder (called *small encoder* for simplicity)
4. The encoding of the crops is recomposed to have the same size as the original image encoding
5. The two encodings of size $\frac{W}{32} \times \frac{H}{32} \times C$ (as shown in Figure 2.11) are concatenated to have a single encoding of size $\frac{W}{32} \times \frac{H}{32} \times 2C$
6. The fused encoding is passed through the decoder

3.3.5 Losses and Metrics

Different loss functions were evaluated each one with a certain objective. To address the unbalanced problem of burned area delineation, we initially considered the dice loss and then we explored the possibility to use compound losses to reach a better

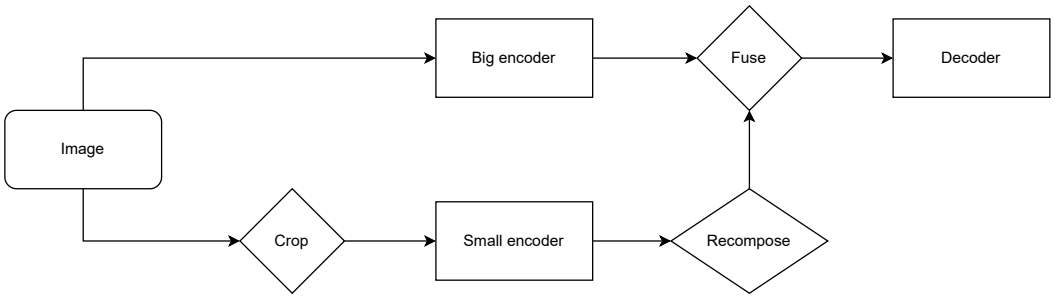


Figure 3.5: Magnifier net

stability point. In particular, we have evaluated binary cross entropy (2.2), dice (2.3), focal (2.4), dice focal (2.5) and asymmetric unified focal (2.6) losses.

To evaluate the goodness of our model we used three known metrics: precision, recall and F1 score. The choice was necessary because of the strong imbalance of the problem as shown in Section 3.2. Precision is the fraction of relevant information among all retrieved instances, while recall is the fraction of retrieved information among all relevant ones. F1 score is simply the harmonic mean of precision and recall. Following the formulas:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

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

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{3.4}$$

where TP is true positive count (number of relevant information the model recognized as relevant), FP is false positive count (number of non-relevant information the model classified as relevant) and FN is false negative count (number of relevant information the model not recognized as relevant).

Chapter 4

Experiments

In this section we are presenting the results of the experiments done using the previously presented methodologies with different losses.

4.1 Settings

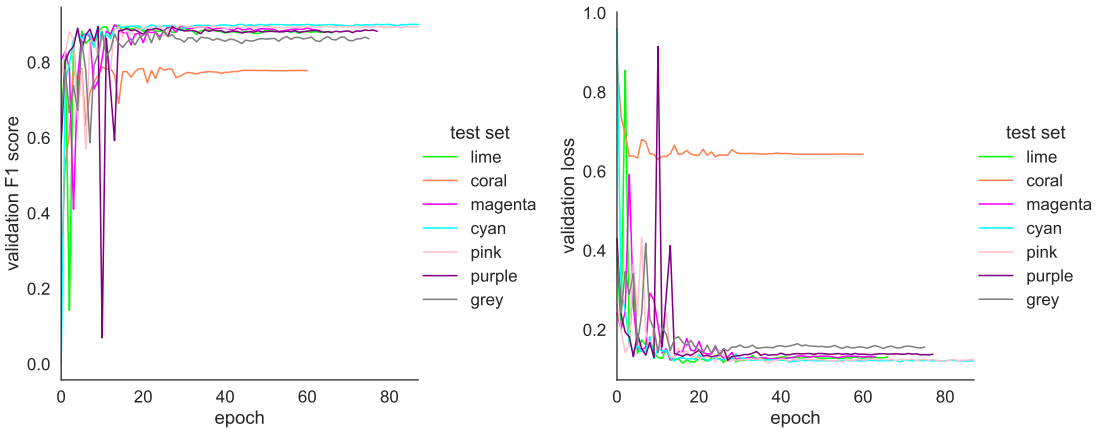
The experiments were run on a single Tesla V100. We made use of PyTorch Lightning framework [84] and the SegFormer implementation of HuggingFace [52] with a pre-trained encoder on Imagenet-1K, but because the original model has only 3 channels (RGB), we replicated the weights for all the 12 channels of the satellite images 4 times cyclically. This allowed us to leverage the pre-trained model even if the number of input channels is different. The applied mapping (satellite image band, RGB channel) is as follows: (B01,R), (B02,R), (B03,G), (B04,B), (B05,G), (B06,B), (B07,R), (B08,G), (B09,B), (B10,R), (B11,G), and (B12,B). Image resolution is set to 512×512 except for the *Crop&Recompose* method, in which a size of 64×64 was used. For *Magnifier Net* the *Big encoder* resolution is 512×512 and the *Small encoder* one is 64×64. These settings permit to use the weights from the other experiments and freezing the two encoders, finetuning only the decoder. We used the AdamW optimizer as in [17] and the starting learning rate was set to 0.001. A decreasing scheduler was chosen to reduce the LR by a factor of 10 every 15 epochs (instead of the polynomial learning rate scheduler used in the original paper) in conjunction with an early stopping mechanism on validation loss, with a tolerance of 10^{-4} and patience of 50 epochs. The maximum number of epochs is 200 and the batch size is 8. To provide better generalization the dataset was augmented with some transformations (the same used in [76]):

Transformation	Probability	Parameters
Random rotation	0.5	Angle: [-50°, 50°]
Random vertical flipping	0.5	-
Random horizontal flipping	0.5	-
Random shear	0.5	Angle: [-20°, 20°]

Table 4.1: Data augmentations and their parameters

4.2 Dice loss

In this series of experiments, we evaluated the effects of dice loss on SegFormer-B3. Dice loss has some hyper-parameters inside the formulation, but they are self-computed according to [36]. From Figure 4.1 we can see a fast convergence of the model. After 20 epochs the model stabilizes the metrics, reaching nearly the same value for the majority of the sets. Training without validating on coral fold seems to affect the model learning: this is probably linked to the fact coral fold is the most "complete" as shown in Figure 3.2. The comparison with the same loss applied in U-Net in Table 4.2 shows how SegFormer performs slightly better in terms of precision and consequently in F1-score, while U-Net has higher values of recall. This means U-Net is overestimating the burned areas. Taking into consideration the standard deviation, SegFormer gets more stable results across the different folds. *Lime* got the worst performances in terms of precision and F1 score for both models, while *grey*, *lime* and *cyan* have low values of recall.



(a) Validation F1 score

(b) Validation loss

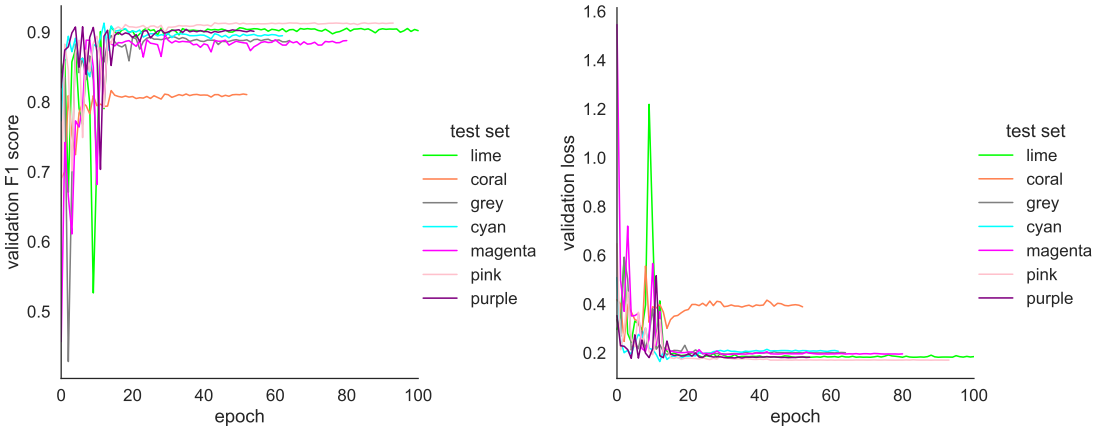
Figure 4.1: Dice loss SegFormer-B3 training.

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Dice MiT-B3	**0.899**	0.790	0.762	**0.712**	0.877	**0.909**	**0.899**	**0.835**	0.080
	Dice U-Net	0.895	**0.797**	**0.817**	0.506	**0.883**	0.907	0.894	0.814	0.142
Precision	Dice MiT-B3	**0.898**	**0.828**	**0.859**	**0.655**	**0.866**	**0.898**	**0.897**	**0.843**	0.087
	Dice U-Net	0.829	0.790	0.704	0.356	0.801	0.848	0.861	0.741	0.177
Recall	Dice MiT-B3	0.901	0.755	0.685	0.779	0.888	0.920	0.902	0.833	0.092
	Dice U-Net	**0.972**	**0.804**	**0.973**	**0.869**	**0.985**	**0.974**	**0.930**	**0.930**	0.068

Table 4.2: Test metrics comparison for Dice loss SegFormer-B3

4.3 Binary cross entropy loss

In this series of experiments, we evaluated application of binary cross entropy loss (BCE) to SegFormer-B3. From Figure 4.2 we can see a fast convergence of the model as before. The loss is lower than using dice loss, but the trend is the same as before. As seen in Table 4.3, BCE does not seem capable to overtake dice loss in terms of precision and F1 score, but it is still better than U-Net. The recall is the only metric which is higher, but it can not reach the top value. The most difficult fold for BCE is *grey* for all the three metrics, while the best performances are obtained in *pink* and *purple*.



(a) Validation F1 score

(b) Validation loss

Figure 4.2: BCE loss SegFormer-B3 training.

