

# A deep learning approach for early wildfire detection from hyperspectral satellite images

Nguyen Thanh Toan, Phan Thanh Cong, Nguyen Quoc Viet Hung, and Jun Jo

*Griffith University, Gold Coast, Australia*

**Abstract**— Wildfires are getting more severe and destructive. Due to their fast-spreading nature, wildfires are often detected when already beyond control and consequently cause billion-scale effects in a very short time. Governments are looking for remote sensing methods for early wildfire detection, avoiding billion-dollar losses of damaged properties. The aim of this study was to develop an autonomous and intelligent system built on top of imagery data streams, which is available from around-the-clock satellites, to monitor and prevent fire hazards from becoming disasters. However, satellite data pose unique challenges for image processing techniques, including temporal dependencies across time steps, the complexity of spectral channels, and adversarial conditions such as cloud and illumination. In this paper, we propose a novel wildfire detection method that utilises satellite images in an advanced deep learning architecture for locating wildfires at pixel level. The detection outputs are further visualised in an interactive dashboard that allows wildfire mitigation specialists to deeply analyse regions of interest in the world-map. Our system is built and tested on the Geostationary Operational Environmental Satellites (GOES-16) streaming data source. Empirical evaluations show the superior performance of our approach over the baselines with 94% F1-score and 1.5 times faster detections as well as its robustness against different types of wildfires and adversarial conditions.

## I. INTRODUCTION

Wildfires are causing damage to both property and human life [1]. Due to the increase of drying fuels and extreme weather conditions [2], wildfires are getting more severe. Particularly, an annual cost of \$1.6 billion are destroyed by wildfires in the US and Australia, and this amount is estimated to be up to six times larger from 2018 to 2050 [3]. In addition, wildfires are a major factor for climate change and they result in the ecological imbalance of the Earth. Apart from the immediate damage, wildfires also cause long-term consequences such as local weather impacts, global warming, and extinction of rare species. Therefore, an early detection of forest fires can significantly reduce unexpected damage to both property and the environment [3], contribute to global environmental stability. Wildfire research becomes an interdisciplinary area involving not only ecologists but also computer scientists to develop algorithmic methods for wildfire detection [4].

So far, various strategies have been continuously conducted in order to early and automatically detect wildfires [5]. For example, the first threshold-based algorithm is FIMMA, which is developed for AVHRR system [6]. It focuses on minimise false alarms by solving the nighttime detection problem. However, it has a low temporal resolution (only 3 times per day) and only applicable over

forested regions. GOES-AFP [7] is developed for GOES-11 and GOES12 data, using masking mechanisms to remove clouds and distinguish water-land complex. Nevertheless, it greedily focuses on recall and thus introduces more false alarms. Another threshold-based fire detection algorithm (aka active fire product) proposed for MODIS [8] is used to address false alarms cause by small forest clearings and the omission of large fires obscured by thick smoke. Still, it also has low temporal frequency (4 shots per day). VIIRS-AFP [9] is built on top of the previous algorithm MODIS-Terra using a contextual approach to detect both day and nighttime biomass burning and other thermal anomalies. However, it uses only a few spectral bands. To sum up, current methods have either not reached competitive accuracy or they can monitor the wildfires in a very low temporal frequency.

Deep learning, in particular deep neural networks, has been successfully applied in solving challenging tasks [10], such as classification [11], object detection [12], [13], and semantic segmentation [14]. Due to these successes, various studies on remote sensing recently deployed deep learning methods on satellite images for land use classification and urban planning [15], [16]. To our knowledge, deep learning models have not been applied for wildfire detection using satellite images. It comes from the fact that the old-generation wildfire satellites often have low temporal, spatial or spectral resolution which prevent accurate and robust detections on small area (e.g.  $1km^2$ ). The new-generation GOES-16 satellite image data overcomes all those limitations, paving the way for new applications of deep learning methods [17], [18].

