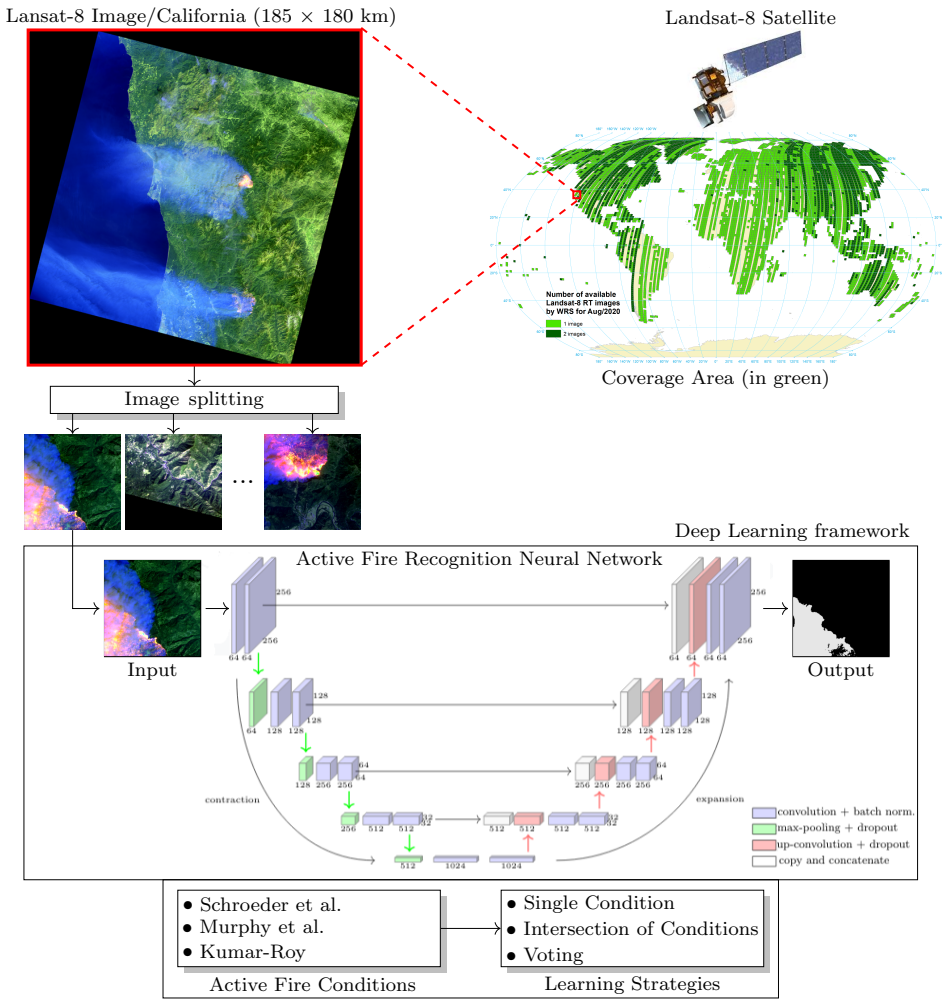
Graphical Abstract

Active Fire Detection in Landsat-8 Imagery: a Large-Scale Dataset and a Deep-Learning Study

Gabriel Henrique de Almeida Pereira, Andre Minoro Fusioka, Bogdan Tomoyuki Nassu, Rodrigo Minetto,



Highlights

Active Fire Detection in Landsat-8 Imagery: a Large-Scale Dataset and a Deep-Learning Study

Gabriel Henrique de Almeida Pereira, Andre Minoro Fusioka, Bogdan Tomoyuki Nassu, Rodrigo Minetto,

- Proposal of a large-scale dataset for active fire detection in Landsat-8 imagery
- Deep learning can approximate and improve the results from handcrafted algorithms
- Deep learning networks can approximate combinations of handcrafted algorithms

Active Fire Detection in Landsat-8 Imagery: a Large-Scale Dataset and a Deep-Learning Study

Gabriel Henrique de Almeida Pereira*, Andre Minoro Fusioka, Bogdan Tomoyuki Nassu, Rodrigo Minetto

Av. Sete de Setembro, 3165, Curitiba, Brazil

Department of Informatics - Federal University of Technology - Paraná

Abstract

Active fire detection in satellite imagery is of critical importance to the management of environmental conservation policies, supporting decision-making and law enforcement. This is a well established field, with many techniques being proposed over the years, usually based on pixel or region-level comparisons involving sensor-specific thresholds and neighborhood statistics. In this paper, we address the problem of active fire detection using deep learning techniques. In recent years, deep learning techniques have been enjoying an enormous success in many fields, but their use for active fire detection is relatively new, with open questions and demand for datasets and architectures for evaluation. This paper addresses these issues by introducing a new large-scale dataset for active fire detection, with over 150,000 image patches (more than 200 GB of data) extracted from Landsat-8 images captured around the world in August and September 2020, containing wildfires in several locations. The dataset was split in two parts, and contains 10-band spectral

*Corresponding author

Email addresses: pereira.gha@hotmail.com (Gabriel Henrique de Almeida Pereira), amfusioka@gmail.com (Andre Minoro Fusioka), btnassu@utfpr.edu.br (Bogdan Tomoyuki Nassu), rminetto@utfpr.edu.br (Rodrigo Minetto)

images with associated outputs, produced by three well known handcrafted algorithms for active fire detection in the first part, and manually annotated masks in the second part. We also present a study on how different convolutional neural network architectures can be used to approximate these handcrafted algorithms, and how models trained on automatically segmented patches can be combined to achieve better performance than the original algorithms — with the best combination having 87.2% precision and 92.4% recall on our manually annotated dataset. The proposed dataset, source codes and trained models are available on Github[1], creating opportunities for further advances in the field.

Keywords: active fire detection, active fire segmentation, active fire dataset, convolutional neural network, landsat-8 imagery.

1. Introduction

The use of satellite imagery is of critical importance to the management of environmental conservation policies. Active fire detection is a field of research that extracts relevant knowledge from such images to support decision-making (Cardil et al. (2019)). For example, fires can be employed as a means to clear out an area after trees are cut down, giving place to pastures and agricultural land, so the presence of active fire can be directly related to deforestation (Portillo-Quintero et al. (2013)) in many important biomes around the world. Furthermore, there is a direct relation between biomass burning and changes to the climate and atmospheric chemistry (Chuvieco et al.

[1] `https://github.com/pereira-gha/activefire`

(2019)). Active fire detection in satellite imagery is also relevant for other monitoring tasks, such as damage assessment, prevention and prediction [2].

Research on the field of active fire detection focuses on providing appropriate methods for determining if a given pixel (corresponding to a georeferenced area) in a multi-spectral image corresponds to an active fire. As the satellites orbit in different altitudes and their systems are equipped with different sensors, with different wavelengths, usually such methods must be designed specifically for each system. For example, images from the Landsat-8 satellite — the one considered in this work — are usually processed by series of conditions such as those proposed by Schroeder et al. (2016), Murphy et al. (2016) or Kumar and Roy (2018). These conditions are especially tuned for the Operational Land Imager (OLI) sensor on board Landsat-8, as detailed in Section 2. Other important sensors and satellites for active fire recognition are: the MODIS sensor, with 250 m to 1 km of spatial resolution per pixel, that equips the NASA Terra and Aqua satellites (orbiting 705 km height, 1-2 days of revisit); the AVHRR sensor, with 1 km of spatial resolution per pixel, on board NOAA-18 (833 km height), NOAA-19 (870 km height) and METOP-B (827 km height) satellites, the three with a daily revisit time; the Visible Infrared Imaging Radiometer Suite (VIIRS) sensor, with 375 m of spatial resolution and on board the joint NASA/NOAA Suomi National Polar-orbiting Partnership (Suomi NPP) and NOAA-20 satellites (824 km

[2] https://effis.jrc.ec.europa.eu/about-effis/technical-background/rapid-damage-assessment

height), both with daily revisit time; the ABI sensor, that equips the geostationary satellite GOES-16 ($\approx$ 36,000 km height), that is able to take images each 10 minutes with spatial resolution from 500 meters to 2 km. In addition, the Sentinel-2 constellation have also been used for research on this field as they produce images similar to Landsat, with spatial resolution between 10 and 60 meters, with 5 days of revisit.

Studies on automatic methods for detecting active fire regions date back to the 1970s, when the most suitable spectral intervals for detecting forest fires were analyzed (Kondratyev et al. (1972)). Matson and Holben (1987) argued that remote sensing was the only viable alternative to monitor active fire in isolated regions. In their study, they used the AVHRR sensor to detect fire activity in the Amazon area. Other active fire detection algorithms have also been developed for AVHRR (Flannigan and Haar (1986); Lee and Tag (1990)), some of them being used as a basis for the development of algorithms for other satellite sensors. Ji and Stocker (2002) and Giglio et al. (2003) developed algorithms that include contextual analysis for active fire detection for the Visible and Infrared Scanner (VIRS) sensor onboard the Tropical Rainfall Measuring Mission (TRMM) satellite. Kaufman et al. (1998) proposed an approach for active fire detection using the MODIS sensor, which was later improved by Giglio et al. (2003). This method is still used today due to its relevance, and because the MODIS sensor is still active, and has an almost daily global coverage, being able to detect fires with size 100-300 m^2 (Maier et al. (2013)). Furthermore, the MODIS detection

algorithm was improved in many ways (Morisette et al. (2005); Giglio et al. (2016)) and served a reference for developing equations for other satellites. As an example, in 2014, Schroeder et al. (2014) proposed an algorithm for active fire detection for the VIIRS sensor based on the MODIS algorithm. The active fire detection algorithms mentioned above, in general, are based on comparisons to fixed threshold in certain bands and statistics from their surrounding region.

In recent years, deep-learning methods (Lecun et al. (2015)) have been promoting major advances in artificial intelligence, inspiring the design of novel architectures and strategies for a wide range of applications in remote sensing, including several related to active fire recognition. That includes architectures for tasks such as: distinguish smoke from similar patterns such as clouds, dust, haze, land and seaside by using images from the Aqua and Terra satellites (Ba et al. (2019)); super-resolution enhancement for the specific context of wildfire on images from the Sentinel-2 satellite; use of temporal information from a same location, before and after wildfires, so as to detect burned/unburned areas in images of Sentinel-1 or VIIRS data (Ban et al. (2020), Pinto et al. (2020)); tackling the imbalanced classification problem, usually present in active fire segmentation, that can negatively impact CNN performance (Langford et al. (2018)); and the use of generative adversarial networks (GAN) to synthesize missing or corrupted multispectral optical images with multitemporal data from satellites (Bermudez et al. (2019)). In summary, although this is a relatively new field, with no specific architec-

ture finding widespread application on remote sensing applications, most of these works are based on a particular type of deep learning network, known as Convolutional Neural Network (CNN) (Lecun et al. (2015)), that tries to discover intricate patterns in massive data by using a sequence of processing layers that allows to learn representations of data with multiple levels of abstraction.

The use of deep learning techniques for active fire recognition is a relatively new field, which still lacks large-scale datasets and architectures for evaluation. This paper brings several contributions to the field. The first contribution is that we introduce a new public dataset for active fire recognition, built from 8,194 Landsat-8 images around the world, covering the period of August 2020, containing large wildfire events in many areas such as the Amazon region, Africa, Australia, United States, among others. The dataset contains 146,214 image patches (around 210 GB uncompressed), including 10-band spectral images and associated outputs produced by three well established, handcrafted algorithms for active fire detection (Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018)). The second contribution is a secondary dataset, also public, containing 9,044 image patches (around 12 GB uncompressed) extracted from 13 Landsat-8 images captured in September 2020, along with manually annotated fire pixels, which can be employed for assessing the quality of automatic segmentation compared to a human specialist. As a third contribution, we present a study on how convolutional neural networks can be used to approximate the handcrafted

fire detection algorithms mentioned above, including the possibility of saving bandwidth resources by reducing the number of analyzed spectral bands, as well as means of combining multiple fire detection outputs to produce more robust detection results. As a fourth contribution, we highlight the comparison between the fire detection algorithms mentioned above and the CNNs when tested against manual annotations from a human specialist. Finally, as a fifth contribution, all the source code of the deep learning and handcrafted algorithms employed for producing this paper are available for free access on github, all of them coded in Python language, which is a step forward for other researchers in the field towards a benchmark, opening opportunities for the study of other Landsat-8 time intervals, or the use of different satellites — this is a first effort in this direction, as the algorithms we analyzed here, to the best of our knowledge, do not have open source versions available.