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	BCE MiT-B3	**0.900**	**0.804**	0.706	**0.715**	0.864	**0.910**	**0.912**	0.830	0.090
	Dice MiT-B3	0.899	0.790	0.762	0.712	0.877	0.909	0.899	**0.835**	0.080
	Dice U-Net	0.895	0.797	**0.817**	0.506	**0.883**	0.907	0.894	0.814	0.142
Precision	BCE MiT-B3	**0.902**	**0.836**	0.652	0.654	**0.871**	0.866	**0.929**	0.816	0.115
	Dice MiT-B3	0.898	0.828	**0.859**	**0.655**	0.866	**0.898**	0.897	**0.843**	0.087
	Dice U-Net	0.829	0.790	0.704	0.356	0.801	0.848	0.861	0.741	0.177
Recall	BCE MiT-B3	0.897	0.774	0.768	0.787	0.857	0.958	0.896	0.848	0.073
	Dice MiT-B3	0.901	0.755	0.685	0.779	0.888	0.920	0.902	0.833	0.092
	Dice U-Net	**0.972**	**0.804**	**0.973**	**0.869**	**0.985**	**0.974**	**0.930**	**0.930**	0.068

Table 4.3: Test metrics comparison for BCE loss SegFormer-B3

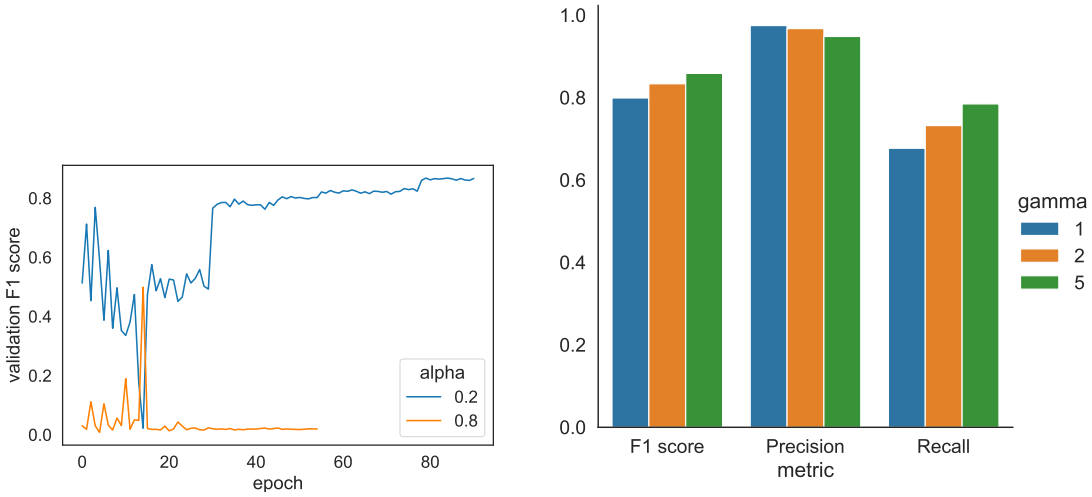
4.4 Focal loss

In this series of experiments, we are evaluating focal loss with SegFormer-B3. We are expecting better results because it is designed for imbalanced datasets.

Focal loss has some hyper-parameters, so before doing complete cross-validation, we invested some tries into their selection. At first, we focused on α parameter (equation 2.4). Knowing the number of positives is about 4 times smaller than negatives, we expect a good choice can be 0.2. Then we tried some values for γ (1, 2, 5) as suggested by the original paper [37]. In Figure 4.3, it is possible to see high values of α (near 1.0) affect negatively the training phase, so we suppose a lower value such as 0.2 will be a better choice. It is also possible to see how γ affects the testing results, so we select $\gamma = 5$ because the recall is slightly better than for other values and the F1 score improved.

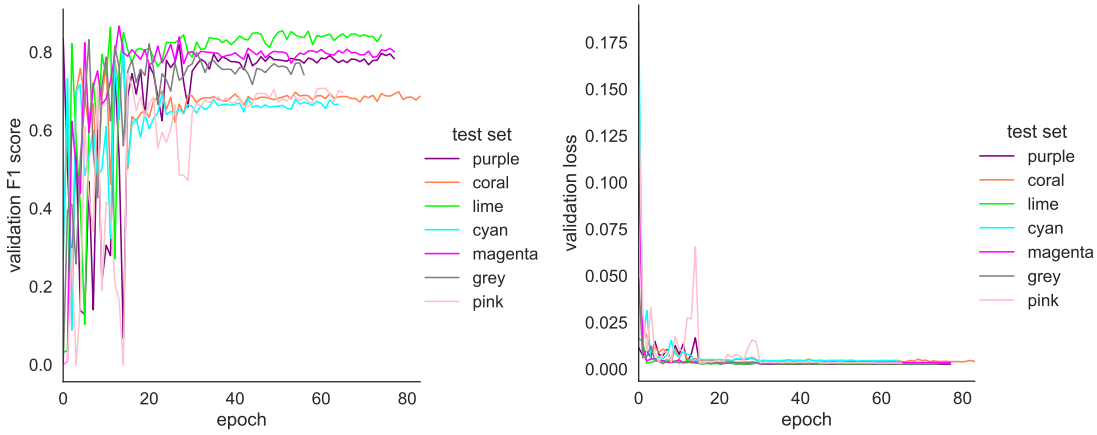
Looking at Figure 4.4, it is possible to see the training phase is not affected by the absence of *coral* in the validation phase, reaching lower losses in every case quicker (less than 20 epochs), but getting lower F1 scores, too.

Table 4.4 highlights the high precision of focal loss (+5% than dice loss), failing in getting better recall (−20% than dice loss), affecting negatively the final F1 score. The stability of the performance is confirmed for all three metrics getting better or similar standard deviations compared to dice loss. *Lime* and *grey* folds are the most difficult folds for both losses applied to SegFormer, while *pink* and *purple* show the best results.



(a) Validation F1 score varying alpha **(b)** Metrics varying gamma

Figure 4.3: Focal loss parameters tuning on test set *purple*



(a) Validation F1 score **(b)** Validation loss

Figure 4.4: Focal loss SegFormer-B3 training

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Dice MiT-B3	**0.899**	0.790	0.762	**0.712**	0.877	**0.909**	**0.899**	**0.835**	0.080
	Dice U-Net	0.895	**0.797**	**0.817**	0.506	**0.883**	0.907	0.894	0.814	0.142
	Focal MiT-B3	0.694	0.732	0.661	0.650	0.830	0.716	0.859	0.734	0.081
Precision	Dice MiT-B3	0.898	0.828	0.859	0.655	0.866	0.898	0.897	0.843	0.087
	Dice U-Net	0.829	0.790	0.704	0.356	0.801	0.848	0.861	0.741	0.177
	Focal MiT-B3	**0.931**	**0.880**	**0.877**	**0.753**	**0.910**	**0.990**	**0.948**	**0.898**	0.075
Recall	Dice MiT-B3	0.901	0.755	0.685	0.779	0.888	0.920	0.902	0.833	0.092
	Dice U-Net	**0.972**	**0.804**	**0.973**	**0.869**	**0.985**	**0.974**	**0.930**	**0.930**	0.068
	Focal MiT-B3	0.553	0.626	0.531	0.572	0.762	0.561	0.785	0.627	0.104

Table 4.4: Test metrics comparison for Focal loss SegFormer-B3

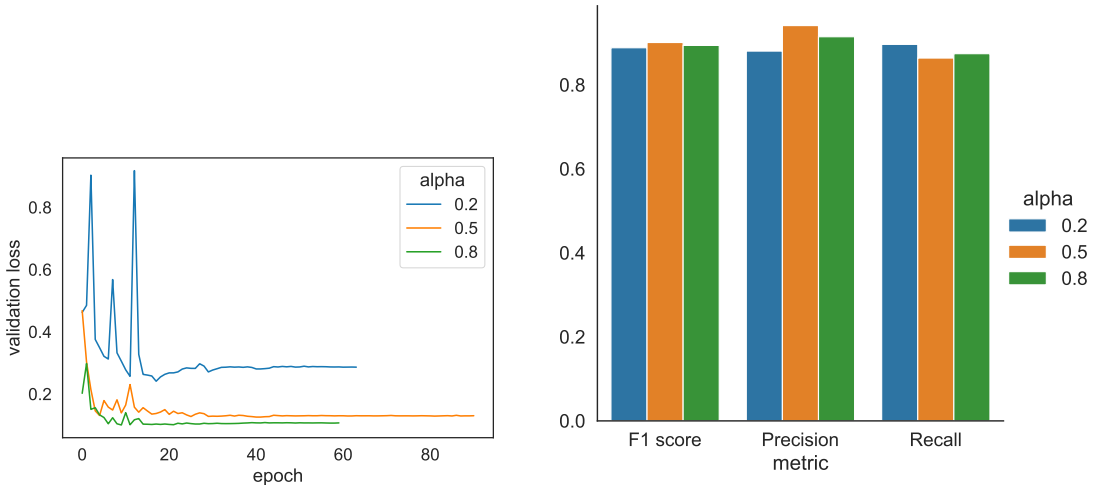
4.5 DiceFocal loss

This series of experiments focus on DiceFocal loss applied to SegFormer-B3. The combination of multiple losses is generally a good way to solve noise problems and to take into account different objectives, so by combining the precision of focal loss with the better recall of dice loss we could improve the final F1 score.

We kept the previously selected hyper-parameters for focal loss ($\alpha = 0.2, \gamma = 5$), but we have to choose a good α for DiceFocal loss (equation 2.5), too. We tried three different values (0.2, 0.5, 0.8) and in Figure 4.5 it is possible to see the effect on the validation and testing phases. Giving too much weight to dice ($\alpha = 0.2$) the validation loss is worse, but it grants better recall at test phase. Giving more weight to focal ($\alpha = 0.8$) the loss and the precision are affected positively. We selected $\alpha = 0.5$ because seems to grant a higher F1 score and precision, not decreasing too much the recall.