Filling this gap, our work uses multispectral images, which is available from around-the-clock satellites, to detect wildfires in a timely manner. Environmental satellites offer not only a wide spatial range but also high availability and timeliness [8]. We leverage the advances of deep learning to enable accurate and robust wildfire detection for monitoring and mitigating applications. The multi-layered nature of deep learning architectures, in particular deep neural networks, will enable to capture multispectral information in both spatial and temporal dimensions [19].

In this work, we present an autonomous and intelligent framework for wildfire detection. We observed several challenges. First, the feature maps of spectral images are sensitive to different spectral bands, different cloud coverage during a day, and different illumination conditions (daytime and nighttime). Second, there are various noises (missing values,

measurement errors) in the streams of satellite data, making it difficult to monitor the behaviour of active wildfires.

The contributions of our work are summarised as follows.

- We establish a novel *streaming data processing pipeline* to collect data streams of satellite, impute missing values, and scrutinise data by regions of interest for real-time wildfire monitoring systems.
- We propose a novel *early wildfire prediction* model by designing a temporal-aware spectral-spatial deep learning architecture that (i) utilise satellite data by pairwise fusion layers, (ii) capture spatial patterns across multiple spectral bands by 3D convolutional layers, and (iii) locate wildfires at the pixel level by patch-wise classification
- We deploy a novel *streaming data visualisation* dashboard to support wildfire mitigation specialists in monitoring, identifying, and pro-acting to wildfires.

The remain of our paper is structured as follows. §II provides an overview of our approach. We then discuss the problem and the detailed solution for each component of our framework, including *streaming data processing* (§III), *early wildfire prediction* (§IV), and *streaming data visualisation* (§V). Finally, §VI presents empirical evaluations and §VII concludes the paper.

## II. APPROACH OVERVIEW

Fig. 1 presents an overview of our framework. The input of our framework is the data streams of satellite sources. Raw satellite images are cleaned by *Streaming Data Processing* component and are transformed into vector formats. Next, the cleaned, ordered, and align satellite data are put through the *Early Wildfire Prediction* component to detect and locate wildfires at pixel-level. Finally, the whole detection inputs and outputs are composed into a *Streaming Data Visualisation* dashboard, thereby completing an autonomous and intelligent wildfire detection system. In the following, we describe the detailed roles and contributions of each component.

**Streaming Data Processing.** The raw data streams from GOES-16 satellites are hosted by cloud services (AWS or Google Cloud). These data sources are fed into our component via Web API temporally. The details are described in §III.

**Early Wildfire Prediction.** Remote prediction of wildfires from satellite images is a challenging task due to several factors: (i) satellite images are hindered by adversarial conditions such as cloud and rain, (ii) spatial patterns of wildfires, and (iii) brightness patterns of wildfires across different spectral bands. In this component, we will leverage the multi-layered nature of deep learning architectures, including 3D convolutional layers and patch-wise classification to overcome these factors. The detail is described in §IV.

**Streaming Data Visualisation.** Most of existing monitoring systems do not offer localisation of wildfires due to lack of alignment between geographic coordinates and pixel-level

detection outputs. With available spatial and temporal information in the previous component, we design a streaming data visualisation dashboard to present fire-positive pixels of satellite images on the world-map. The detail is described in §V.

## III. STREAMING DATA PROCESSING

**Query by regions of interest.** Images provided by GOES-16 covers the whole America continent. A scrutinisation query is needed to focus on a particular region of interest  $R = (x_1, y_1, x_2, y_2, r)$ , where  $x_i$  and  $y_i$  are latitudes and longitudes; and  $r$  is the spatial resolution (size of each pixel) and can be varied between 0.5–2km. Our pilot experiments show that the 1km resolution works best for our prediction.

The requirement of a ROI query is that the distance between each pixels should be similar to real-world geo-distance since it plays an important role to detect wildfires accurately as well as to integrate with other data domains such as weather measurements. Therefore, we use Lambert projection instead of other popular projections such as Mercator (used in Google Map) or Cube (used in ArcGlobe). Lambert projection is also known as conic map projection and commonly used in aeronautical charts and national/regional mapping systems [20].