The rest of this paper is organized as follows. In section 2, we describe the proposed datasets, including the procedure for generating them. In section 3, we describe the CNN architectures for active fire detection and the performed experiments. Results are presented and discussed in section 4. Finally, section 5 presents conclusions and points to future work.

2. Materials

The Landsat program, sponsored by NASA/USGS, provides since 1972 continuous acquisition of high-quality satellite imagery of the Earth, becom-

ing a key instrument for the continuous monitoring of the environment. As described by Roy et al. (2014), the Landsat-8 satellite, which we consider in this research, orbits the Earth at an altitude of 705 km, with data segmented into scenes with 185×180 km, defined according to the second World-wide Reference System (WRS) in a 16-day revisit period. This satellite uses an Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) sensors to acquire eleven channels of multi-spectral data $\{c_1, \ldots, c_{11}\}$. The images are encoded in the TIFF format, with resolution of $\approx 7,600 \times 7,600$ pixels, with 16 bits per pixel per channel. Each multispectral pixel corresponds to 30 meters of spatial resolution (ground).

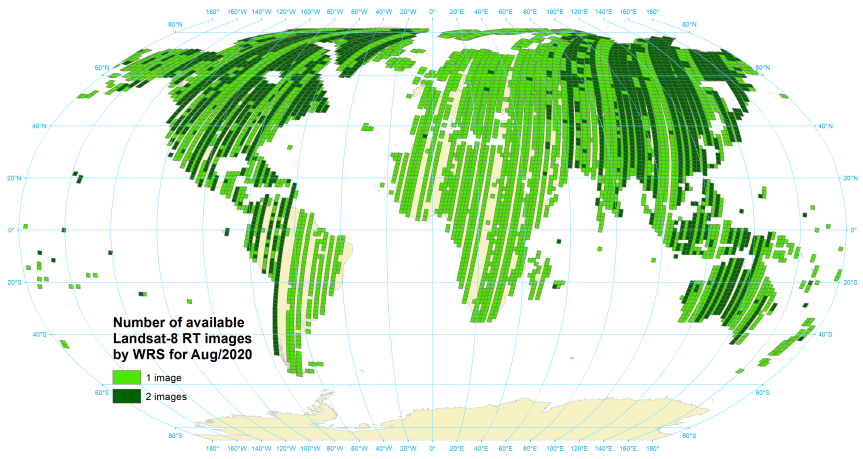


Figure 1: Landsat-8 regions used in our dataset: light green regions have one image for August 2020 and dark green regions have two images. Some regions did not have any real time images available, and are shown in yellow, along with Antarctica, which was not considered in this work. Each rectangle corresponds to an Landsat-8 WRS image scene with $\approx 7,600 \times 7,600$ pixels covering a 185×180 km area.

We processed all the Landsat-8 real time images available for August 2020 around the globe, excluding only the Antarctic continent, to a total of 8,194 images and 1.6 TB of data. Figure 1 shows scene locations and the number of real time images available from WRS.

The choice of this particular time slice was not random. Although the northern and southern hemispheres have different climatic behavior along the months — especially regarding the seasons, and the occurrence of rain, snow or droughts — there is a peak in fire activity around the same time in both hemispheres. Ferreira et al. (2020) present a study showing the duration of fire seasons, and how they are spread along the months and across the globe. They show that August and September are the most critical months, with fire seasons occurring in all continents, except Antarctica (but including the Arctic region). In 2020 this situation was not different, with several wildfire events occurring at the same time in different locations: in mid-August there were wildfires in Australia, Sibera, in the Parana Delta in Argentina, in several regions of the United States (Colorado, California, Oregon, Utah and Washington), and in the Amazon and Pantanal regions in Brazil.

Unfortunately, it is unfeasible to manually annotate this massive amount of data in order to create masks with potential active fire pixels. However, such a large amount of data is needed for properly training deep convolutional networks. Therefore, in order to automatically generate segmentation masks, we relied on three set of conditions, well established in the field, designed by Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018).

Figure 2 shows the regions analyzed by these authors when developing and testing their sets of conditions — note that our dataset has a more thorough coverage of the globe, as shown in Figure 1.

In the following sections, we detail the sets of conditions for active fire detection proposed by Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018). For the sake of simplicity, let ρ_i be in the rest of this section the reflectance of channel c_i. The equation to convert between Landsat-8 channels to reflectance are detailed by the USGS [3]. For the Murphy et al. (2016) and Kumar and Roy (2018) sets of conditions, reflectances are corrected for the solar zenith angle; while for the Schroeder et al. (2016) conditions, this correction is not performed. Moreover, it is worth noting that Schroeder et al. (2016) perform a multi-temporal analysis to avoid false alarms over persistent heat sources. That approach was adapted and used by Kumar and Roy (2018), and could also bring beneficial results to our work. However, it was not include here since our focus is on active fire recognition for single images. After describing these conditions, we detail the procedure for extracting the segmentation masks, and a secondary dataset, containing manually annotated images.

2.1. Schroeder et al. conditions

Schroeder et al. (2016) proposed a set of conditions that uses seven

[3] https://www.usgs.gov/land-resources/nli/landsat/using-usgs-landsat-level-1-data-product

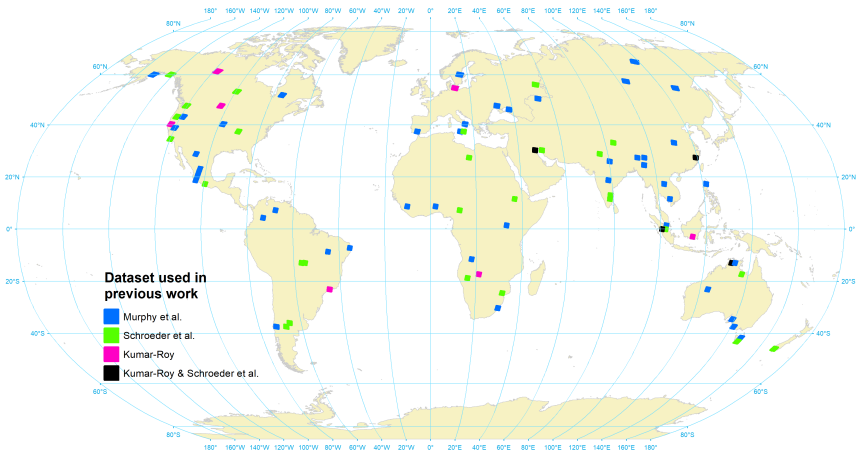


Figure 2: Locations covered in the studies from Murphy et al. (2016), Schroeder et al. (2016) and Kumar and Roy (2018), which observed, respectively, 45 regions, 30 regions and 11 distinct regions. Multiple temporal views were used for some of these regions.

Landsat-8 channels, $c_1 - c_7$, to identify active fire pixels. If all conditions are satisfied the pixel is classified as true (fire), otherwise as false (non-fire).

The first set of conditions identifies unambiguous cases:

$$((R_{75} > 2.5) \text{ and } (\rho_7 - \rho_5 > 0.3) \text{ and } (\rho_7 > 0.5)) \text{ or}$$

$$((\rho_6 > 0.8) \text{ and } (\rho_1 < 0.2) \text{ and } (\rho_5 > 0.4 \text{ or } \rho_7 < 0.1))$$

where R_{ij} is the ratio between the reflectance in channels i and j (ρ_i/ρ_j).

Then, by using a neighborhood of 61×61 pixels, centered at each fire pixel candidate, the above conditions are relaxed so as to consider the region

context, namely

$$(R_{75} > 1.8) \text{ and } (\rho_7 - \rho_5 > 0.17) \text{ and } (R_{76} > 1.6) \text{ and}$$
$$(R_{75} > \mu_{R_{75}} + \max(3\sigma_{R_{75}}, 0.8)) \text{ and } (\rho_7 > \mu_{\rho_7} + \max(3\sigma_{\rho_7}, 0.08))$$

where μ and σ denote the mean and standard deviation of the pixels in a 61×61 window pixel neighborhood. These means and standard deviations are computed excluding fire and water pixels, the latter being classified by:

$$(\rho_4 > \rho_5) \text{ and } (\rho_5 > \rho_6) \text{ and } (\rho_6 > \rho_7) \text{ and } (\rho_1 - \rho_7 < 0.2)$$
$$\text{and } ((\rho_3 > \rho_2) \text{ or } ((\rho_1 > \rho_2) \text{ and } (\rho_2 > \rho_3) \text{ and } (\rho_3 > \rho_4)))$$

For further details about the above conditions refer to the work of Schroeder et al. (2016).

2.2. Murphy et al. conditions

Murphy et al. (2016) proposed a set of conditions that is non-contextual, in the sense that it does not rely on statistics computed from each pixel's neighborhood. It is based on channels c_5, c_6 and c_7.

Unambiguous active fire pixels are first identified by:

$$(R_{76} \geq 1.4) \text{ and } (R_{75} \geq 1.4) \text{ and } (\rho_7 \geq 0.15)$$

where R_{ij} is the ratio between the reflectance in channels i and j. Potential

fires that are in the immediate neighborhood of an unambiguous fire (in a 3×3 window), are also classified as fires. Potential fires are those that satisfy:

$$((R_{65} \geq 2) \text{ and } (\rho_6 \geq 0.5)) \text{ or } ((\rho_7 \text{ is saturated}) \text{ or } (\rho_6 \text{ is saturated}))$$

Criteria for identifying saturated bands are defined by the USGS [4].

2.3. Kumar-Roy conditions

The conditions proposed by Kumar and Roy (2018) are based on channels $c_2 - c_7$. This work is more recent than the others, and builds upon some of the insights behind previous work. Unambiguous fire pixels are identified as those that satisfy

$$\rho_4 \leq 0.53\ \rho_7 - 0.214$$

They also take as unambiguous fire pixels those that are in the immediate 8-pixel vicinity of a fire pixel classified by the above condition, and which satisfy the more relaxed condition

$$\rho_4 \leq 0.35\ \rho_6 - 0.044$$

[4] https://www.usgs.gov/land-resources/nli/landsat/landsat-collection-1-level-1-quality-assessment-band

Potential fire pixels are identified by

$$((\rho_4 \leq 0.53\ \rho_7 - 0.125)\ \text{ or }\ (\rho_6 \leq 1.08\ \rho_7 - 0.048))$$

and are only kept if they satisfy the contextual test

$$(R_{75} > \mu_{R_{75}} + \max(3\sigma_{R_{75}}, 0.8)) \text{ and } (\rho_7 > \mu_{\rho_7} + \max(3\sigma_{\rho_7}, 0.08))$$

where R_{75} is the ratio between the reflectance in channels 7 and 5 (ρ_7/ρ_5), and μ and σ denote the mean and standard deviation of the pixels in a neighborhood around each candidate fire pixel. This contextual test is the same used by the Schroeder et al. (2016) conditions, with two differences. The first difference is the condition used to detect water pixels

$$(\rho_2 \geq \rho_3 \geq \rho_4 \geq \rho_5)$$

The second difference is that the region size is not fixed: we test progressively larger neighborhoods (5×5, 7×7, 9×9, and so on, up to maximum of 61×61 pixels), stopping when the region contains at least 25% of pixels not classified as unambiguous or potential fires, or as water — these pixels are excluded when computing the means and standard deviations. One point that is not addressed in the original paper is how to proceed when the equations detect a large area containing potential fire pixels — in these cases,

even with a 61×61 neighborhood there are candidate fire pixels mostly surrounded by excluded pixels, and the contextual test becomes unreliable. We considered such cases as non-fires, since doing otherwise made the algorithm highly prone to false detections.