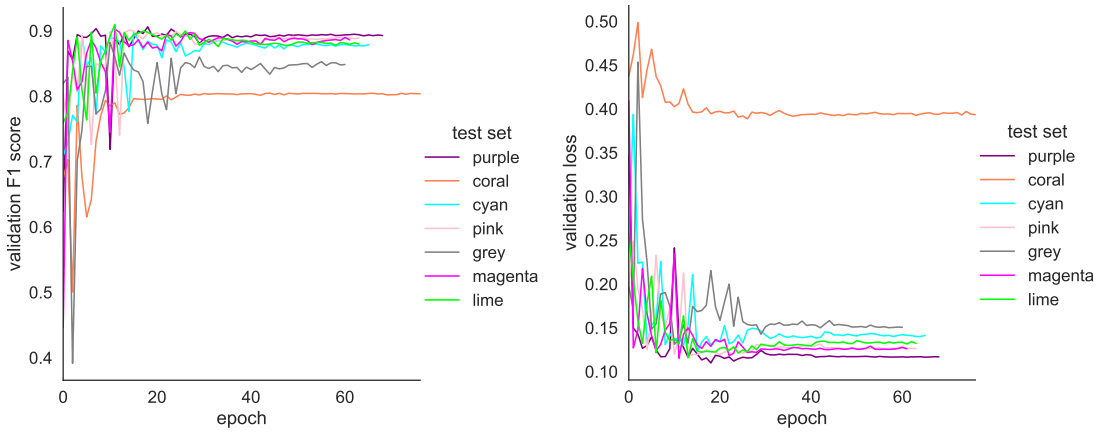
After the hyperparameters selection, we did complete cross-validation. In Figure 4.6, it is possible to note the training phase does not show any substantial difference with respect to previous training and excluding *coral* from the training-validation phase affects the validation metrics as seen before.

From Table 4.5 we can see how DiceFocal got better recall than dice and focal only, not reaching U-Net. It also got better precision than dice only, not reaching focal only results. These settings reach the best F1 score in terms of the mean value (+1% than dice loss), granting a general better stability over all three metrics. The worst performances are confirmed over *grey* and *lime* folds, while *pink* and *purple* get the best ones.



(a) Validation loss varying alpha **(b)** Metrics varying alpha

Figure 4.5: DiceFocal loss parameter tuning on test set *purple*



(a) Validation F1 score **(b)** Validation loss

Figure 4.6: DiceFocal loss SegFormer-B3 training.

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Dice U-Net	**0.895**	0.797	**0.817**	0.506	**0.883**	0.907	0.894	0.814	0.142
	DiceFocal MiT-B3	0.891	**0.805**	0.788	**0.721**	**0.883**	**0.923**	**0.907**	**0.845**	0.075
	Focal MiT-B3	0.694	0.732	0.661	0.650	0.830	0.716	0.859	0.734	0.081
Precision	Dice U-Net	0.829	0.790	0.704	0.356	0.801	0.848	0.861	0.741	0.177
	DiceFocal MiT-B3	0.901	0.823	0.876	0.693	0.871	0.893	0.922	0.854	0.078
	Focal MiT-B3	**0.931**	**0.880**	**0.877**	**0.753**	**0.910**	**0.990**	**0.948**	**0.898**	0.075
Recall	Dice U-Net	**0.972**	**0.804**	**0.973**	**0.869**	**0.985**	**0.974**	**0.930**	**0.930**	0.068
	DiceFocal MiT-B3	0.880	0.787	0.716	0.751	0.894	0.955	0.892	0.839	0.088
	Focal MiT-B3	0.553	0.626	0.531	0.572	0.762	0.561	0.785	0.627	0.104

Table 4.5: Test metrics comparison for DiceFocal loss SegFormer-B3

4.6 Asymmetric Unified Focal loss

In this series of experiments, we trained SegFormer-B3 with Asymmetric Unified Focal loss. Since the main problem of DiceFocal seems to favour too much high precision, not considering the recall, this formulation can help solve the issue. In this loss, we have to select 3 different hyperparameters: λ (weight of focal loss), γ (weight to control rare class enhancement) and δ (relative weight of positive samples vs negatives).

In the original paper, they suggest using $\delta = 0.6$ to balance recall and precision. Our experiments (Figure 4.7) also suggest using this value, although the F1 score is higher with $\delta = 0.2$ there is more difference between precision and recall. With $\gamma = 0.1$, as shown in Figure 4.7, we got not only the best precision, but also a good recall. The experiments show the best value for λ is 0.5, because it got not only a recall similar to $\lambda = 0.8$, but also the best precision with respect to other tries (Figure 4.7).

The training phase seems to be more unstable in the initial steps (Figure 4.8), but subsequently, they reach a stable point with losses comparable to DiceFocal ones. The training without *coral* reached lower losses than in the case of DiceFocal.

Table 4.6 shows how we reached a better recall, although we lose some percentage points in precision. The F1 score got a general improvement of +1% on DiceFocal loss, decreasing the results on *lime*, but increasing the ones on *grey*.

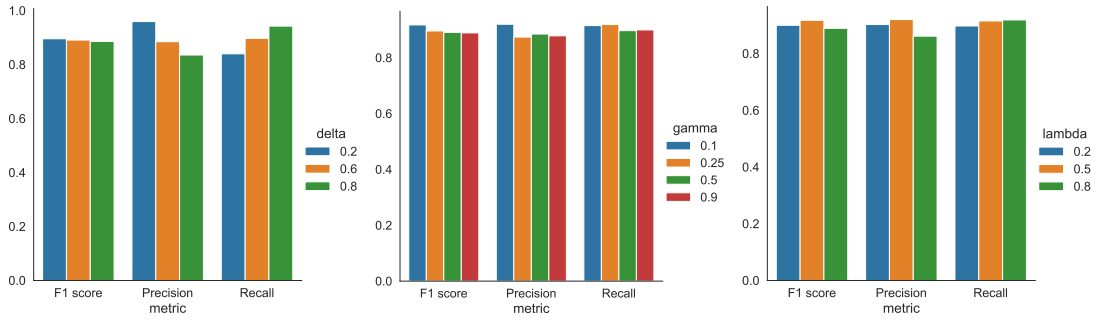
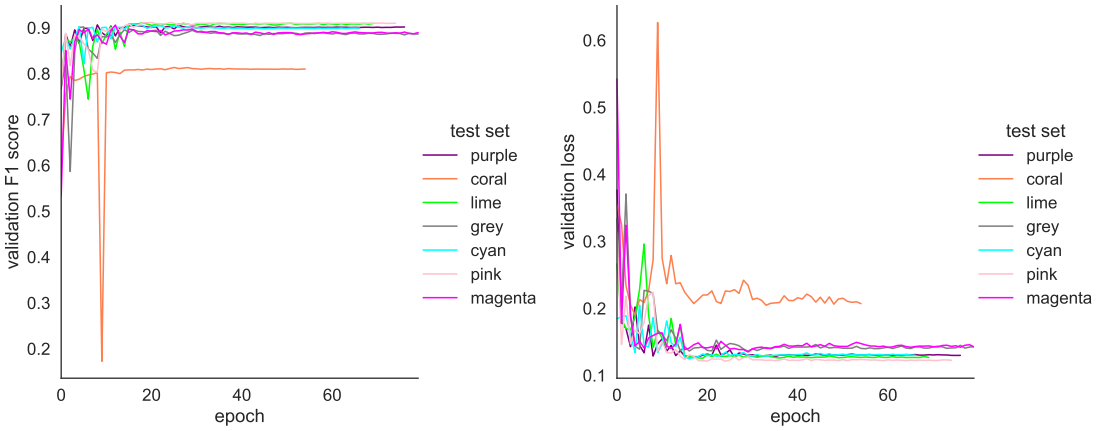


Figure 4.7: Test metrics grouped by parameter on test set *purple*



(a) Validation F1 score

(b) Validation loss

Figure 4.8: Asymmetric Unified Focal loss SegFormer-B3 training.

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Dice U-Net	0.895	0.797	0.817	0.506	0.883	0.907	0.894	0.814	0.142
	DiceFocal MiT-B3	0.891	**0.805**	0.788	**0.721**	0.883	**0.923**	0.907	0.845	0.075
	UnifiedFocal MiT-B3	**0.902**	**0.805**	**0.857**	0.688	**0.888**	0.915	**0.918**	**0.853**	0.083
Precision	Dice U-Net	0.829	0.790	0.704	0.356	0.801	0.848	0.861	0.741	0.177
	DiceFocal MiT-B3	**0.901**	0.823	**0.876**	**0.693**	**0.871**	**0.893**	**0.922**	**0.854**	0.078
	UnifiedFocal MiT-B3	0.897	**0.828**	0.842	0.574	0.855	0.866	0.920	0.826	0.115
Recall	Dice U-Net	**0.972**	**0.804**	**0.973**	**0.869**	**0.985**	**0.974**	**0.930**	**0.930**	0.068
	DiceFocal MiT-B3	0.880	0.787	0.716	0.751	0.894	0.955	0.892	0.839	0.088
	UnifiedFocal MiT-B3	0.908	0.783	0.872	0.857	0.925	0.970	0.915	0.890	0.060

Table 4.6: Test metrics comparison for Asymmetric Unified Focal loss SegFormer-B3

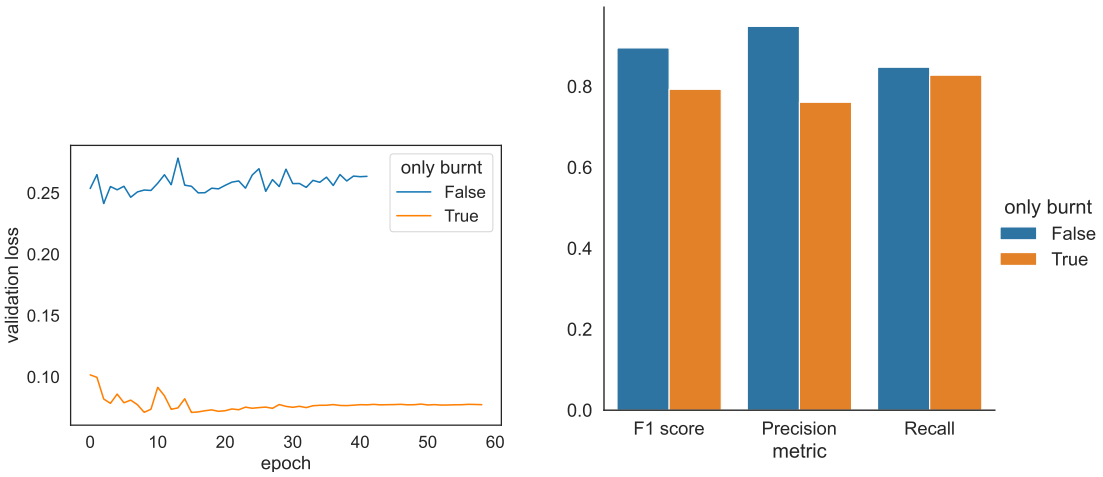
4.7 Crop & Recompose

In this series of experiments, we reduce the size of the images in training phase to 64×64 and in test phase we crop images of size 512×512 into patches of size 64×64 and then we recompose them as explained in section 3.3.2. The model is SegFormer-B3 using DiceFocal loss with the previously selected parameters.

