Automatic Wildfire Mapping Using Convolutional Neural Networks and Infrared Line-Scan Images

Rian Dolphin
University College Dublin, Ireland
June 22, 2021

1 Introduction

Every year wildfires affect several hundred million hectares of forest and other vegetation around the world [1]. Though wildfires are critical for ecosystems in some geographical regions, they also cause significant damage from an ecological, economic and human point of view [2]. When burning uninterrupted, wildfires have a rapid forward rate of spread. As a result, time is of the essence and bushfire managers require timely, accurate information about the location and rate of spread of active fires in order to tackle them effectively.

A primary source of information on wildfire intensity and location is obtained from aircraft carrying infrared cameras which fly over and record the intensity and location of fires. The camera scans the fire in lines to construct an image known as an 'infrared line-scan', an example of which is shown in Figure 1. Currently, the Country Fire Authority (CFA) label the fire boundaries in each line-scan image manually by hand-drawing polygons around the edges of the fire using geospatial software. However, during times of intense firefighting activity, this process can create a bottleneck in delivering timely information to operational firefighting teams.

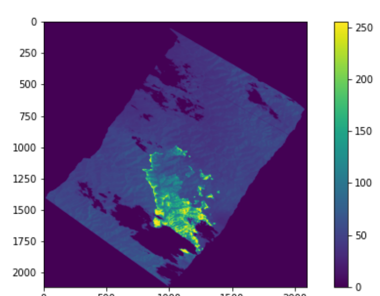


Figure 1: Example of a Line-Scan Image

In order to address this issue, we propose a computer vision framework which automatically classifies each pixel of a line-scan as being "on fire" or "not on fire". This modelling framework drastically increases the speed at which line-scan images can be analysed and wildfires identified. The benefit of this is twofold: not only can fires be identified and responded to faster but also CFA man hours can be utilised more effectively on other tasks.

2 Problem Considerations

Before developing the modelling framework, we spent considerable time thinking about the challenge at hand and how we could shape the future of wildfire mapping. Currently, line-scan aircraft are manned aircraft which first capture the line-scan images while in the air and then these images are given to the CFA for human analysis. In this submission we propose a uniqe and innovative solution that involves transitioning to unmanned aerial vehicles (UAVs) and utilising the rapidly evolving area of edge computing to effectively implement the computer vision framework we have developed.

Since wildfires are often inaccessible by ground vehicles due to mountainous terrain [3], it is no surprise that line-scan aircraft are an essential tool in the CFA's arsenal. However, flying over wildfires can be extremely dangerous, with thirty-seven firefighters having died in aerial firefighting accidents in the last decade [4]. Consequently, we believe that UAVs are a promising means of monitoring large forest fires and have the potential to create a better working world for firefighters across the globe, a core value of ours.

Edge computing is a distributed computing framework that brings both computation and data storage closer to the location where it is needed. There are numerous benefits when compared to a traditional cloud-based computing paradigm [5]; however, most relevant to the wildfire mapping domain is the significantly reduced time between an observation flight and the availability of actionable information. The trade-off we must contend with is that of reduced computational power since the data is being processed "on the edge" (i.e. on the drone/UAV itself). As a result, we propose a model with a relatively small computational cost and memory usage.

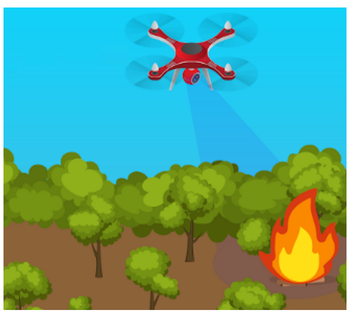


Figure 2: Illustration of UAV Approach

3 The Model

In this section we describe our modelling approach in detail along with the reasoning behind each element of it and the journey to reach our final model.

3.1 Initial Steps

The data available for this challenge is kindly provided by the mapping officers at Victoria's CFA. It consists of line-scan images along with fire-edge polygons from the 2018/19 fire season. These polygons act as the ground truth where an area enclosed by any one of the polygons has been deemed as "on fire" by the CFA.

The task at hand is to classify whether a given coordinate within a line-scan is "on fire" or not. Each pixel in the line-scan images ranges from 0-255 in value which immediately allows for a simple threshold model to act as a baseline. Following this approach, we looked to use a wider array of pixels surrounding the centre pixel. Using both a feature-based machine learning approach as well as direct application of vanilla feed-forward neural networks on the surrounding pixels reduced model error but still left room for improvement.

Convolutional Neural Networks (CNNs) are among the most successful and widely used architectures in the deep learning community, especially for computer vision tasks [6]. In fact, they specialise in processing data that has a grid-like topology. For this reason, they are a natural choice for this problem. Transitioning to a CNN framework was the most important breakthrough in improving our score. The final architecture will be discussed in further detail in the following section.