2.4. Segmentation masks

In order to create segmentation masks, we used the conditions from Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) over the original scenes with $\approx 7,600 \times 7,600$ pixels. Figure 3 shows the locations were fires have occurred in our dataset for each set of conditions.

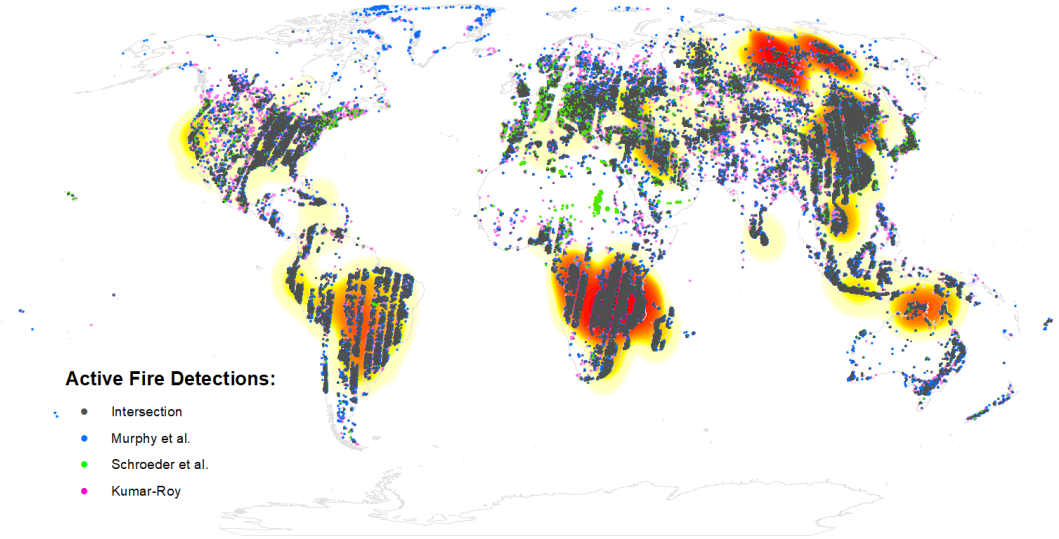


Figure 3: Fire incidence detection and heat map around the globe for August 2020 considering the outputs of Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018). Dark gray dots represent wildfire locations simultaneously detected by the three algorithms.

This dataset was then cropped into image patches with 256×256 pixels without overlap. The number of fire pixels per image patch for each set of conditions is shown in the histogram in Figure 4. It can be seen that most patches have a small number of fire pixels.

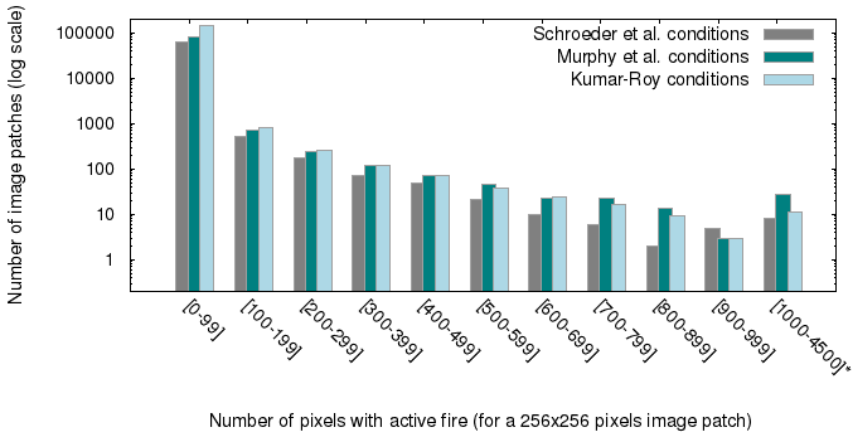


Figure 4: Distribution of image patches per fire pixel count. The total number of image patches (with 256×256 pixel size) for the Schroeder et al. conditions is 62,385, for Murphy et al. conditions is 82,036 and 143,133 for the Kumar-Roy conditions. The last bin, marked with *, has a larger interval because some few image patches had a large number of active fire pixels — keeping the bins with the same interval would result in a very sparse histogram.

Our dataset is publicly available, and contains 146,214 image patches, 256×256 pixels each, around 180 GB uncompressed, along with their corresponding segmentation masks produced by the three sets of conditions discussed above (approximately 28 GB uncompressed). The patches are 10-band, 16-bit TIFF images, with channels $c_1, \ldots, c_7$ and $c_9, \ldots, c_{11}$. The panchromatic channel c_8, with 15 meters of spatial resolution, will not be considered in this work.

Figure 5 shows some image samples with active fire and the correspond-

ing segmentation masks for the three set of conditions. It can be seen that, although the three algorithms generally agree on the presence of fire on a certain level, the pixelwise segmentation tends to differ between them. The Murphy et al. (2016) conditions seem more sensitive to low level intensities (ember), while all the algorithms have problems with very high intensities (which appear with a greenish tinge in the RGB visualization). These differences will be further discussed in Section 4.

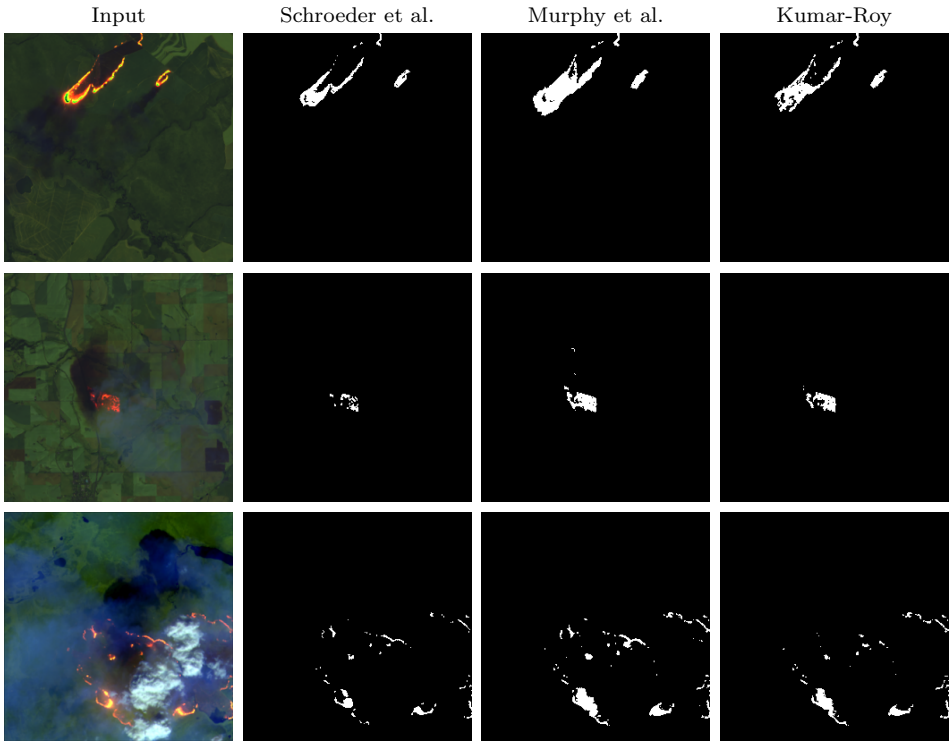


Figure 5: Image patches and the corresponding segmentation masks, extracted from the proposed dataset. For display purposes we used channels c_7, c_6 and c_2 to compose the RGB bands, however, the original patches contain 10 bands.

We believe that this dataset can prove useful and relevant for other re-

searchers working on active fire detection, since it provides the Landsat-8 data in a friendly format that can be directly used by existing machine learning tools, and provides a challenging target for benchmarking, with samples covering a large variety of scenarios, including desert, rain forest, agricultural land, water, snow, clouds, cities, mountains, etc.

2.5. Manually annotated dataset

To further validate our trained models, we selected 13 Landsat-8 images, with $\approx 7,600 \times 7,600$ pixels, which were split into 9,044 image patches with 256×256 pixels each, without overlap, in a total of 12 GB (uncompressed), captured in September 2020, and manually annotated them. Figure 6 shows the locations of these images.

It can be seen that these images are well spread spatially around the globe, covering the main continents, as well as the different features that may exist in these regions. There are active fires in 10 of these images, with different cloud cover and fire properties — some have many fire pixels, some have only a few; some have large fires, some have small isolated fire pixels; with the fire intensity also varying. The remaining 3 images have no active fires, but cover scenarios that proved challenging for at least one of the sets of conditions, which produced false positives: one from the Sahara desert, where the Schroeder et al. (2016) conditions detected fires, one from Greenland, where the Murphy et al. (2016) and Kumar and Roy (2018) conditions detected fires, and one from a city (Milan), since all the algorithms

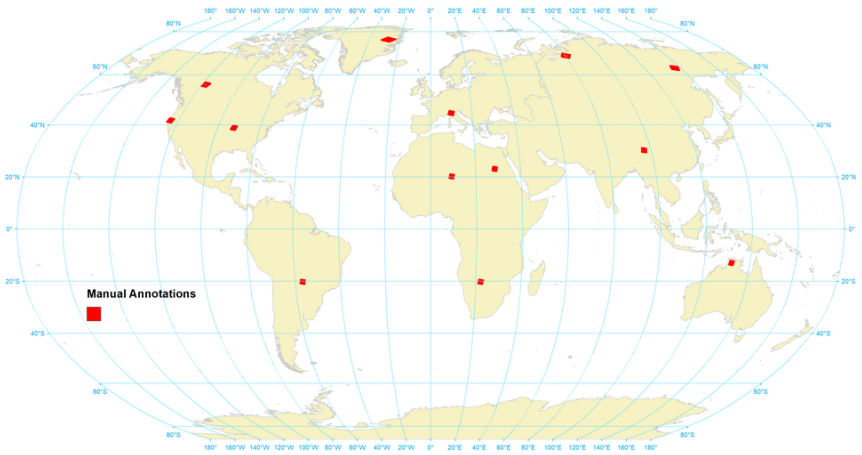


Figure 6: Active fire manual annotations: the active fire present in 13 Landsat-8 images (depicted by red squares) from September of 2020 — with $\approx 7,600 \times 7,600$ pixels and selected to represent distinct regions of the planet — were manually annotated by the authors. These images and reference masks are also available as a dataset.

seem to produce false detections inside large urban settlements. Figure 7 shows an example of manually annotated image.

The purpose of evaluating the learned models against these images is having a second reference for validating them, avoiding some potential bias that might arise from training them on the outputs from the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) algorithms. For this reason, in this work, the manually annotated images will be exclusively for testing the models, but not for training them.