Before starting complete cross-validation, we want to understand if the training dataset can exclude images without any burned pixels or if it is better to keep them because in bigger images there will be a lot of unburned areas. From Figure 4.9 it is possible to see the validation loss is generally lower for the filtered dataset, but the model trained on all possible images gets slighter better results for all metrics in the testing phase.

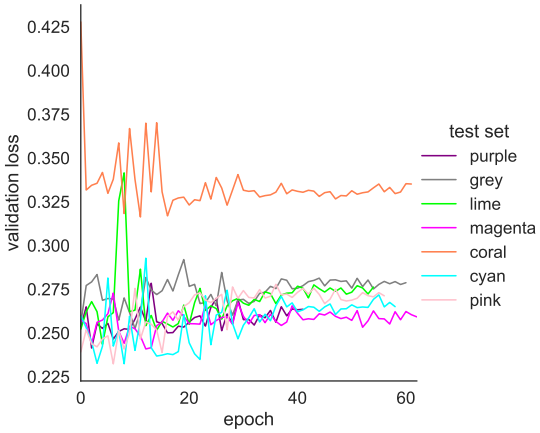
In the training phase, the convergence is as fast as expected because of the higher number of images. The model tends to over-fit early after about 20 epochs 4.10.

From Table 4.7 we can conclude using small patches with less context is affecting negatively precision and F1 score, but not the recall. These values are generally not too much different from the ones for greater patches, so we can assume the context can be inferred from small images, too.



(a) Validation loss **(b)** Test metrics

Figure 4.9: Metrics including vs excluding from training and validation datasets images without any burned pixels. Test fold is *purple*.



(a) Validation loss

Figure 4.10: Crop&Recompose validation loss

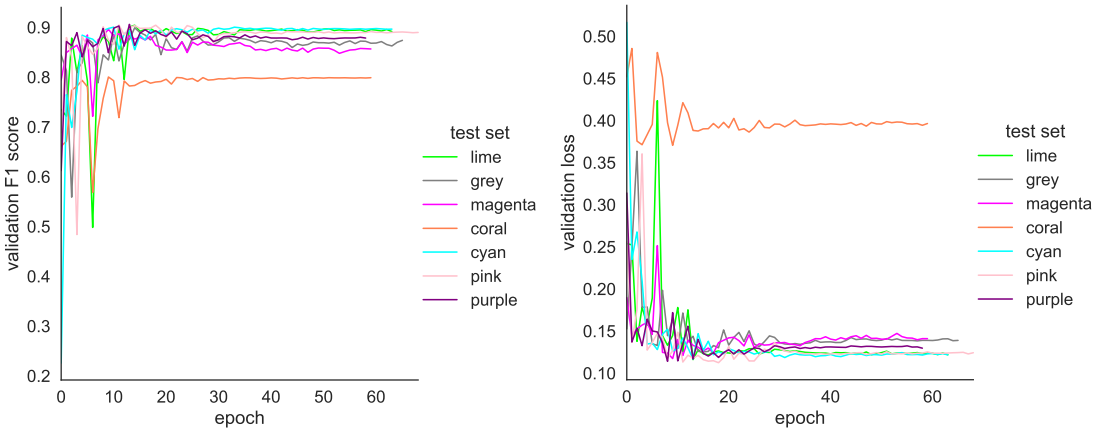
		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Crop&Recompose MiT-B3	**0.909**	0.791	0.778	0.694	**0.897**	0.917	0.895	0.840	0.086
	Dice U-Net	0.895	0.797	**0.817**	0.506	0.883	0.907	0.894	0.814	0.142
	DiceFocal MiT-B3	0.891	**0.805**	0.788	**0.721**	0.883	**0.923**	**0.907**	**0.845**	0.075
Precision	Crop&Recompose MiT-B3	0.881	0.806	0.852	0.657	0.854	**0.903**	**0.948**	0.843	0.094
	Dice U-Net	0.829	0.790	0.704	0.356	0.801	0.848	0.861	0.741	0.177
	DiceFocal MiT-B3	**0.901**	**0.823**	**0.876**	**0.693**	**0.871**	0.893	0.922	**0.854**	0.078
Recall	Crop&Recompose MiT-B3	0.938	0.777	0.717	0.736	0.944	0.932	0.847	0.842	0.099
	Dice U-Net	**0.972**	**0.804**	**0.973**	**0.869**	**0.985**	**0.974**	**0.930**	**0.930**	0.068
	DiceFocal MiT-B3	0.880	0.787	0.716	0.751	0.894	0.955	0.892	0.839	0.088

Table 4.7: Test metrics comparison for *Crop&Recompose*

4.8 Cloud coverage channel

In this series of experiments, we train SegFormer-B3 using DiceFocal loss with the previously selected parameters. In this case, we used 13 channels instead of 12 as explained in section 3.3.3. Looking at Figure 4.11 there is nothing different to note, everything appears coherent with the DiceFocal SegFormer-B3 with 12 channels.

From Table 4.8 we can see adding the cloud channel is providing better precision, but the recall is negatively affected. The differences are not too evident, leaving the F1 score practically unaltered, so we can conclude introducing some redundancy will not grant sufficient benefits to the model. The analysis of some images (Figure 4.12) highlights how the model is forced to take clearer decisions with the introduction of cloud coverage channel, but this does not always mean a better decision is taken. The images suffering from missing parts due to cloud noise seem to be getting no advantages using one more channel.



(a) Validation F1 score **(b)** Validation loss

Figure 4.11: Cloud channel SegFormer-B3 training

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Cloud Channel MiT-B3	0.896	0.796	**0.796**	**0.723**	0.875	**0.925**	0.897	0.844	0.073
	Dice MiT-B3	**0.899**	0.790	0.762	0.712	0.877	0.909	0.899	0.835	0.080
	DiceFocal MiT-B3	0.891	**0.805**	0.788	0.721	**0.883**	0.923	**0.907**	**0.845**	0.075
Precision	Cloud Channel MiT-B3	0.896	**0.828**	0.862	0.683	0.869	**0.917**	**0.940**	**0.856**	0.085
	Dice MiT-B3	0.898	**0.828**	0.859	0.655	0.866	0.898	0.897	0.843	0.087
	DiceFocal MiT-B3	**0.901**	0.823	**0.876**	**0.693**	**0.871**	0.893	0.922	0.854	0.078
Recall	Cloud Channel MiT-B3	0.896	0.767	**0.740**	0.768	**0.883**	0.934	0.857	0.835	0.076
	Dice MiT-B3	**0.901**	0.755	0.685	**0.779**	0.888	0.920	**0.902**	0.833	0.092
	DiceFocal MiT-B3	0.880	**0.787**	0.716	0.751	**0.894**	**0.955**	0.892	**0.839**	0.088

Table 4.8: Test metrics comparison for Cloud channel SegFormer-B3

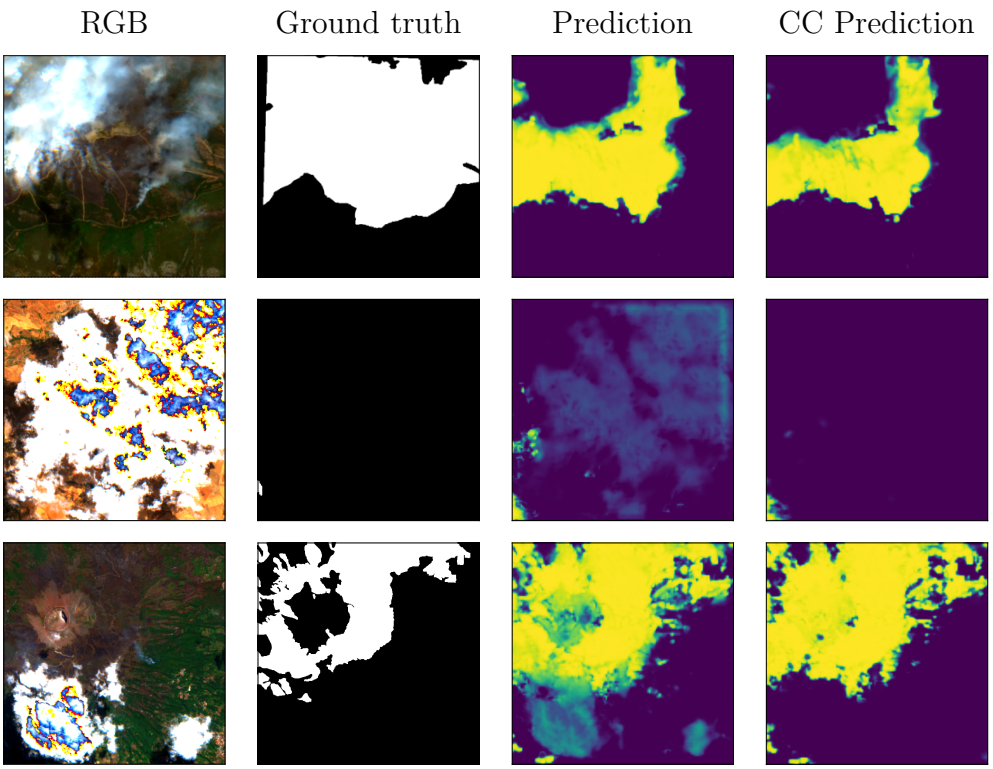
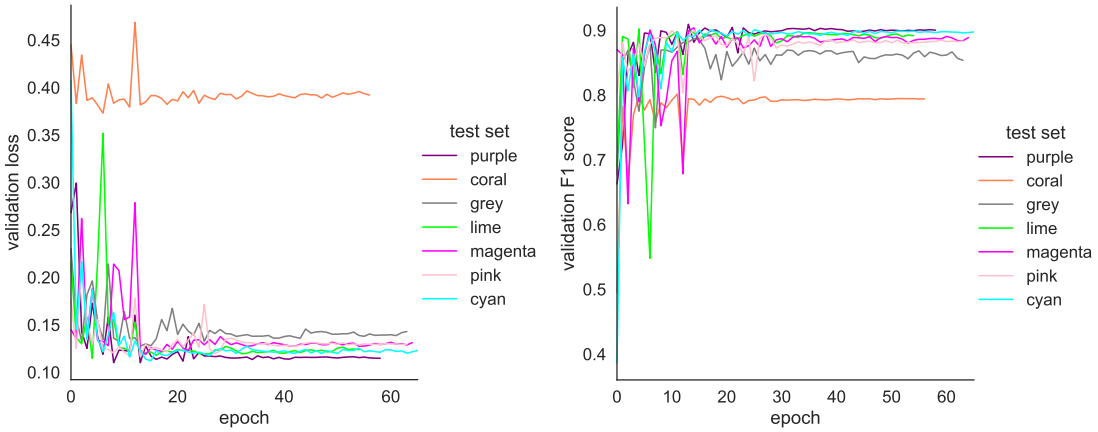