3.2 Model Framework

For our final modelling framework, one key idea was the aforementioned method of using an array of pixels centred around the target pixel, rather than using the whole image, as an input. In other words, given the target pixel location, the model considered only a square sub-image surrounding that coordinate. We hoped that this would provide two benefits: firstly, to reduce the amount of noise in the input by only considering the area immediately surrounding the target coordinates, and secondly, to reduce the input size of our model which would save memory and speed up computation time.

Of course, the size of this grid is a hyper-parameter and many different sizes were tested to find the best-performing. In the end we settled on using a 160×160 grid centred on the target pixel as our array. However, this resulted in 25,600 values which we would ideally reduce as it negatively impacts on both the memory usage and computational power; two influential elements in our proposed edge computing implementation.

To address this issue we decided to sample every second pixel which reduced our final model input to an 81×81 array. As a result, the number of values in our input was reduced by 75% but validation test-

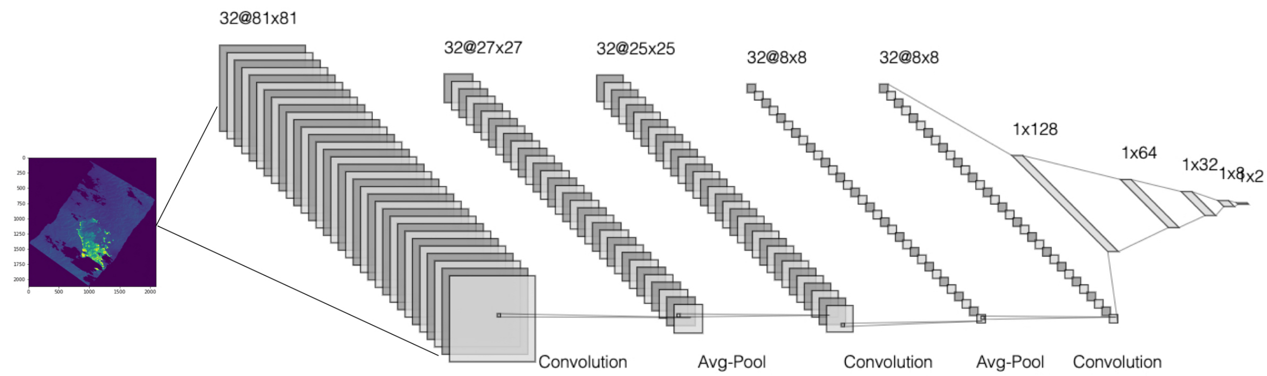


Figure 3: Model Architecture

ing indicated that there was minimal impact on performance. Figure 4 shows three examples of these model input arrays.

The CNN model architecture is illustrated in Figure 3 and is made up of:

- Three convolutional layers, each containing 32 filters with kernal size 3×3 and stride 1.
- An average pooling layer after each convolutional layer with 3×3 pool size.
- Four fully connected layers taking the flattened convolutional output as an input and finally outputting a probability that the centre pixel is "on fire".

Though the CNN architecture may seem complex, this is actually not the case. Firstly, the number of trainable parameters is actually quite low relative to a vanilla feed-forward neural network. This is because the filters are iteratively applied to the input array which limits the number of parameters. Additionally, the pooling layers are an effective way of reducing the dimensionality of the input data while retaining information. It is worth noting that max pooling is the most common approach for CNNs; however, validation analysis indicated that average pooling achieved superior results for this particular application.

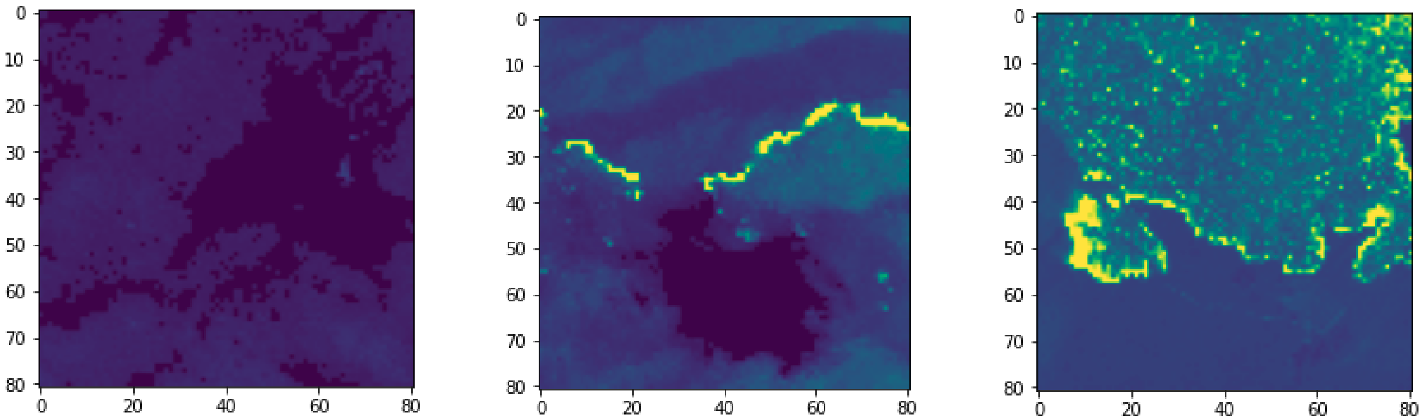
Secondly, there are no complex calculations needed to compute features. For example, consider a simple feature such as taking the mean pixel value in an image results in summing approximately 6.25 million numbers[1]. As such, the proposed architecture is made up of relatively simple computations and the resulting time taken to generate a prediction is just 0.043 seconds.