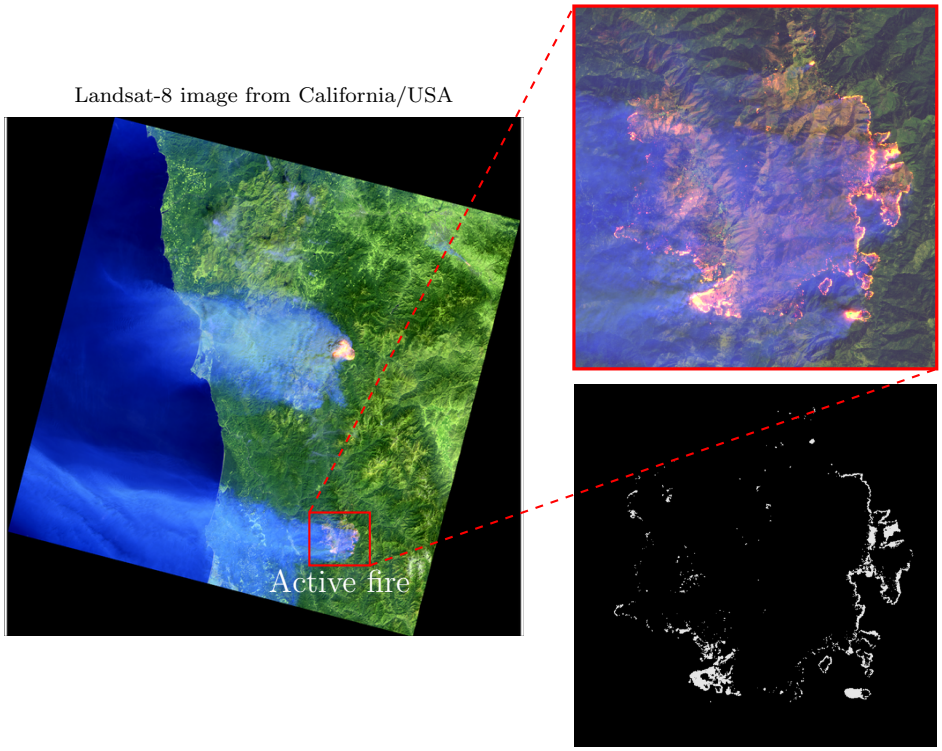


Figure 7: Manual annotation example: on the left, a Landsat-8 image from California, USA — reference code LC08_L1TP_046031_20200908_20200908_01_RT, WRS 046/031, from September 8th, 2020. This image has many wildfire regions, with a total of 55,386 active fire pixels being identified. Highlighted in red, one of the two major wildfire areas (top right) with an extension of 22 × 22 km, with approximately 10,000 active fire pixels; the corresponding manual annotations are also shown (bottom right).

3. Methods

3.1. Convolutional Neural Networks for Active Fire Detection

Convolutional neural networks (CNNs)(Goodfellow et al. (2016)) are a type of feedforward neural network architecture that, in recent years, have been enjoying a lot of attention, due to their success in a large variety of

tasks, many of them related to image processing (Garcia et al. (2018); Wang et al. (2020)), analysis (Petersson et al. (2016); Litjens et al. (2017)) and recognition (Paoletti et al. (2019); Yao et al. (2019)). This popularization is mainly because to advances both in processing capabilities — particularly the use of Graphic Processing Units (GPUs) for massively parallel computations — and algorithms and techniques, which led to the ability of learning deeper and more complex models. The reviews by Zhu et al. (2017), Ma et al. (2019a) and Yuan et al. (2020) discuss the major breakthroughs of deep learning for the remote sensing field, including main challenges, and the current state-of-the-art in environmental parameter retrieval, data fusion and downscaling, image registration, scene classification, object detection, land use and land cover classification and region segmentation.

A deep convolutional network is composed by a series of layers, each layer applying to the output of the previous layer a number of filters via the linear convolution operation, followed by some kind of activation function that introduces non-linearities (Goodfellow et al. (2016)). Deeper layers combine features extracted by previous layers, in such a way that the network is capable of encoding progressively more complex concepts (Lecun et al. (2015)). The training procedure for a CNN consists of discovering weights (coefficients and biases for the filters), which allow the input data to be transformed so as to approximate the desired output. In this paper, we consider only the so called supervised learning (Chapelle et al. (2010)), in which the network receives a number of labeled samples — inputs and their associated outputs —

and iteratively adjusts its weights via a backpropagation algorithm (Rumelhart et al. (1986)) to make its output for a given input more and more similar to the presented sample output. Although CNNs are the current state of the art technique for a number of machine learning and computer vision tasks, there are several concerns that must be kept in mind when working with them. In this paper, we do not focus on issues directly related to challenges faced by CNNs in a mathematical or algorithmic level, but we do highlight that complex architectures demand a high computational power to be trained (i.e. whenever possible, it is desirable to have smaller and simpler architectures); and that a proper model must be trained with a large amount of unbiased data, something that raises concerns over both the required infrastructure needed for transmission/storage and the availability of good quality public datasets.

The design of robust and highly optimized CNN architectures for active fire detection is not the main focus of this work. Nevertheless, it is important to demonstrate that a CNN trained on the proposed dataset is able to produce results similar to those obtained by conventional algorithms such as those from Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018). Our tests were based on networks derived from the U-Net architecture from Ronneberger et al. (2015), a very popular architecture for image segmentation tasks. U-Net is a fully convolutional network, with two symmetric halves, the first with pooling operations that decrease the data resolution and the second with upsampling operations that restore the data

to its original resolution. The first half extracts basic features and context, and the second half allows for a precise pixel-level localization of each feature, with skip connections between the two halves being used to combine the features from a layer with more abstract features from deeper layers.

Figure 8 shows the basic U-Net architecture employed in our tests. It is essentially the same concept from the original U-Net, but our implementation adds batch normalization after each convolutional layer, and contains additional dropout layers, which help avoiding problems with overfitting (Goodfellow et al. (2016)) and is related to how the error propagates during training.

We considered 3 variations of the U-Net architecture. The first one, called hereafter U-Net (10c), is the basic U-Net architecture, taking as input a 10-channel image containing all the Landsat-8 image bands. The second architecture, called hereafter U-Net (3c), keeps the same structure, but replacing the input with a 3-channel image containing only bands c_7, c_6, and c_2 — the rationale behind this decision is checking whether it is possible to obtain a good enough approximation of the results while relying on a reduced number of bands, something that may be useful for reducing bandwidth, memory usage and storage space. E.g. it is possible to have a considerable economy of bandwidth resources when processing the images in Amazon Web Services (AWS), where the original Landsat-8 images are hosted for free download and where is possible to download only selected bands.

The third architecture, called hereafter U-Net-Light (3c), is a reduced

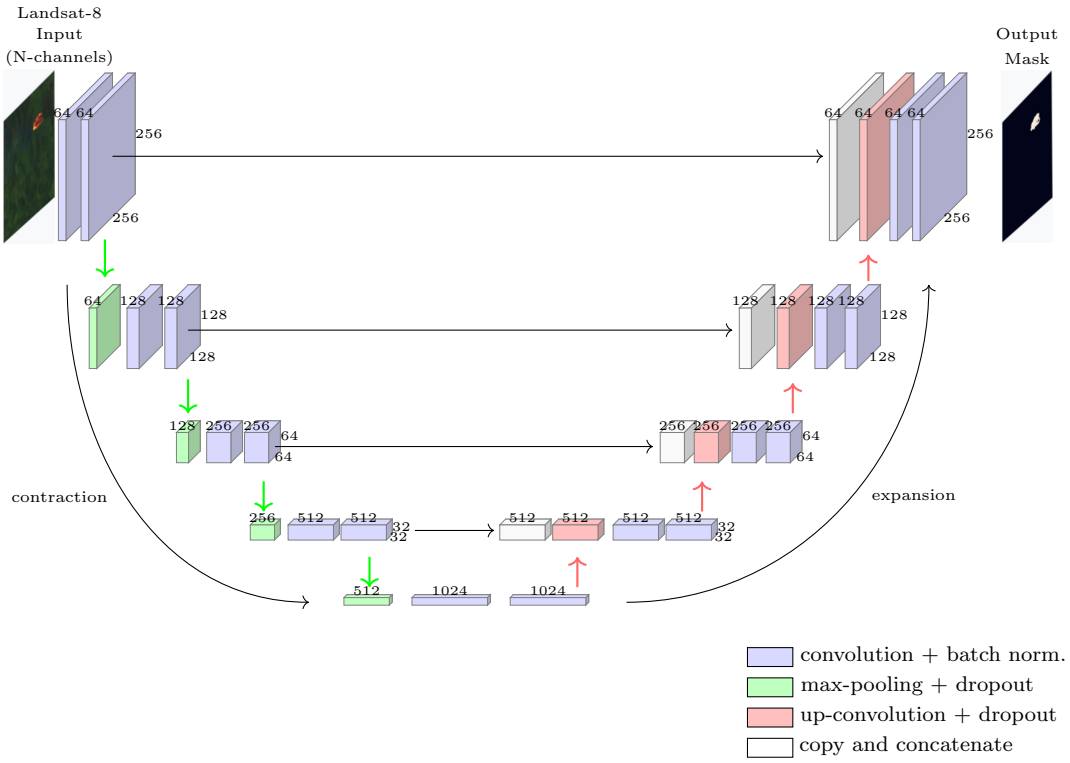


Figure 8: U-Net architecture for image segmentation.

version of the 3-channel U-Net, with the number of filters per layer being divided by 4 (i.e. 16 filters in the first layer, 32 in the second, etc.). We call this a "light" version of the original U-Net network. Relevant parameters for these architectures are shown in Table 1.

Table 1: U-Net-based architectures for active fire recognition.

	# of input channels	# of trainable parameters	model size (MB)	inference time per patch (ms)
U-Net (10c)	10	34,529,153	132	36,8
U-Net (3c)	3	34,525,121	132	36
U-Net-Light (3c)	3	2,161,649	8,5	25,5

All the variations have as output a 1-channel binary image, with 256×256

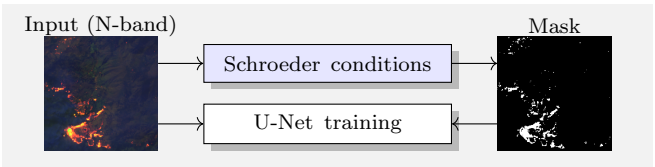
pixels, where 1 and 0 represent, respectively, fire and non-fire locations. To obtain these binary outputs, the CNN outputs are thresholded so that any pixel with value above 0.25 is set to 1 (this threshold was empirically defined after an initial test run on a small fraction of the dataset — we observed a performance gain of $\approx 1-2\%$ on the F-score, computed as explained in Section 3.3, over using the standard 0.5 value).

Each of the 3 CNN architectures was trained and tested to approximate 5 different situations: each of the 3 considered sets of conditions, as well as their intersection and a "best-of-three" voting (i.e. a pixel in the mask is set as active fire if at least two sets of conditions agree that it is a fire pixel). This adds up to a total of 15 tested scenarios, as illustrated in Figure 9.

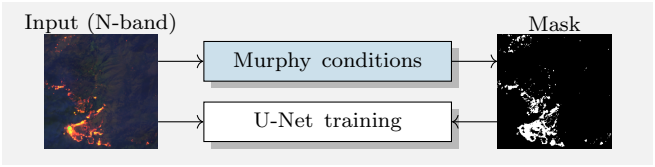
3.2. Implementation and Experimental Setup

The proposed CNN architectures were implemented and tested, to verify their ability to approximate the results from the sets of conditions from Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018). We also evaluated the performance of the CNNs and sets of conditions on a set of manually annotated images.