Figure 4.12: Comparison of prediction with and without cloud channel. *RGB* is the input image, *Ground truth* is the expected mask, *Prediction* is the prediction heatmap in viridis colormap without cloud channel and *CC Prediction* is the prediction heatmap with the cloud channel.

4.9 SegFormer-B0

In this series of experiments, we train the lighter version of SegFormer, SegFormer-B0 (Table 3.2), to see if there is a performance degrading or can achieve competitive results due to the simplicity of the task. We keep DiceFocal loss with parameters $\alpha = 0.2, \gamma = 5$. In the training phase, the validation metrics are consistent with the ones seen with SegFormer-B3 (Figure 4.13).

In the testing phase, we can see a downgrade in performance as expected (Table 4.9), but considering the low number of parameters (12 times less than MiT-B3 as shown in Table 3.2) and the low computational cost (9 times lower than MiT-B3 [17]) it can be considered a good competitor.



(a) Validation loss **(b)** Validation F1 score

Figure 4.13: DiceFocal loss SegFormer-B0 training

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Dice MiT-B3	**0.899**	0.790	0.762	0.712	0.877	0.909	0.899	0.835	0.080
	DiceFocal MiT-B0	0.898	0.787	0.755	0.671	**0.884**	**0.927**	**0.908**	0.833	0.096
	DiceFocal MiT-B3	0.891	**0.805**	**0.788**	**0.721**	0.883	0.923	0.907	**0.845**	0.075
Precision	Dice MiT-B3	0.898	**0.828**	0.859	0.655	0.866	0.898	0.897	0.843	0.087
	DiceFocal MiT-B0	0.876	0.810	0.864	0.613	**0.879**	**0.926**	**0.935**	0.843	0.110
	DiceFocal MiT-B3	**0.901**	0.823	**0.876**	**0.693**	0.871	0.893	0.922	**0.854**	0.078
Recall	Dice MiT-B3	0.901	0.755	0.685	**0.779**	0.888	0.920	**0.902**	0.833	0.092
	DiceFocal MiT-B0	**0.920**	0.766	0.670	0.742	0.888	0.929	0.882	0.828	0.101
	DiceFocal MiT-B3	0.880	**0.787**	**0.716**	0.751	**0.894**	**0.955**	0.892	**0.839**	0.088

Table 4.9: Test metrics comparison for DiceFocal loss SegFormer-B0

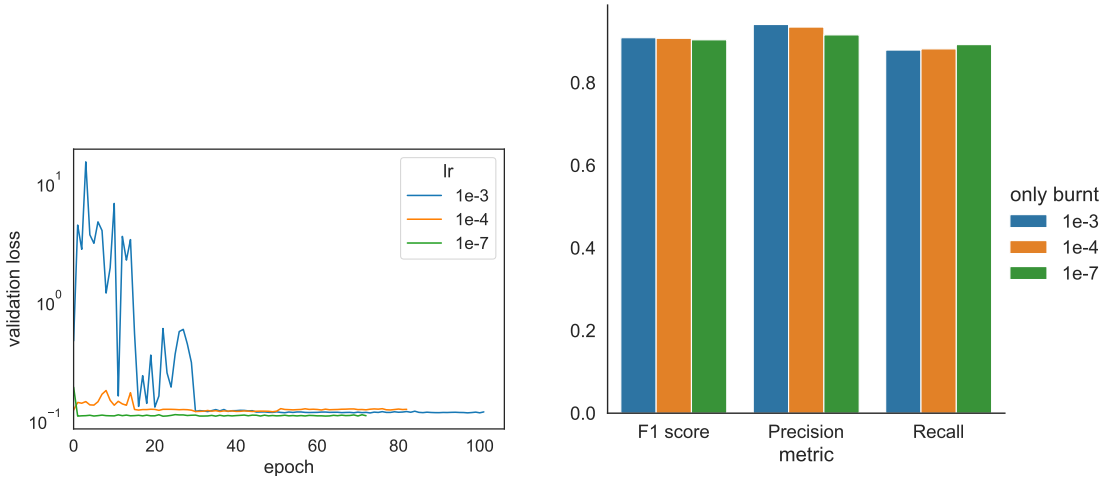
4.10 Magnifier net

For this experiments, we used the architecture presented in section 3.3.4 composed by two SegFormer-B3 with DiceFocal loss to exploit the weights obtained from previous experiments. To decide a starting learning rate we tested some values: 10^{-7} (the ones reached by the weights), 10^{-3} (the default one used in other experiments) and 10^{-4} (a mid value between the two). From Figure 4.14, it is possible to see the validation loss is initially more unstable with higher learning rates, but the test F1 score is high because of the precision. For this reason, we choose to use 10^{-3} as starting learning rate.

During the training, the initial instability of the validation loss can be seen in some of the folds, while some others as *lime*, *grey* and *pink* do not show this trend (Figure 4.15).

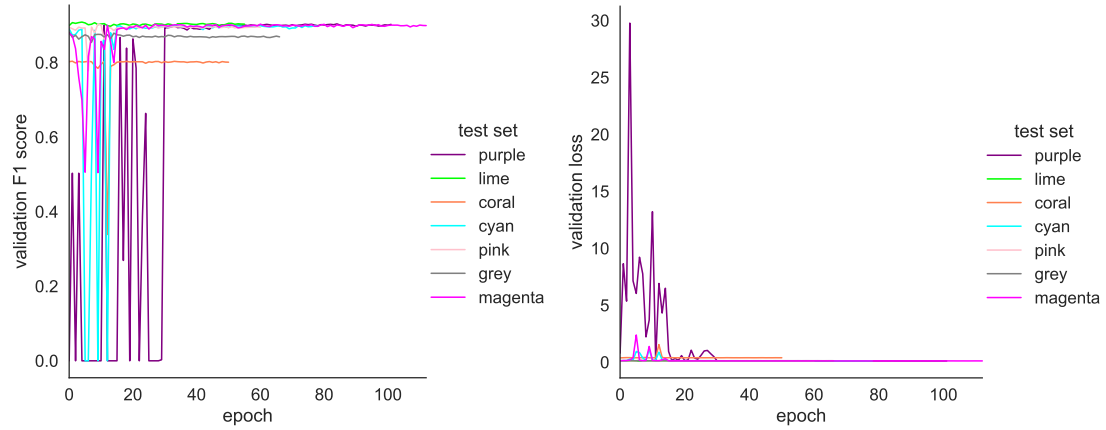
In Figure 4.10 the comparison with the simple SegFormer-B3 with DiceFocal

highlights better performance achieved by *Magnifier*. The improvement is too low. It is probably not really worth doubling the number of parameters, but more investigations are needed to understand if it is possible to achieve greater improvement with other settings.



(a) Validation loss **(b)** Test metrics

Figure 4.14: Magnifier test metrics and validation loss grouped by learning rate on test set *purple*.



(a) Validation F1 score **(b)** Validation loss

Figure 4.15: Magnifier net training

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Crop&Recompose MiT-B3	**0.909**	0.791	0.778	0.694	**0.897**	0.917	0.895	0.840	0.086
	Dice MiT-B3	0.899	0.790	0.762	0.712	0.877	0.909	0.899	0.835	0.080
	DiceFocal MiT-B3	0.891	**0.805**	0.788	**0.721**	0.883	0.923	0.907	**0.845**	0.075
	Magnifier Net	0.905	0.804	**0.801**	0.717	0.883	**0.926**	**0.909**	**0.849**	0.077
Precision	Crop&Recompose MiT-B3	0.881	0.806	0.852	0.657	0.854	**0.903**	**0.948**	0.843	0.094
	Dice MiT-B3	0.898	0.828	0.859	0.655	0.866	0.898	0.897	0.843	0.087
	DiceFocal MiT-B3	0.901	0.823	**0.876**	**0.693**	0.871	0.893	0.922	0.854	0.078
	Magnifier Net	**0.902**	**0.836**	0.847	0.688	**0.874**	0.898	0.941	**0.855**	0.082
Recall	Crop&Recompose MiT-B3	**0.938**	0.777	0.717	0.736	**0.944**	0.932	0.847	0.842	0.099
	Dice MiT-B3	0.901	0.755	0.685	**0.779**	0.888	0.920	**0.902**	0.833	0.092
	DiceFocal MiT-B3	0.880	**0.787**	0.716	0.751	0.894	0.955	0.892	0.839	0.088
	Magnifier Net	0.908	0.775	**0.760**	0.749	0.892	**0.956**	0.879	**0.845**	0.083