3.3 Training & Validation

Given that the model input is obtained by taking a small subsection of each line-scan, the 129 training line-scan images resulted in millions of unique training cases. However, due to the memory limitations on the Azure virtual machine all of the training cases could not be stored simultaneously. As a result, training was carried out using random subsets of training data at any given time.

One major consideration given the data provided was that of class imbalance. In the training line-scans the number of "on fire" pixels is hugely outnumbered by "not on fire" pixels. As a result, it is imperative that care is taken to avoid algorithmic bias against the minority class creeping into the model. In order to account for this we utilised over-

[1] Average image size assumed to be 2500×2500.



(a) $\mathbb{P}$(Fire)=1% (b) $\mathbb{P}$(Fire)=42% (c) $\mathbb{P}$(Fire)=99%

Figure 4: Examples of Input – Output Pairs

sampling to create a more even distribution among the training data.

The model was trained using a categorical cross entropy loss function and the RMSprop optimizer in Keras. Additionally, batches were used to produce more stable convergence and dropout was applied to each layer as a safeguard against overfitting.

When engaging in machine learning it is essential that data leakage is prevented in both validation and testing. During the course of the competition, we ensured that a portion of the training data was reserved for hyperparameter tuning and evaluation prior to submission. In terms of testing and final scores, the design of the competition ensured there was a hold-out testing set which competitors did not have access to and this provided a fair means of evaluation.

3.4 Potential Limitations

The core idea of deep learning is that the algorithm itself progressively "learns" higher-level features from the raw input. Though a major advantage in some respects, less feature engineering needed for example, it can also be a hindrance as it often reduces interpretability and prevents feature importance analysis. In real-world application this opacity can often be a deal breaker; however, we believe that in this particular application domain being alerted to danger areas quickly and accurately is more important than knowing exactly why an area was alerted upon.

Another potential limitation of our approach is that deep learning models require a lot of time and data to train on. However, the image data possessed by the CFA provides a huge amount of training data and, given the relatively low number of trainable parameters in our model, the computational resources required would be very reasonable.

A major limitation of our bigger picture approach is, of course, the expertise and funds needed to transition to UAVs and an edge computing infrastructure. However, recent literature in the area has suggested that this is the way forward for wildfire detection and a number of cost effective implementations have been proposed [7].

4 Conclusion

In conclusion, we have proposed an innovative big-picture approach to tackle the growing problem of wildfires which is applicable across the globe. In addition to this, the details of a comprehensive deep learning framework has been presented which can quickly and accurately identify areas where wildfires are burning. When developing the approach, the application domain and end users were kept in mind at all times and as such certain features were prioritised such as speed and accuracy.

The firefighters and communities on the front-line remained at the forefront of our approach throughout the competition and we truly believe that our proposed solution has the potential to build a better world where the damage caused by wildfires can be successfully contained.

Acknowledgements: Thank you to the team at EY for organising this challenge, all the judges and Microsoft for use of the Azure platform. A note that in this written submission and the accompanying video I have referred to "we" and "our" for better readability but all work was carried out solely by me.

References

[1] Johann Georg Goldammer. Early warning systems for the prediction of an appropriate response to wildfires and related environmental hazards. *Health Guidelines for Vegetation Fire Events*, pages 6–9, 1998.

[2] Penelope Morgan, Colin C Hardy, Thomas W Swetnam, Matthew G Rollins, and Donald G Long. Mapping fire regimes across time and space: understanding coarse and fine-scale fire patterns. *International Journal of Wildland Fire*, 10(4):329–342, 2001.

[3] David W Casbeer, Randal W Beard, Timothy W McLain, Sai-Ming Li, and Raman K Mehra. Forest fire monitoring with multiple small uavs. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 3530–3535. IEEE, 2005.

[4] Is aerial firefighting worth it? https://www.hcn.org/issues/47.13/after-a-record-setting-wildfire-a-washington-county-prepares-for-the-next-one/the-cost-benefit-analysis-of-aerial-firefighting. Accessed: 2021-06-02.

[5] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[6] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[7] Osama M Bushnaq, Anas Chaaban, and Tareq Y Al-Naffouri. The role of uav-iot networks in future wildfire detection. *IEEE Internet of Things Journal*, 2021.