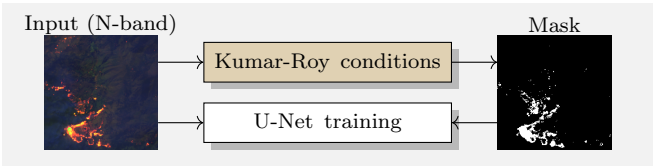
The image patches from the dataset were randomly divided in 3 sets, for training, validation and test, containing, respectively, 40%, 10% and 50% of the samples. The experiments were carried out on an Intel Core i8, 64 GB of RAM, running Linux, with a Titan Xp (12 GB) GPU. The implementation



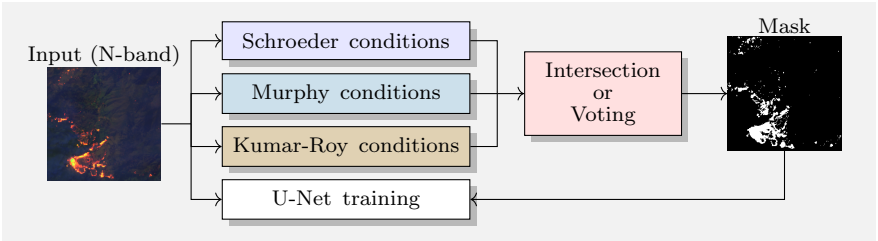
(a)



(b)



(c)



(d)

Figure 9: Different situations for training and testing the deep convolutional neural networks. Each of the 3 network architectures (U-Net (10c), U-Net (3c) and U-Net-Light (3c)) was trained and tested to approximate the outputs of 5 different sets of conditions: the 3 algorithms for active fire detection Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018), as shown in (a), (b) and (c), as well as their intersection and a "best-of-three" voting, as shown in (d).

was in the Python language, using the TensorFlow[5] library. We trained each architecture using Adam optimization (learning rate of 0.001), a batch size of

[5]www.tensorflow.org

16, and binary cross entropy as the loss function, for 50 epochs, or until the loss on the validation set was not improved for at least 5 epochs. Figure 10 shows an example of one training run (loss × epoch) — in that case, for the U-Net (3c) trained on the intersection of the 3 algorithms (Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018)). All other training runs had a similar behavior, with the number of epochs usually fluctuating between 13∼50.

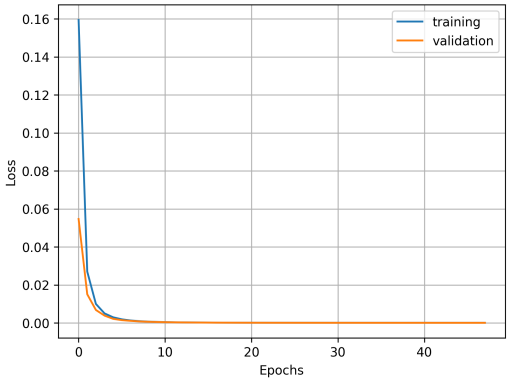


Figure 10: Training loss for the U-Net (3c), taking the intersection of 3 algorithms as target. Other training runs had a similar behavior. After around 10 epochs, there are slight variations to the loss, which are hard to see due to the scale in the y axis, which was automatically selected by the library we used.

3.3. Evaluation metrics

We evaluated the architectures according to the primary metrics used in the main semantic segmentation challenges such as PASCAL VOC[6] (Everingham et al. (2010)), KITTI[7] (Fritsch et al. (2013)), and COCO[8](Lin et al.

[6] Visual Object Challenge
[7] Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago
[8] Common Objects in Context — http://cocodataset.org

(2014)). The F-score (Chinchor and Sundheim (1993)), also known as dice coefficient, is a widely used metric for ranking purposes, and consists of the harmonic mean of precision P and recall R metrics

$$F = \frac{2}{1/P + 1/R} \tag{1}$$

such that

$$P = \frac{tp}{tp + fp} \qquad R = \frac{tp}{tp + fn} \tag{2}$$

where tp (true positives) is the number of correctly classified fire pixels, fp (false positives) the number of non-fire pixels incorrectly classified as fire, and fn (false negatives) the number of fire pixels incorrectly classified as non-fire.

We also report the Intersection-Over-Union (IoU) metric (O. Pinheiro et al. (2015)), also known as Jaccard index

$$\text{IoU} = \frac{tp}{tp + fn + fp} \tag{3}$$

The IoU metric conveys the same information as the F-score — IoU = $F/(2 - F)$) — but it is also widely in many segmentation challenges, so for clarity we report both values. All metrics were computed as per the COCO evaluation implementation: tp, fp and fn values are accumulated for all the images and the metrics are computed only once for the entire dataset (i.e. the reported values are not the average per-patch performance, but the global

per-pixel performance).

As in the KITTI challenge (Lyu et al. (2019)), we ranked the algorithm's performances according to the F-score metric. One common metric that we do not list in our experiments is the pixel accuracy metric, which takes into account the number of true negatives (non-fire pixels detected as such). In our tests, we observed a large class imbalance between fire and non-fire pixels — more than 99% of the total samples are non-fire pixels — and that always resulted in very high accuracy — even for an extreme case of a detector that fails to detect any fire pixels could still seem to perform reasonably well, making pixel accuracy a misleading metric.

4. Results and Discussion

4.1. Performance on the automatically segmented patches

For the first round of experiments, we used the CNN architectures trained and validated with 50% of the image patches against the other 50% reserved for testing, in order to show how well each trained architecture can approximate the original segmentations obtained by the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) conditions, as well as combinations of their outputs (their intersection and a "best-of-three" voting). In total, we had 15 testing scenarios, as shown in Table 2, with 3 CNN architectures (U-Net (10c), U-Net (3c) and U-Net-Light (3c)) to approximate the 5 target segmentation masks.

As can be seen, all architectures were able to reproduce the behavior of

Table 2: Active fire recognition performance: we used the set of conditions from Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) to train three convolutional neural networks for active fire segmentation: U-Net (10c) uses as input a 10-channel image from Landsat-8; U-Net (3c) uses as input a 3-channel image, containing only bands c_7, c_6, and c_2, which have a good response for active fire; and U-Net-Light (3c) is a reduced version of the 3-channel U-Net. The performances are related to the CNN ability to reproduce the masks from Schroeder et al. (2016), Murphy et al. (2016), Kumar and Roy (2018), and also the intersection and consensus of their masks, in a testing set never seen by these architectures.

Mask		CNN Architecture	Metrics (%)			
			P	R	IoU	F
Schroeder *et al.*		U-Net (10c)	86.8	**89.7**	78.9	88.2
		U-Net (3c)	89.8	88.8	**80.7**	**89.3**
		U-Net-Light (3c)	**90.8**	86.1	79.2	88.4
Murphy *et al.*		U-Net (10c)	**93.6**	92.5	87.0	93.0
		U-Net (3c)	89.1	**97.6**	87.2	93.2
		U-Net-Light (3c)	92.6	95.1	**88.4**	**93.8**
Kumar-Roy		U-Net (10c)	**84.6**	**94.1**	**80.3**	**89.1**
		U-Net (3c)	84.2	90.6	77.4	87.3
		U-Net-Light (3c)	76.8	93.2	72.7	84.2
Intersection	Schroeder *et al.*	U-Net (10c)	84.4	**99.7**	84.2	91.4
	Murphy *et al.*	U-Net (3c)	**93.4**	92.4	**86.7**	**92.9**
	Kumar-Roy	U-Net-Light (3c)	87.4	97.3	85.4	92.1
Voting	Schroeder *et al.*	U-Net (10c)	**92.9**	95.5	**89.0**	**94.2**
	Murphy *et al.*	U-Net (3c)	91.9	95.3	87.9	93.6
	Kumar-Roy	U-Net-Light (3c)	90.2	**96.5**	87.3	93.2

the different algorithms reasonably well. It is interesting to note that a larger network (U-Net (10c)) is not necessarily superior to a network that uses a reduced number of channels (U-Net (3c)), nor to a less complex network with a reduced number of channels (U-Net-Light (3c)). That means it is likely that this particular application does not demand very complex architectures, and more importantly, that channels c_7, c_6 and c_2 indeed contain most of the information needed for detecting active fires. Moreover, there is a tendency

that models with higher precision have lower recall, and vice-versa, but not to a large margin in most cases, which shows the learning procedure is balancing false and missed detections.

Figures 11, 12 and 13 compare the outputs produced by the CNNs to the outputs of, respectively, the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) sets of conditions, for image samples from the test set. It can be seen that the CNNs indeed succeed in approximating the target masks, especially considering the presence of fire in a certain area, with most differences at the pixel level being small. Even the 3-channel architectures are successful in most cases, once again confirming that channels c_7, c_6 and c_2 contain most of the information needed for detecting active fires.

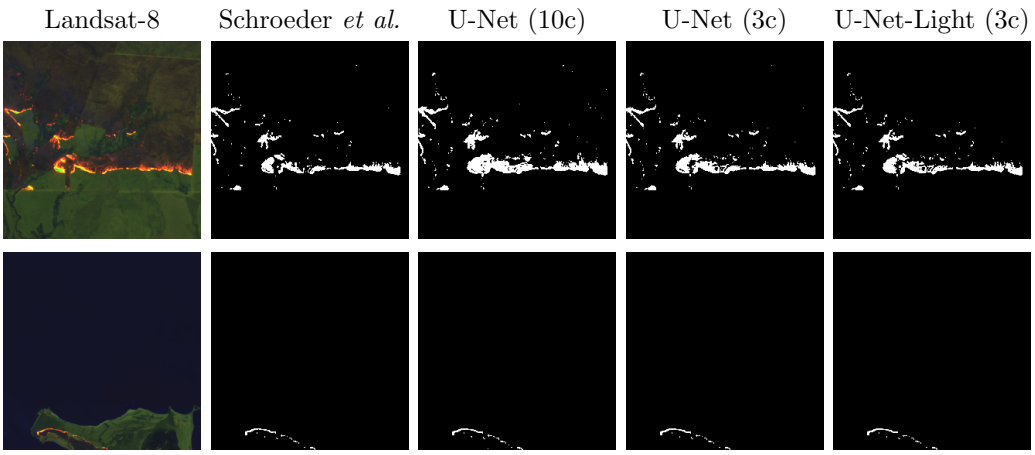


Figure 11: Active fire segmentation results obtained by the algorithm from Schroeder et al. (2016), and by the networks trained using those outputs as the target masks.

The second row from Figure 13, which has several small clusters of fire pixels detected by the original algorithm and the 10-channel CNN, but which

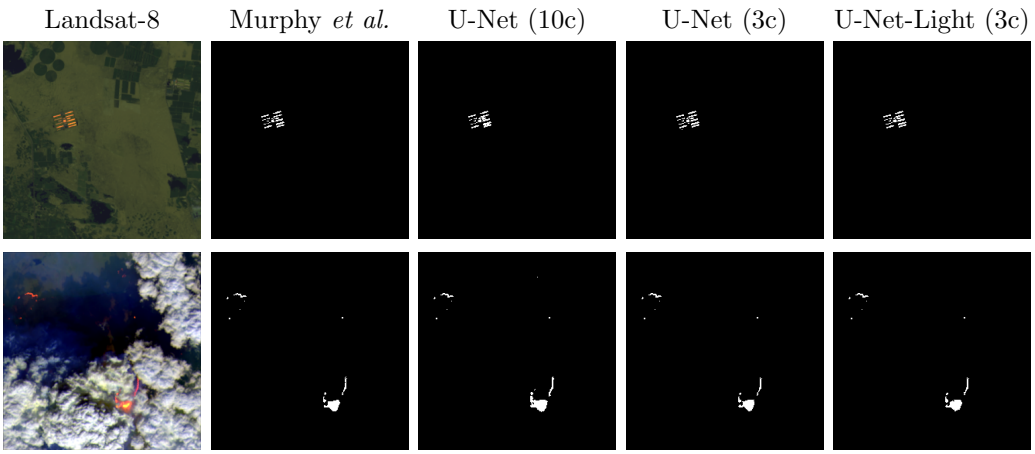


Figure 12: Active fire segmentation results obtained by the algorithm from Murphy et al. (2016), and by the networks trained using those outputs as the target masks.

were not found by the 3-channel networks (and which are, consistently, hard to see in the 3-channel visualization), is an exception of good approximation for the networks with 3-channels.