Table 4.10: Test metrics comparison for *Magnifier Net*

4.11 Experiments summary

Now we compare the most interesting results we got. From the metrics summary in Table 4.11, we can see:

1. SegFormer grants a better F1 score in every version with different losses (at least +2% than U-Net), increasing the mean precision (at least +10% than U-Net) and decreasing mean recall (at most −10% than U-Net);

2. The use of compound losses (DiceFocal and Asymmetric Unified Focal) grant better and more stable results than using a single loss (Dice loss);

3. All models have difficulties in the same folds. This makes thinking there are some intrinsic properties that are affecting the final results.

		coral	cyan	grey	lime	magenta	pink	purple	mean	std
F1 score	Dice MiT-B3	0.899	0.790	0.762	0.712	0.877	0.909	0.899	0.835	0.080
	DiceFocal MiT-B0	0.898	0.787	0.755	0.671	0.884	**0.927**	0.908	0.833	0.096
	DiceFocal MiT-B3	0.891	**0.805**	0.788	**0.721**	0.883	0.923	0.907	0.845	0.075
	UnifiedFocal MiT-B3	**0.902**	**0.805**	**0.857**	0.688	**0.888**	0.915	**0.918**	**0.853**	0.083
Precision	Dice MiT-B3	0.898	**0.828**	0.859	0.655	0.866	0.898	0.897	0.843	0.087
	DiceFocal MiT-B0	0.876	0.810	0.864	0.613	**0.879**	**0.926**	**0.935**	0.843	0.110
	DiceFocal MiT-B3	**0.901**	0.823	**0.876**	**0.693**	0.871	0.893	0.922	**0.854**	0.078
	UnifiedFocal MiT-B3	0.897	**0.828**	0.842	0.574	0.855	0.866	0.920	0.826	0.115
Recall	Dice MiT-B3	0.901	0.755	0.685	0.779	0.888	0.920	0.902	0.833	0.092
	DiceFocal MiT-B0	**0.920**	0.766	0.670	0.742	0.888	0.929	0.882	0.828	0.101
	DiceFocal MiT-B3	0.880	**0.787**	0.716	0.751	0.894	0.955	0.892	0.839	0.088
	UnifiedFocal MiT-B3	0.908	0.783	**0.872**	**0.857**	**0.925**	**0.970**	**0.915**	**0.890**	0.060

Table 4.11: Test metrics summary of the most indicative models and losses

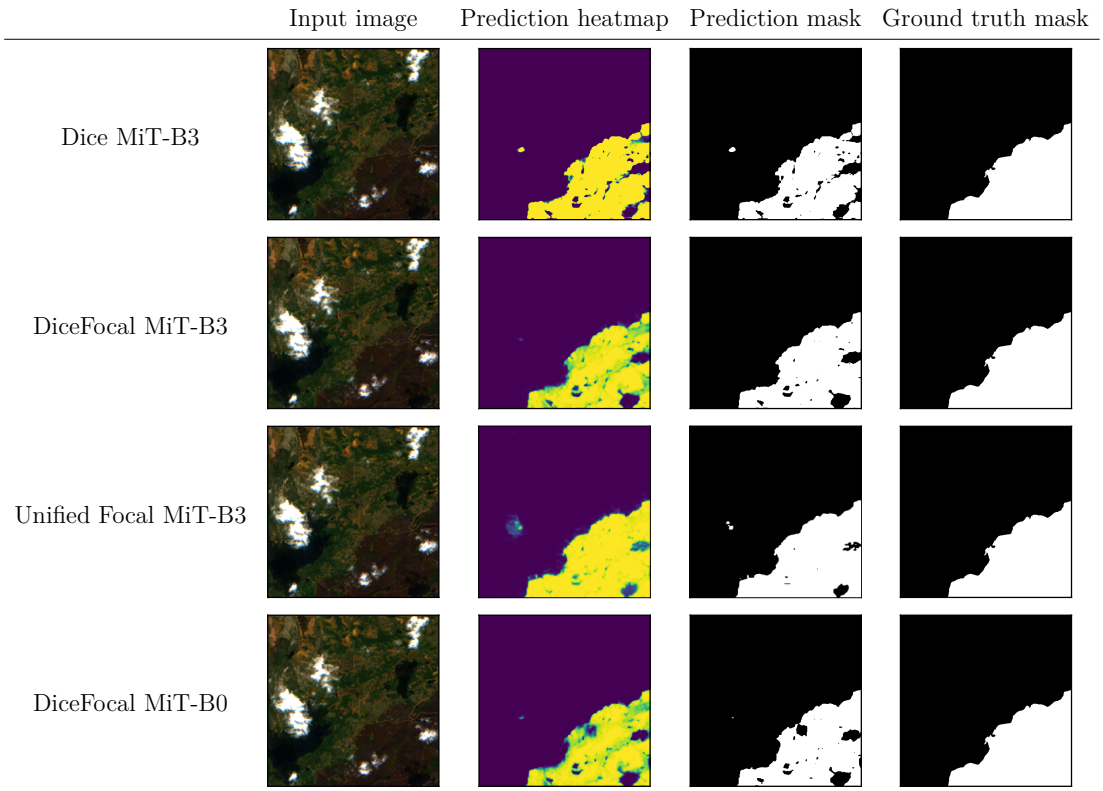


Figure 4.16: Same image of grey fold with different models and losses

4.12 Interpretability

We used the *captum* library [85] to understand how each channel contributes to the final prediction.

We applied the Integrated Gradients method [86], which consists in assigning an importance score to each input feature by approximating the integral of gradients of the model's output with respect to the inputs along the path (straight line) from given references to inputs. The approximation was made using the Gauss-Legendre quadrature rule in 50 steps.

At first, we calculate the F1 scores of the test dataset using the pre-trained model and then we split the dataset into three equal parts according to the score: the ones with the best scores (we call it *best set*), the ones with the worst scores (we call it *worst set*) and the remaining. We applied the algorithm to *worst* and *best* sets, to understand which are the channels that affect negatively the predictions and which are the ones that grant good results.

We tested the SegFormer-B0 and SegFormer-B3 with DiceFocal loss on a well-performing dataset (*purple*) and on a bad performing one (*grey*). In Figures 4.17

and 4.18 it is possible to see the importance score for each band of the two datasets.

In *purple* the most informative channels are 11 and 12 for both models, while for *grey* this is true only in B0 version. The channels 2, 3, 4 acquire more importance in *worst* sets.

In B3 version it is possible to note the worst scores are obtained when each band increase its contribution to the final result.

In B0 version the band 12 increased importance and the decrease of band 11 seem a constant in getting bad results. In *grey* set, it is possible to see how bands 8, 8A and 9 greatly increase their impact in the *worst* set.

We can conclude some bands have more impact on the results as expected, in particular the ones related to vegetation (8, 8A, 9, 11, 12), while RGB (2, 3, 4) seems linked to providing misleading information.

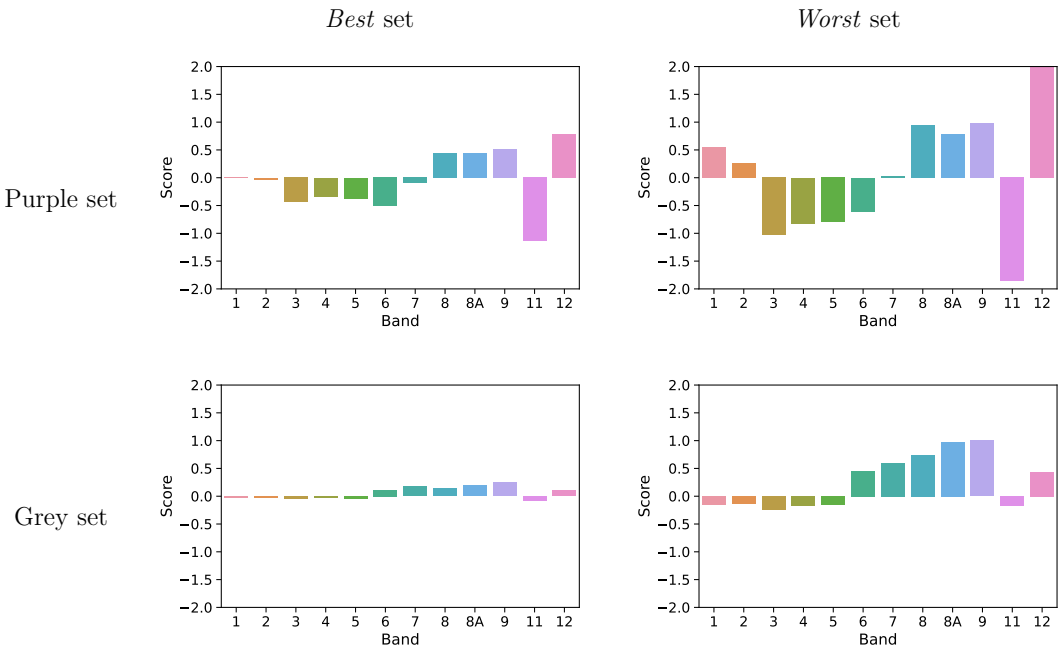


Figure 4.17: DiceFocal Mit-B3 mean importance scores for each band

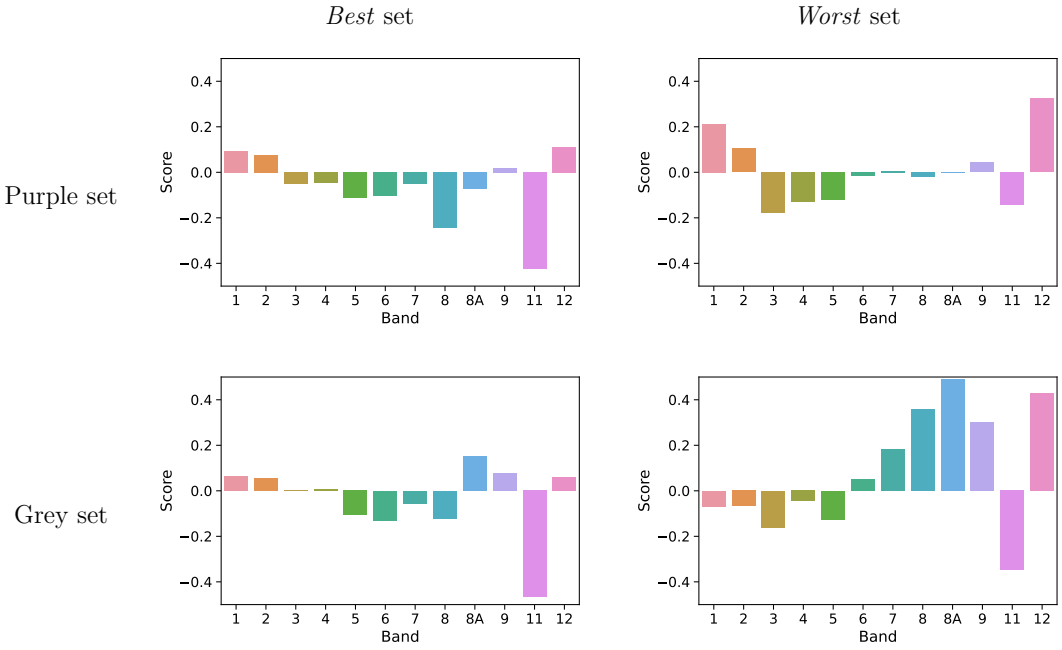


Figure 4.18: DiceFocal Mit-B0 mean importance scores for each band

Chapter 5

Conclusion

In this thesis we investigated how a novel vision transformer architecture, SegFormer, can be a good substitute for known CNN-based architectures in the context of remote sensing and burned area delineation, providing not only better results, but also better performance in terms of computational cost and number of parameters. Furthermore, we analyzed the effectiveness of several loss functions and different versions of the SegFormer architecture, achieving superior results in terms of precision and F1 score with respect to state-of-the-art models. We started a simple investigation on the relation between the results and the input channels showing how they affect the final predictions.

In future works, we plan to apply self-supervised learning and multi-modal transformers to the combinations of different satellite acquisitions, such as Sentinel-1 and Sentinel-2. It can be also of great interest to furtherly investigate the regression problem by trying to predict the amount of damage for each pixel. Concerning the performance gap observed through the various folds, we plan to do a more accurate investigation of the motivations behind this. *Magnifier Net* seems to grant better results, but more settings need to be tested to find out the best one.

Bibliography

[1] W. Matt Jolly, Mark A. Cochrane, Patrick H. Freeborn, Zachary A. Holden, Timothy J. Brown, Grant J. Williamson, and David M. J. S. Bowman. «Climate-induced variations in global wildfire danger from 1979 to 2013». In: *Nature Communications* 6.1 (July 2015), p. 7537. ISSN: 2041-1723 (cit. on p. 1).

[2] Michael Goss, Daniel L. Swain, John T. Abatzoglou, Ali Sarhadi, Crystal A. Kolden, A. Park Williams, and Noah S. Diffenbaugh. «Climate change is increasing the likelihood of extreme autumn wildfire conditions across California». In: *Environmental Research Letters* 15.9 (Aug. 2020), p. 094016. ISSN: 1748-9326 (cit. on p. 1).

[3] John T. Abatzoglou and A. Park Williams. «Impact of anthropogenic climate change on wildfire across western US forests». In: *Proceedings of the National Academy of Sciences* 113.42 (2016), pp. 11770–11775 (cit. on p. 1).

[4] Ruth A Engel, Miriam E Marlier, and Dennis P Lettenmaier. «On the causes of the summer 2015 Eastern Washington wildfires». In: *Environmental Research Communications* 1.1 (Feb. 2019), p. 011009. DOI: 10.1088/2515-7620/ab082e. URL: https://doi.org/10.1088/2515-7620/ab082e (cit. on p. 1).

[5] Jon E. Keeley and Alexandra D. Syphard. «Historical patterns of wildfire ignition sources in California ecosystems». In: *International Journal of Wildland Fire* 27.12 (2018), pp. 781–799 (cit. on p. 1).

[6] Joseph W. Mitchell. «Power line failures and catastrophic wildfires under extreme weather conditions». In: *Engineering Failure Analysis* 35 (2013). Special issue on ICEFA V- Part 1, pp. 726–735. ISSN: 1350-6307 (cit. on p. 1).

[7] Songyi Dian et al. «Integrating Wildfires Propagation Prediction Into Early Warning of Electrical Transmission Line Outages». In: *IEEE Access* 7 (2019), pp. 27586–27603 (cit. on p. 1).

[8] M. J. Gunsch et al. «Ubiquitous influence of wildfire emissions and secondary organic aerosol on summertime atmospheric aerosol in the forested Great Lakes region». In: *Atmospheric Chemistry and Physics* 18.5 (2018), pp. 3701–3715 (cit. on p. 1).

[9] Munich RE. *Record hurricane season and major wildfires – The natural disaster figures for 2020*. 2021 (cit. on p. 1).

[10] Munich RE. *Wildfires. Clear indicators that climate change is changing the risks*. 2019 (cit. on p. 1).

[11] Jim Schieck and Samantha J Song. «Changes in bird communities throughout succession following fire and harvest in boreal forests of western North America: literature review and meta-analyses». In: *Canadian Journal of Forest Research* 36.5 (2006), pp. 1299–1318. DOI: 10.1139/x06-017 (cit. on p. 1).

[12] David Pilliod, R. Bury, Erin Hyde, Christopher Pearl, and Paul Corn. «Fire and amphibians in North America». In: *Forest Ecology and Management* 178 (June 2003), pp. 163–181. DOI: 10.1016/S0378-1127(03)00060-4 (cit. on p. 1).

[13] JASON T. FISHER and LISA WILKINSON. «The response of mammals to forest fire and timber harvest in the North American boreal forest». In: *Mammal Review* 35.1 (2005), pp. 51–81. DOI: https://doi.org/10.1111/j.1365-2907.2005.00053.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2907.2005.00053.x (cit. on p. 1).

[14] Matthew C. Hansen and Thomas R. Loveland. «A review of large area monitoring of land cover change using Landsat data». In: *Remote Sensing of Environment* 122 (2012). Landsat Legacy Special Issue, pp. 66–74. ISSN: 0034-4257 (cit. on p. 2).

[15] Anju Asokan and J. Anitha. «Change detection techniques for remote sensing applications: a survey». In: *Earth Science Informatics* 12.2 (Mar. 2019), pp. 143–160 (cit. on p. 2).

[16] Haojie Ma, Yalan Liu, Yuhuan Ren, and Jingxian Yu. «Detection of Collapsed Buildings in Post-Earthquake Remote Sensing Images Based on the Improved YOLOv3». In: *Remote Sensing* 12.1 (2020). ISSN: 2072-4292 (cit. on p. 2).

[17] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. «SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers». In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 12077–12090 (cit. on pp. 2, 5, 6, 12, 13, 19, 23, 35).

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597 (cit. on pp. 2, 9, 10, 19).

[19] Olga Russakovsky et al. «ImageNet Large Scale Visual Recognition Challenge». In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cit. on pp. 5–7).

[20] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. DOI: 10.48550/ARXIV.1704.04861. URL: https://arxiv.org/abs/1704.04861 (cit. on pp. 5, 6).

[21] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2018. DOI: 10.48550/ARXIV.1812.08008. URL: https://arxiv.org/abs/1812.08008 (cit. on pp. 5, 6).

[22] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. *Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training*. 2021. DOI: 10.48550/ARXIV.2111.01118. URL: https://arxiv.org/abs/2111.01118 (cit. on pp. 5, 6).

[23] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. *Uformer: A General U-Shaped Transformer for Image Restoration*. 2021. DOI: 10.48550/ARXIV.2106.03106. URL: https://arxiv.org/abs/2106.03106 (cit. on pp. 5, 6).

[24] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. *MoViNets: Mobile Video Networks for Efficient Video Recognition*. 2021. DOI: 10.48550/ARXIV.2103.11511. URL: https://arxiv.org/abs/2103.11511 (cit. on pp. 5, 6).

[25] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. *PolyTransform: Deep Polygon Transformer for Instance Segmentation*. 2019. DOI: 10.48550/ARXIV.1912.02801. URL: https://arxiv.org/abs/1912.02801 (cit. on pp. 5, 6).

[26] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo, and Tong Lu. *Panoptic SegFormer: Delving Deeper into Panoptic Segmentation with Transformers*. 2021. DOI: 10.48550/ARXIV.2109.03814. URL: https://arxiv.org/abs/2109.03814 (cit. on pp. 5, 6, 12).

[27] C. Van Der Malsburg. «Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms». In: Springer Berlin Heidelberg, 1986 (cit. on p. 6).

[28] Seymour A. Papert. *The Summer Vision Project*. URL: http://hdl.handle.net/1721.1/6125 (cit. on p. 6).

[29] Richard Szeliski. *Computer Vision: Algorithms and Applications*. 2010 (cit. on p. 6).

[30] Herbert F Schantz. *The history of OCR, optical character recognition*. 1982 (cit. on p. 6).

[31] David Marr. *Vision : a computational investigation into the human representation and processing of visual information*. 1982 (cit. on p. 6).

[32] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. «Neocognitron: A neural network model for a mechanism of visual pattern recognition». In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.5 (1983), pp. 826–834. DOI: 10.1109/TSMC.1983.6313076 (cit. on p. 6).

[33] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. «Backpropagation Applied to Handwritten Zip Code Recognition». In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541 (cit. on p. 7).

[34] P. Viola and M. Jones. «Rapid object detection using a boosted cascade of simple features». In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517 (cit. on p. 7).

[35] Jun Ma. *Segmentation Loss Odyssey*. 2020. DOI: 10.48550/ARXIV.2005.13449. URL: https://arxiv.org/abs/2005.13449 (cit. on pp. 7, 8).

[36] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. «Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations». In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 240–248 (cit. on pp. 8, 24).

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. «Focal Loss for Dense Object Detection». In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2999–3007 (cit. on pp. 8, 26).