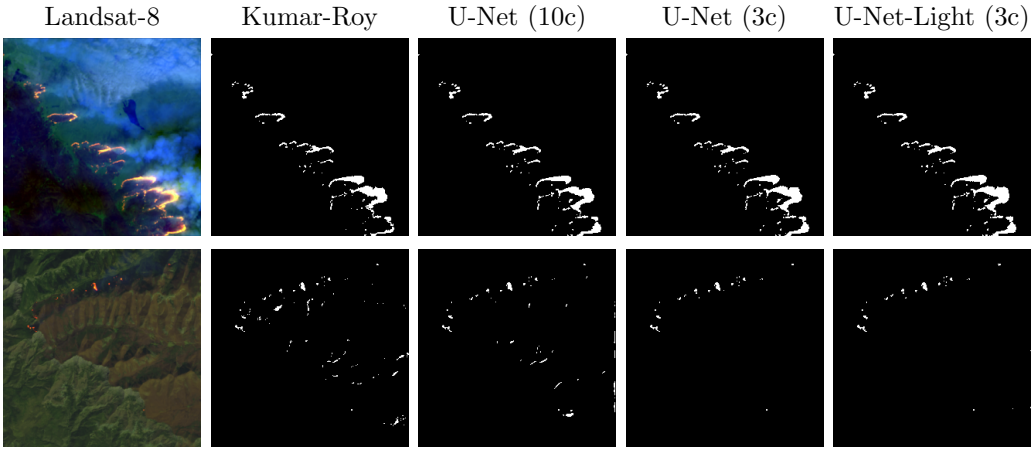


Figure 13: Active fire segmentation results obtained by the algorithm from Kumar and Roy (2018), and by the networks trained using those outputs as the target masks.

In Figures 14 and 15, we compare the outputs from the three sets of

conditions to their combination (first their intersection, then the "best-of-three" voting scheme), as well as the output of the 3-channel U-Net trained to approximate that combination.

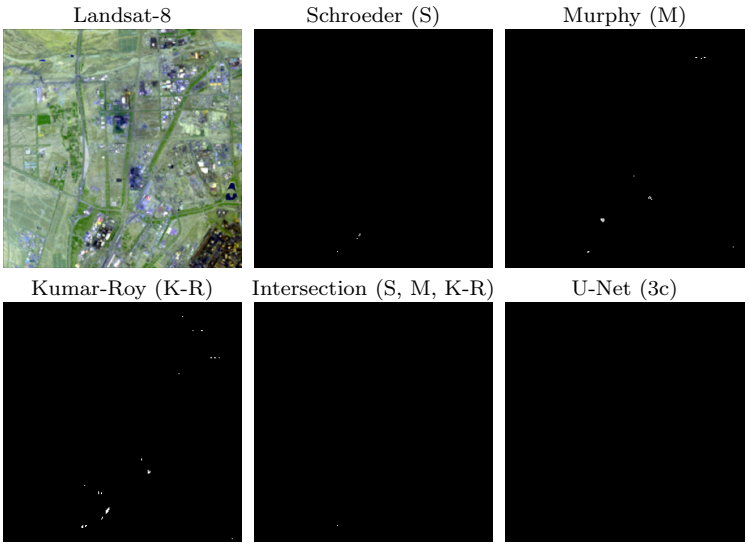


Figure 14: The Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) algorithms had false detections inside a city, but their intersection and the 3-channel U-Net trained to approximate it were able to avoid them.

Figure 14 shows that, while the three sets of conditions may incorrectly detect active fires inside urban settlements, their intersection is usually more robust (although that may also result in lower recall). As for Figure 15, it shows the voting scheme helps reducing behaviors associated with only one of the sets of conditions, such as the tendency of the Murphy *et al.* conditions to produce thicker fire clusters and, in this example, the "hole" in the lower fire region from the Kumar-Roy conditions. In both cases, the trained CNNs were successful in approximating the combined masks (both samples come from the test set). Another question brought by these images is: how do the

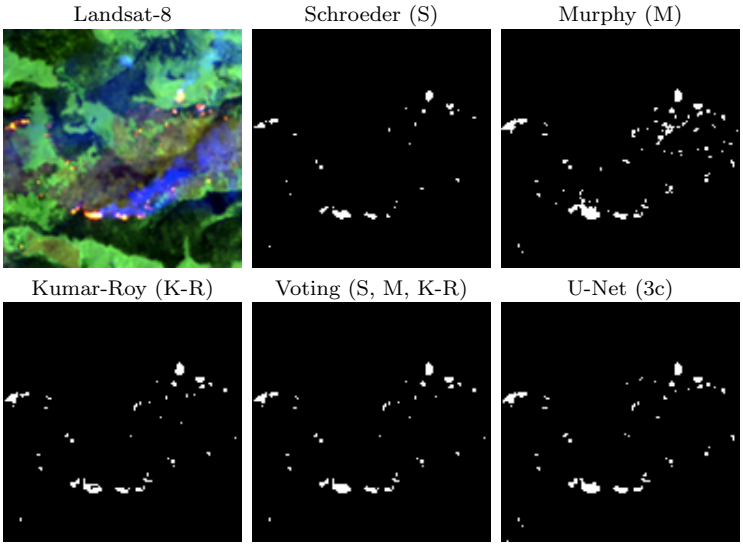


Figure 15: The 3-channel U-Net was able to approximate a "best-of-three" voting scheme that takes the outputs from the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) algorithms. The voting scheme reduces errors caused by artifacts produced by only one of the algorithms.

3 sets of conditions compare to each other, and to the networks that learn to combine them? This matter will be analyzed in more depth in Section 4.2.

4.2. Performance on the manually annotated images

For the second round of experiments, we fed the CNN models trained in the first round (Section 4.1) with Landsat-8 image patches from the manually annotated dataset, and compared the results with the hand-segmented masks. For reference, the outputs from the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) algorithms were also compared to the hand-segmented masks. This allows us to observe how well the networks learned from the automatically segmented patches perform when compared to a human specialist (one must keep in mind, however, that different specialists

would produce distinct segmentations, since it is very subjective sometimes to separate the frontiers of burnt areas from areas that are still burning, all this added to the presence of smoke and many levels of fire intensity). For works on burnt areas the reader can refer to Chen et al. (2020); Malambo and Heatwole (2020).

Table 3 shows the performances obtained for each one of the 15 scenarios involving CNN architectures, as described in section 4.1, as well as, the performances for the original set of conditions of Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) without the use of any CNN architecture. As can be seen, the best F-scores were around 90% — this is expected, as there are biases from both the human specialist labeling the data and the algorithms themselves, which are tuned to perform under certain assumptions. As an example, we also show in Figure 16 a selected image patch from Landsat-8, with a large area of active fire, along with its associated manual segmentation, and also the segmentation masks produced for each scenario being considered.

As can be seen in Table 3, the original conditions from Murphy et al. (2016) performed better than those from Schroeder et al. (2016) and Kumar and Roy (2018) in the manually annotated dataset, mostly due to its tendency to detect more fire pixels than the other conditions, resulting in a very high recall, albeit at the cost of a lower precision (i.e. it is more prone to false detections). The Kumar and Roy (2018) original conditions had lower precision and recall compared to the Schroeder et al. (2016) conditions — in

Table 3: Active fire recognition performance against **manual annotations** for 13 Landsat-8 scenes. The entries with — do not use a CNN, the performances are computed directly from the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) sets of conditions against the manual annotations. The remaining entries correspond to deep networks trained with masks automatically extracted using these sets of conditions. All entries were compared against the same manual annotations and the boldface values show the best performance for each metric.

Mask		CNN Architecture	Metrics (%)			
			P	R	IoU	F
Schroeder *et al.* (non-CNN)		—	88.1	70.2	64.1	78.1
Schroeder *et al.*		U-Net (10c)	86.9	88.4	78.0	87.7
		U-Net (3c)	89.5	80.5	73.6	84.8
		U-Net-Light (3c)	89.0	78.8	71.8	83.6
Murphy *et al.* (non-CNN)		—	76.6	96.1	74.3	85.2
Murphy *et al.*		U-Net (10c)	74.8	96.9	73.0	84.4
		U-Net (3c)	72.7	**97.2**	71.2	83.2
		U-Net-Light (3c)	75.5	96.9	73.7	84.9
Kumar-Roy (non-CNN)		—	82.3	68.4	59.7	74.7
Kumar-Roy		U-Net (10c)	82.2	94.2	78.3	87.8
		U-Net (3c)	84.5	93.4	79.8	88.8
		U-Net-Light (3c)	78.8	96.9	76.9	86.9
Intersection (non-CNN)		—	**92.6**	57.6	55.1	71.0
Intersection	Schroeder *et al.*	U-Net (10c)	91.8	75.4	70.6	82.8
	Murphy *et al.*	U-Net (3c)	90.8	73.5	68.4	81.2
	Kumar-Roy	U-Net-Light (3c)	90.8	72.8	67.8	80.8
Voting (non-CNN)		—	87.7	79.1	71.2	83.2
Voting	Schroeder *et al.*	U-Net (10c)	83.6	94.0	79.3	88.5
	Murphy *et al.*	U-Net (3c)	86.4	93.0	81.1	89.6
	Kumar-Roy	U-Net-Light (3c)	87.2	92.4	**81.4**	**89.7**

fact, it fails to detect several fire pixels in the example of Figure 16, an issue that will be further discussed briefly. All the sets of conditions had problems detecting the core of the fire at the bottom of the image, which has very high intensity.

As for the CNNs, most differences between them and the original sets of

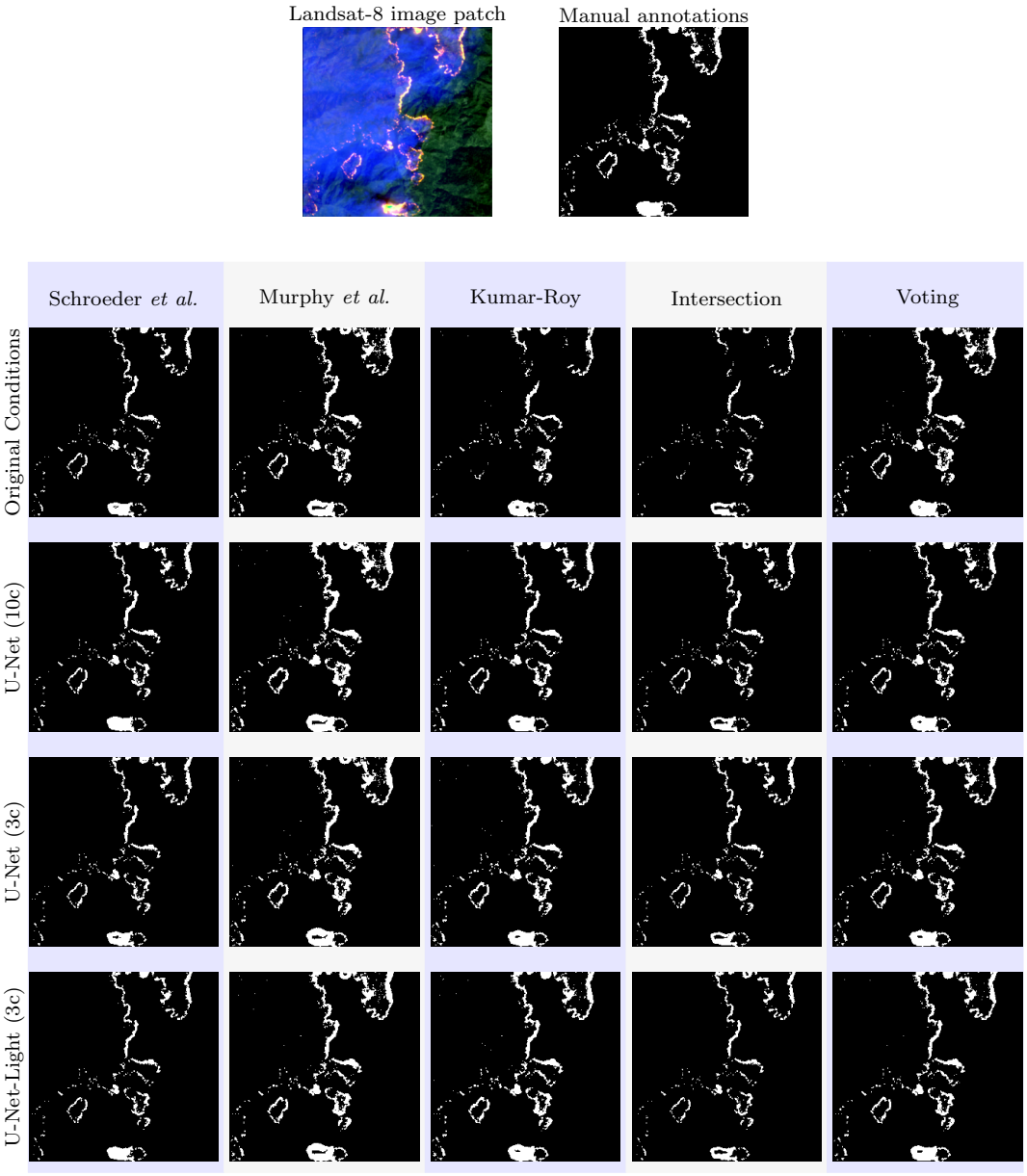


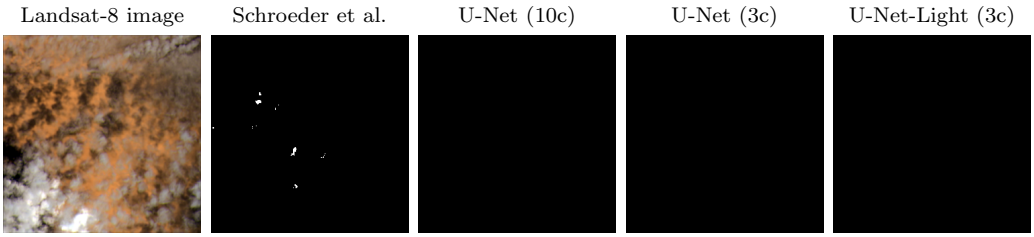
Figure 16: Active fire segmentation results produced by the Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018) sets of conditions, the deep networks trained to approximate them, and their combined outputs (intersection and "best of three" voting), for a Landsat-8 image patch.

conditions were actually positive, with the deep networks performing better than the Schroeder et al. (2016) and Kumar and Roy (2018) sets of conditions, with higher recall but without a corresponding hit on precision, as shown in Table 3. In Figure 16, the networks trained to approximate the Kumar and Roy (2018) conditions were able to detect many fire pixels missed by the original algorithm. The deep networks have also shown a higher recall (but lower precision) compared to the Murphy et al. (2016) conditions. As expected, by taking the intersection of three network outputs, we obtained higher precision but a considerably lower recall, since this combination keeps only those pixels that all three networks agree are fire pixels, that is, it is very restrictive (the intersection of the three original sets of conditions leads to the highest precision, but at the cost of a very low recall). The best overall performance — including the original sets of conditions — was obtained by the voting scheme applied to the outputs of three networks, which resulted in high recall without a great impact on the precision, with the 3-channel light U-Net being the one whose results were the most similar to the manual segmentations, but with no significant difference for the larger 3-channel U-Net. We emphasize that these networks were trained using only automatically segmented samples, so the improved performance cannot be caused by some bias learned from the manually annotated patches, which were only seen in the tests.

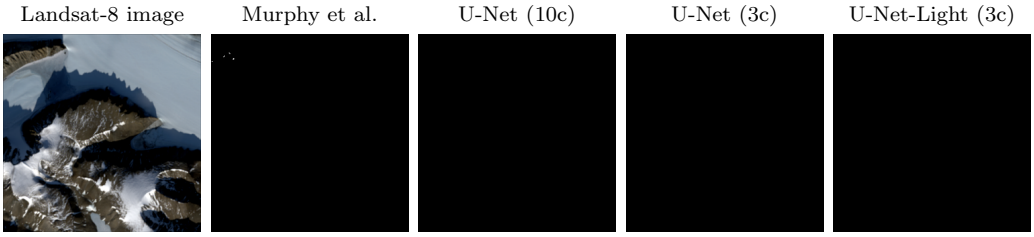
Besides the quantitative results discussed so far, we analyzed a number of individual cases where the performance or recall obtained by one of the

approaches was particularly low. One example is the segmentation from the Kumar and Roy (2018) original conditions shown in Figure 16, which missed many fire pixels. We observed that these pixels were in fact detected as fire pixels, but were later discarded after being regarded as water pixels, due to some very similar values in bands ρ_2 to ρ_5. Other cases are presented in Figure 17, which shows failure cases for each one of the sets of conditions. In the first row, Figure 17(a), the Schroeder et al. (2016) conditions produced false detections in a Sahara desert region. In the second and third rows, Figure 17(b, c), the Murphy et al. (2016) and Kumar and Roy (2018) conditions produced false detections in some Greenland regions. In the map previously shown in Figure 3 these errors can be seen as green dots in the Sahara region and blue and magenta dots in Greenland — but note the map refers to data from August 2020, while the examples in Figure 17 were taken in September 2020, indicating that the algorithms are consistently producing false detections in some situations.

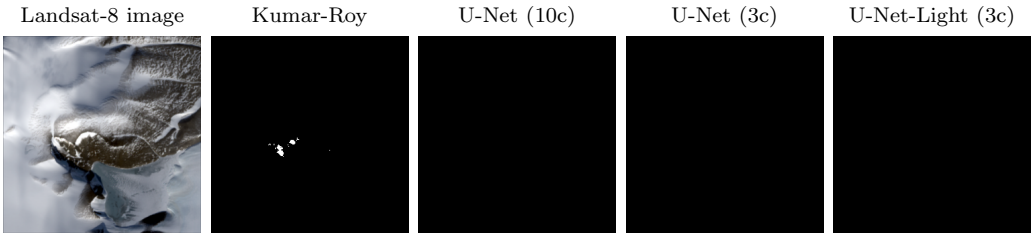
By analyzing these cases in further detail, by looking at each step and component from the conditions, we noticed that most errors were not caused by fundamental problems with the conditions themselves, but by reflectance values that appear very close to the thresholds from one rule or another. For example, some errors could be avoided if the 0.2 threshold from one unambiguous fire condition from Schroeder et al. (2016) was changed to 0.194 — but that could potentially also lead to real fires not being detected. This highlights one of the challenges faced when handcrafting sets of rules: some



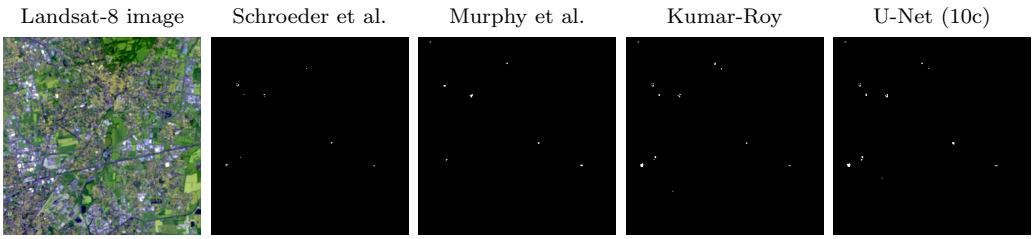
(a) Segmentations from Schoeder et al. conditions and for U-Net architectures based of his conditions.



(b) Segmentations from Murphy conditions and for U-Net architectures based of his conditions.



(c) Segmentations from Kumar-Roy conditions and for U-Net architectures based of his conditions.



(d) Segmentations from Schoeder et al., Murphy et al, Kumar-Roy and a U-Net architecture.

Figure 17: Classification errors: (a) region from Sahara desert WRS: 189/046, date 2020/09/18; (b) and (c) from Greenland, WRS: 003/006, date: 2020/09/11; and (d) city of Milan, Italy, WRS: 193/029, date: 2020/09/14, where all original conditions and CNN architectures reported misclassifications for bright and/or reflective urban elements.

situations must be simplified to keep the algorithm readable and understandable, so thresholds are usually rounded to 1 or 2 decimal places, and only the most important relations between bands are considered. Machine learning techniques are able to encode more complex rules, with more precise weights, coefficients and thresholds; although this may also mean that the learned rules and relations can be hard to understand and describe by a human. In any case, the deep networks were able to avoid the errors in the samples shown in the first three rows of Figure 17. One case that was not solved by the CNNs is shown in the last row (d) of Figure 17 — all the approaches produced false detections around urban areas. From these examples, we believe that these kinds of persistent errors must be addressed differently, by considering temporal analysis as described by Schroeder et al. (2016), since it is very unlikely that an active fire will remain active at the same location for several months.

4.3. Discussion

In sections 4.1 and 4.2, we discussed particular results regarding, respectively, how well the deep learning models could approximate the results from handcrafted algorithms, and how the CNNs and handcrafted sets of conditions performed when compared to a human specialist. The main findings were that the CNNs can indeed approximate the handcrafted algorithms, even with a reduced architecture and relying on only 3 channels of sensor data. Moreover, we noted that, by combining multiple CNN outputs, it is

possible to obtain better performance compared to the original sets of conditions, and the situations where the CNNs differ from the original algorithms can actually be positive, allowing more complex rules and relations, albeit with the trade-off of being hard to understand or describe by a human.

It is worth noting that the CNN architectures we described in this paper were the ones that reported the most promising results, however, several other variations were tested, such as architectures without batch normalization (which failed, most likely due to the vanishing gradients problem), variations with more and less convolutional layers or filters per layer (i.e. deeper/shallower, and wider/narrower architectures). We emphasize that we did not focus on finding highly optimized CNN architectures, the feasibility of the approach being our main concern. Thus, the results we obtained could be potentially improved in a number of ways: future research could explore the use of image super-resolution (Gargiulo et al. (2019), Ma et al. (2019b)); ensembles of distinct convolutional neural network architectures (Minetto et al. (2019)); and spatio-temporal information (Boschetti et al. (2017)) to differentiate between temporally-persistent fire pixels from places like factories or rooftops from deforestation wildfires. Furthermore, machine learning-driven approaches can be quickly adapted to new sensors (Mateo-Garca et al. (2020)) (e.g. Sentinel-2), possibly relying on transfer learning, without requiring specifically designed sets of conditions.

5. Conclusions

In this paper, we addressed the problem of active fire detection using deep learning techniques. We introduced a new large-scale dataset, containing 146,214 image patches extracted from the Landsat-8 satellite, along with associated outputs produced by three well established handcrafted algorithms (Schroeder et al. (2016), Murphy et al. (2016) and Kumar and Roy (2018)). A second dataset was also created, with 9,044 image patches and annotations manually produced by a human specialist. We hope these datasets can prove useful for the community, since they set a challenging target while keeping the data in a friendly format for existing machine learning tools, allowing researchers to test different architectures and approaches. Besides the datasets, we presented a study on how convolutional neural networks can be used do approximate the outputs from the considered handcrafted algorithms, as well as on how the outputs from multiple models can be combined to achieve better performance than the individual sets of conditions.

Our study showed that handcrafted sets of conditions for active fire recognition are hard to be defined, and small variations in fixed thresholds may cause false positives or negatives, on the other hand, deep learning techniques are able to encode more complex rules to improve over this aspect but at the cost of relations that are hard to understand and describe by a human.

Future work shall focus on exploring alternative methods for active fire detection over the proposed datasets, as well as better exploring the hy-

pothesis that models trained over samples from one satellite can perform well when applied to other satellites. It is also possible to develop means of combining detection results from multiple satellites, with different responses, spatial resolutions, and capture cycles; aiming at improving dependability and producing estimates with higher resolution in both space and time.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors would like to thank also the research Brazilian agencies CNPq, CAPES and FAPESP.

References

Ba, R., Chen, C., Yuan, J., Song, W., Lo, S., 2019. SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention. Remote Sensing 11, 1–22. doi:10.3390/rs11141702.