[38] Wentao Zhu, Yufang Huang, Liang Zeng, Xuming Chen, Yong Liu, Zhen Qian, Nan Du, and Xiaohui Xie. «AnatomyNet: Deep Learning for Fast and Fully Automated Whole-volume Segmentation of Head and Neck Anatomy». In: *Medical Physics* 46 (Nov. 2018) (cit. on p. 9).

[39] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. *Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation*. 2021. DOI: 10.48550/ARXIV.2102.04525. URL: https://arxiv.org/abs/2102.04525 (cit. on p. 9).

[40] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. «On Face Segmentation, Face Swapping, and Face Perception». In: *2018 13th IEEE International Conference on Automatic Face; Gesture Recognition (FG 2018)*. Xi'an, China: IEEE Press, 2018, pp. 98–105 (cit. on p. 9).

[41] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. «A Comparative Study of Real-Time Semantic Segmentation for Autonomous Driving». In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 700–70010 (cit. on p. 9).

[42] Zhiqiong Wang, Mo Li, Huaxia Wang, Hanyu Jiang, Yudong Yao, Hao Zhang, and Junchang Xin. «Breast Cancer Detection Using Extreme Learning Machine Based on Feature Fusion With CNN Deep Features». In: *IEEE Access* 7 (2019), pp. 105146–105158 (cit. on p. 9).

[43] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. «Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs». In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 2229–2235 (cit. on p. 9).

[44] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. «UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation». In: *ICASSP 2020*. 2020, pp. 1055–1059 (cit. on p. 9).

[45] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. *ParseNet: Looking Wider to See Better*. 2015. DOI: 10.48550/ARXIV.1506.04579. URL: https://arxiv.org/abs/1506.04579 (cit. on p. 9).

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. *Pyramid Scene Parsing Network*. 2016. DOI: 10.48550/ARXIV.1612.01105. URL: https://arxiv.org/abs/1612.01105 (cit. on pp. 9, 10).

[47] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. «DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848 (cit. on pp. 9, 11).

[48] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. «Learning Deconvolution Network for Semantic Segmentation». In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1520–1528 (cit. on p. 10).

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. «Attention is All You Need». In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964 (cit. on p. 12).

[50] Ozan Oktay et al. *Attention U-Net: Learning Where to Look for the Pancreas*. 2018. DOI: 10.48550/ARXIV.1804.03999. URL: https://arxiv.org/abs/1804.03999 (cit. on p. 12).

[51] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. *Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation*. 2020. DOI: 10.48550/ARXIV.2003.07853. URL: https://arxiv.org/abs/2003.07853 (cit. on p. 12).

[52] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2019. DOI: 10.48550/ARXIV.1910.03771 (cit. on pp. 12, 23).

[53] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186 (cit. on p. 12).

[54] Tom Brown et al. «Language Models are Few-Shot Learners». In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901 (cit. on p. 12).

[55] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. DOI: 10.48550/ARXIV.2010.11929 (cit. on pp. 12, 13).

[56] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. *Scaling Vision Transformers*. 2021. DOI: 10.48550/ARXIV.2106.04560 (cit. on p. 12).

[57] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. «Swin Transformer: Hierarchical Vision Transformer using Shifted Windows». In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9992–10002 (cit. on pp. 12, 13).

[58] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. *Colorization Transformer*. 2021. DOI: 10.48550/ARXIV.2102.04432. URL: https://arxiv.org/abs/2102.04432 (cit. on p. 12).

[59] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. *Transformer for Single Image Super-Resolution*. 2021. DOI: 10.48550/ARXIV.2108.11084 (cit. on p. 12).

[60] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. *Video Super-Resolution Transformer*. 2021. DOI: 10.48550/ARXIV.2106.06847 (cit. on p. 12).

[61] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. «Twins: Revisiting the Design of Spatial Attention in Vision Transformers». In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 9355–9366 (cit. on p. 12).

[62] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. «Training data-efficient image transformers and distillation through attention». In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR, 2021, pp. 10347–10357 (cit. on p. 12).

[63] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. «LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference». In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 12239–12249 (cit. on p. 12).

[64] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. «Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions». In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 548–558 (cit. on p. 12).

[65] GISGeography. *Spectral Signature Cheatsheet – Spectral Bands in Remote Sensing*. https://gisgeography.com/spectral-signature/. 2022 (cit. on p. 14).

[66] D.P. Roy, L. Boschetti, and S.N. Trigg. «Remote sensing of fire severity: assessing the performance of the normalized burn ratio». In: *IEEE Geoscience and Remote Sensing Letters* 3.1 (2006), pp. 112–116 (cit. on p. 14).

[67] 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». In: *Remote Sensing of Environment* 222 (2019), pp. 1–17. ISSN: 0034-4257 (cit. on p. 14).

[68] Federico Filipponi. «BAIS2: Burned Area Index for Sentinel-2». In: *Proceedings* 2.7 (2018). ISSN: 2504-3900 (cit. on p. 14).

[69] Jay D. Miller and Andrea E. Thode. «Quantifying burn severity in a heterogeneous landscape with a relative version of the delta Normalized Burn Ratio (dNBR)». In: *Remote Sensing of Environment* 109.1 (2007), pp. 66–80. ISSN: 0034-4257 (cit. on p. 14).

[70] Wu Bin, Liu Ming, Jia Dan, Li Suju, Cong Qiang, Wang Chao, Zhu Yang, Yin Huan, and Zhu Jun. «A Method of Automatically Extracting Forest Fire Burned Areas Using Gf-1 Remote Sensing Images». In: *IGARSS 2019*. 2019, pp. 9953–9955 (cit. on p. 14).

[71] Grace Puyang Emang, Yoshiya Touge, and So Kazama. «Evaluating Trees Crowns Damage for the 2017 Largest Wildfire in Japan Using Sentinel-2A NDMI». In: *IGARSS 2020*. 2020, pp. 6794–6797 (cit. on p. 14).

[72] Nobuyuki Otsu. «A threshold selection method from gray-level histograms». In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66 (cit. on p. 14).

[73] Luigi Saulino, Angelo Rita, Antonello Migliozzi, Carmine Maffei, Emilia Allevato, Antonio Pietro Garonna, and Antonio Saracino. «Detecting Burn Severity across Mediterranean Forest Types by Coupling Medium-Spatial Resolution Satellite Imagery and Field Data». In: *Remote Sensing* 12.4 (2020). ISSN: 2072-4292 (cit. on p. 15).

[74] Lisa Knopp, Marc Wieland, Michaela Rättich, and Sandro Martinis. «A Deep Learning Approach for Burned Area Segmentation with Sentinel-2 Data». In: *Remote Sensing* 12.15 (2020), p. 2422. ISSN: 2072-4292 (cit. on p. 15).

[75] Simone Monaco, Salvatore Greco, Alessandro Farasin, Luca Colomba, Daniele Apiletti, Paolo Garza, Tania Cerquitelli, and Elena Baralis. «Attention to Fires: Multi-Channel Deep Learning Models for Wildfire Severity Prediction». In: *Applied Sciences* 11.22 (2021). ISSN: 2076-3417 (cit. on pp. 15, 17).

[76] Alessandro Farasin, Luca Colomba, and Paolo Garza. «Double-Step U-Net: A Deep Learning-Based Approach for the Estimation of Wildfire Damage Severity through Sentinel-2 Satellite Data». In: *Applied Sciences* 10.12 (2020). ISSN: 2076-3417 (cit. on pp. 15, 17, 23).

[77] Seyd Teymoor Seydi, Mahdi Hasanlou, and Jocelyn Chanussot. «DSMNN-Net: A Deep Siamese Morphological Neural Network Model for Burned Area Mapping Using Multispectral Sentinel-2 and Hyperspectral PRISMA Images». In: *Remote Sensing* 13.24 (2021), p. 5138. ISSN: 2072-4292 (cit. on p. 15).

[78] Seyd Teymoor Seydi, Mahdi Hasanlou, and Jocelyn Chanussot. «A Quadratic Morphological Deep Neural Network Fusing Radar and Optical Data for the Mapping of Burned Areas». In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 4194–4216 (cit. on p. 15).

[79] Rafik Ghali, Moulay A. Akhloufi, Marwa Jmal, Wided Souidene Mseddi, and Rabah Attia. «Wildfire Segmentation Using Deep Vision Transformers». In: *Remote Sensing* 13.17 (2021), p. 3527. ISSN: 2072-4292 (cit. on p. 15).

[80] GISGeography. *Sentinel 2 Bands and Combinations*. https://gisgeography.com/sentinel-2-bands-combinations/. 2022 (cit. on p. 15).

[81] Justin Braaten, Warren Cohen, and Zhiqiang Yang. «Automated cloud and cloud shadow identification in Landsat MSS imagery for temperate ecosystems». In: *Remote Sensing of Environment* 169 (Nov. 2015), pp. 128–138 (cit. on p. 20).

[82] Lei Ding, Dong Lin, Shaofu Lin, Jing Zhang, Xiaojie Cui, Yuebin Wang, Hao Tang, and Lorenzo Bruzzone. «Looking Outside the Window: Wide-Context Transformer for the Semantic Segmentation of High-Resolution Remote Sensing Images». In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–13 (cit. on p. 20).

[83] Marco Seeland, Michael Rzanny, Nedal Alaqraa, Jana Wäldchen, and Patrick Mäder. «Plant species classification using flower images—A comparative study of local feature representations». In: *PLOS ONE* 12 (Feb. 2017), e0170629 (cit. on p. 20).

[84] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. 2019. URL: `https://github.com/Lightning-AI/lightning` (cit. on p. 23).

[85] Narine Kokhlikyan et al. *Captum: A unified and generic model interpretability library for PyTorch*. 2020. DOI: `10.48550/ARXIV.2009.07896`. URL: `https://arxiv.org/abs/2009.07896` (cit. on p. 39).

[86] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. *Axiomatic Attribution for Deep Networks*. 2017. DOI: `10.48550/ARXIV.1703.01365`. URL: `https://arxiv.org/abs/1703.01365` (cit. on p. 39).