Ban, Y., Zhang, P., Nascetti, A., Bevington, A.R., Wulder, M.A., 2020. Near Real-Time Wildfire Progression Monitoring with Sentinel-1 SAR Time Series and Deep Learning. Scientific Reports 10. doi:10.1038/s41598-019-56967-x.

Bermudez, J.D., Happ, P.N., Q's, R., 2019. Synthesis of Multispectral Optical Images From SAR/Optical Multitemporal Data Using Conditional

Generative Adversarial Networks 16, 1220–1224. doi:10.1109/LGRS.2019.2894734.

Boschetti, M., Busetto, L., Manfron, G., Laborte, A., Asilo, S., Pazhanivelan, S., Nelson, A., 2017. Phenorice: A method for automatic extraction of spatio-temporal information on rice crops using satellite data time series. Remote Sensing of Environment (Elsevier) 194, 347 – 365. doi:10.1016/j.rse.2017.03.029.

Cardil, A., Monedero, S., Ramrez, J., Silva, C.A., 2019. Assessing and reinitializing wildland fire simulations through satellite active fire data. Journal of Environmental Management (Elsevier) 231, 996–1003. doi:https://doi.org/10.1016/j.jenvman.2018.10.115.

Chapelle, O., Schlkopf, B., Zien, A., 2010. Semi-Supervised Learning. 1st ed., The MIT Press. doi:10.5555/1841234.

Chen, D., Loboda, T.V., Hall, J.V., 2020. A systematic evaluation of influence of image selection process on remote sensing- based burn severity indices in north american boreal forest and tundra ecosystems. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 159, 63 – 77. doi:10.1016/j.isprsjprs.2019.11.011.

Chinchor, N., Sundheim, B., 1993. Muc-5 evaluation metrics, in: Conference on Message Understanding, Association for Computational Linguistics, USA. p. 6978. doi:10.3115/1072017.1072026.

Chuvieco, E., Mouillot, F., van der Werf, G.R., San Miguel, J., Tanase, M., Koutsias, N., Garca, M., Yebra, M., Padilla, M., Gitas, I., Heil, A., Hawbaker, T.J., Giglio, L., 2019. Historical background and current developments for mapping burned area from satellite earth observation. Remote Sensing of Environment (Elsevier) 225, 45 – 64. doi:10.1016/j.rse.2019.02.013.

Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision (IJCV) 88, 303338. doi:10.1007/s11263-009-0275-4.

Ferreira, L.N., Vega-Oliveros, D.A., Zhao, L., Cardoso, M.F., Macau, E.E., 2020. Global fire season severity analysis and forecasting. Computers & Geosciences (Elsevier) 134, 104339. doi:10.1016/j.cageo.2019.104339.

Flannigan, M.D., Haar, T.H.V., 1986. Forest fire monitoring using NOAA satellite AVHRR. Canadian Journal of Forest Research 16, 975–982. doi:10.1139/x86-171.

Fritsch, J., Khnl, T., Geiger, A., 2013. A new performance measure and evaluation benchmark for road detection algorithms, in: International IEEE Conference on Intelligent Transportation Systems, pp. 1693–1700. doi:10.1109/ITSC.2013.6728473.

Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J., 2018. A survey on deep learning tech-

niques for image and video semantic segmentation. Applied Soft Computing (Elsevier) 70, 41 – 65. doi:10.1016/j.asoc.2018.05.018.

Gargiulo, M., Dell'Aglio, D.A.G., Iodice, A., Riccio, D., Ruello, G., 2019. A CNN-Based Super-Resolution Technique for Active Fire Detection on Sentinel-2 Data, in: PhotonIcs Electromagnetics Research Symposium (Spring), pp. 418–426. doi:10.1109/PIERS-Spring46901.2019.9017857.

Giglio, L., Descloitres, J., Justice, C.O., Kaufman, Y.J., 2003. An enhanced contextual fire detection algorithm for modis. Remote Sensing of Environment (Elsevier) 87, 273 – 282. doi:10.1016/S0034-4257(03)00184-6.

Giglio, L., Schroeder, W., Justice, C.O., 2016. The collection 6 MODIS active fire detection algorithm and fire products. Remote Sensing of Environment (Elsevier) 178, 31–41. doi:10.1016/j.rse.2016.02.054.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. The MIT Press.

Ji, Y., Stocker, E., 2002. Seasonal, intraseasonal, and interannual variability of global land fires and their effects on atmospheric aerosol distribution. Journal of Geophysical Research: Atmospheres 107, ACH 10–1–ACH 10–11. doi:10.1029/2002JD002331.

Kaufman, Y.J., Justice, C.O., Flynn, L.P., Kendall, J.D., Prins, E.M., Giglio, L., Ward, D.E., Menzel, W.P., Setzer, A.W., 1998. Potential global fire

696 monitoring from eos-modis. Journal of Geophysical Research: Atmo-
697 spheres 103, 32215–32238. doi:10.1029/98JD01644.

698 Kondratyev, K.Y., Dyachenko, L., Binenko, V., Chernenko, A., 1972. De-
699 tection of Small Fires and Mapping of Large Forest Fires by Infrared Im-
700 agery, in: International Symposium on Remote Sensing of Environment,
701 p. 12971303.

702 Kumar, S.S., Roy, D.P., 2018. Global operational land imager Landsat-8
703 reflectance-based active fire detection algorithm. International Journal of
704 Digital Earth 11, 154–178. doi:10.1080/17538947.2017.1391341.

705 Langford, Z., Kumar, J., Hoffman, F., 2018. Wildfire Mapping in Interior
706 Alaska Using Deep Neural Networks on Imbalanced Datasets, in: IEEE
707 International Conference on Data Mining Workshops, pp. 770–778. doi:10.
708 1109/ICDMW.2018.00116.

709 Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
710 doi:10.1038/nature14539.

711 Lee, T.F., Tag, P.M., 1990. Improved Detection of Hotspots using the
712 AVHRR 3.7-um Channel. Bulletin of the American Meteorological Society
713 71, 1722 – 1730. doi:10.1175/1520-0477.

714 Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár,
715 P., Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context,
716 in: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), European

Conference on Computer Vision (ECCV), Springer, Cham. pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Snchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical Image Analysis (Elsevier) 42, 60 – 88. doi:10.1016/j.media.2017.07.005.

Lyu, Y., Bai, L., Huang, X., 2019. Road Segmentation using CNN and Distributed LSTM, in: IEEE International Symposium on Circuits and Systems, pp. 1–5. doi:10.1109/ISCAS.2019.8702174.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019a. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 152, 166–177. doi:10.1016/j.isprsjprs.2019.04.015.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019b. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 152, 166 – 177. doi:10.1016/j.isprsjprs.2019.04.015.

Maier, S.W., Russell-Smith, J., Edwards, A.C., Yates, C., 2013. Sensitivity of the modis fire detection algorithm (mod14) in the savanna region of the northern territory, australia. ISPRS Journal of Photogrammetry and

Remote Sensing (Elsevier) 76, 11 – 16. doi:10.1016/j.isprsjprs.2012.11.005. terrestrial 3D modelling.

Malambo, L., Heatwole, C.D., 2020. Automated training sample definition for seasonal burned area mapping. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 160, 107 – 123. doi:10.1016/j.isprsjprs.2019.11.026.

Mateo-Garca, G., Laparra, V., Lpez-Puigdollers, D., Gmez-Chova, L., 2020. Transferring deep learning models for cloud detection between landsat-8 and proba-v. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 160, 1 – 17. doi:10.1016/j.isprsjprs.2019.11.024.

Matson, M., Holben, B., 1987. Satellite detection of tropical burning in brazil. International Journal of Remote Sensing 8, 509–516. doi:10.1080/01431168708948657.

Minetto, R., Segundo, M.P., Sarkar, S., 2019. Hydra: An ensemble of convolutional neural networks for geospatial land classification. IEEE Transactions on Geoscience and Remote Sensing 57, 6530–6541. doi:10.1109/TGRS.2019.2906883.

Morisette, J.T., Giglio, L., Csiszar, I., Justice, C.O., 2005. Validation of the modis active fire product over southern africa with aster data. International Journal of Remote Sensing 26, 4239–4264. doi:10.1080/01431160500113526.

Murphy, S.W., de Souza Filho, C.R., Wright, R., Sabatino, G., Correa Pabon, R., 2016. Hotmap: Global hot target detection at moderate spatial resolution. Remote Sensing of Environment (Elsevier) 177, 78 – 88. doi:10.1016/j.rse.2016.02.027.

O. Pinheiro, P.O., Collobert, R., Dollar, P., 2015. Learning to Segment Object Candidates, in: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems (NIPS), pp. 1990–1998. doi:10.5555/2969442.2969462.

Paoletti, M., Haut, J., Plaza, J., Plaza, A., 2019. Deep learning classifiers for hyperspectral imaging: A review. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 158, 279 – 317. doi:10.1016/j.isprsjprs.2019.09.006.

Petersson, H., Gustafsson, D., Bergstrom, D., 2016. Hyperspectral image analysis using deep learning a review, in: International Conference on Image Processing Theory, Tools and Applications, pp. 1–6. doi:10.1109/IPTA.2016.7820963.

Pinto, M.M., Libonati, R., Trigo, R.M., Trigo, I.F., DaCamara, C.C., 2020. A deep learning approach for mapping and dating burned areas using temporal sequences of satellite images. ISPRS Journal of Photogrammetry and Remote Sensing (Elsevier) 160, 260 – 274. doi:10.1016/j.isprsjprs.2019.12.014.

779 Portillo-Quintero, C., Sanchez-Azofeifa, A., Marcos do Espirito-Santo, M.,
780 2013. Monitoring deforestation with modis active fires in neotropical dry
781 forests: An analysis of local-scale assessments in mexico, brazil and bolivia.
782 Journal of Arid Environments (Elsevier) 97, 150–159. doi:https://doi.
783 org/10.1016/j.jaridenv.2013.06.002.

784 Ronneberger, O., P.Fischer, Brox, T., 2015. U-Net: Convolutional Net-
785 works for Biomedical Image Segmentation, in: Medical Image Computing
786 and Computer-Assisted Intervention, Springer. pp. 234–241. doi:10.1007/
787 978-3-319-24574-4_28.

788 Roy, D., Wulder, M., Loveland, T., et al., 2014. Landsat-8: Science and
789 product vision for terrestrial global change research. Remote Sensing of
790 Environment (Elsevier) 145, 154 – 172. doi:10.1016/j.rse.2014.02.001.

791 Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning Represen-
792 tations by Back-propagating Errors. Nature 323, 533–536. doi:10.1038/
793 323533a0.

794 Schroeder, W., Oliva, P., Giglio, L., Csiszar, I.A., 2014. The new viirs
795 375m active fire detection data product: Algorithm description and ini-
796 tial assessment. Remote Sensing of Environment (Elsevier) 143, 85 – 96.
797 doi:10.1016/j.rse.2013.12.008.

798 Schroeder, W., Oliva, P., Giglio, L., Quayle, B., Lorenz, E., Morelli, F.,

2016. Active fire detection using Landsat-8/OLI data. Remote Sensing of Environment (Elsevier) 185, 210 – 220. doi:10.1016/j.rse.2015.08.032.

Wang, Z., Chen, J., Hoi, S.C.H., 2020. Deep learning for image super-resolution: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 1, 1–22. doi:10.1109/TPAMI.2020.2982166.

Yao, G., Lei, T., Zhong, J., 2019. A review of convolutional-neural-network-based action recognition. Pattern Recognition Letters (Elsevier) 118, 14 – 22. doi:10.1016/j.patrec.2018.05.018.

Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., Xu, H., Tan, W., Yang, Q., Wang, J., Gao, J., Zhang, L., 2020. Deep learning in environmental remote sensing: Achievements and challenges. Remote Sensing of Environment (Elsevier) 241, 111716. doi:10.1016/j.rse.2020.111716.

Zhu, X.X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine 5, 8–36. doi:10.1109/MGRS.2017.2762307.