The ROI query is processed by the SatPy library, which is compatible with data streams, netCDF format, Lambert projection, and preserves meta-data (timestamp, size, etc.) during the resampling process of scrutinising the original data samples (big view) into different ROIs uniformly.

**Data storage.** The processed images by ROI query and missing-data query are stored in PNG format (each band has a PNG file). The radiance values of spectral bands are also kept in CSV format (Fig. 2). While the PNG format is used for visual analytics such as qualitative validation and visualisation, the CSV format is used as input for the prediction model.

## IV. EARLY WILDFIRE PREDICTION

### A. Problem Statement

Let  $S = \{1, \dots, n, \dots, N\}$  be a set of integers indexing  $N$  pixels of a multispectral image, and let  $L = \{1, 2\}$  be a set of integers indexing 2 classes: *fire* and *no-fire*. The image  $X = \{x_1, \dots, x_n, \dots, x_N\}$  is a set composed of  $N$  feature vectors, where the  $M$ -dimensional feature vector  $x_n = \{x_n^1, \dots, x_n^M\}$  corresponds to the  $n$ -th pixel [21]. Here ABI images have 16 spectral bands, thus  $M = 16$ .

**Problem 1** (Pixel-wise classification). *Pixel-wise classification is a mapping process of a many-to-many function  $f : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{N \times 2}$  from an image of pixels  $X$  to the set of label vectors  $Y$  corresponding to the input pixels and a single label vector  $y$ .*

$$Y = f(X) \quad (1)$$

where the label  $Y = \{y_1, \dots, y_n, \dots, y_N\}$  is represented as a set composed of  $N$  label vectors corresponding to  $N$  pixels. The label  $y_n = \{y_n^1, y_n^2\}$  is 2-dimensional label

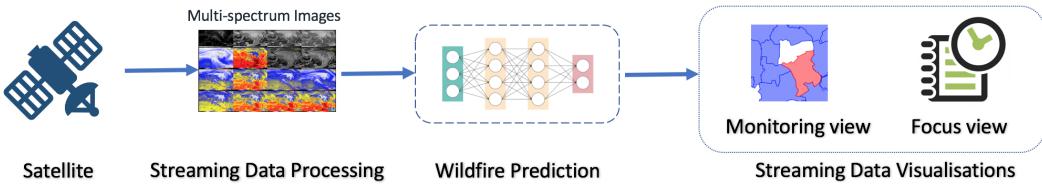


Fig. 1: Overview of the proposed wildfire detection system.

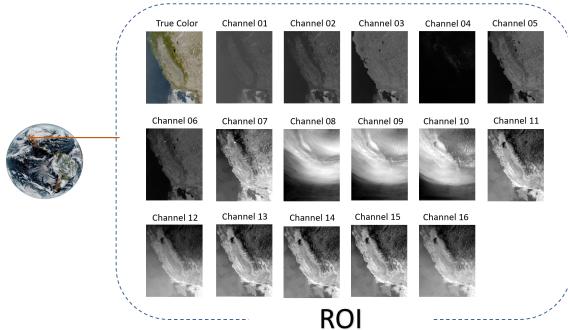


Fig. 2: 16-channel spectral images of a region of interest

vector, where the elements  $y_n^l$  represents the possibility that the pixel  $x_n$  and the whole image  $X$  belong to the class  $l \in L$ , respectively. The final label  $y_n^*$  for each pixel  $x_n$  will be decided by the maximum score:  $y_n^* = \text{argmax}_{l \in L} y_n^l$ .

GOES-16 satellite images offer high temporal, spatial or spectral resolution, which precedes old generation satellite data. Due to these data characteristics, we argue that designing a mapping process needs to satisfy the following requirements:

- (R1) *Spatial dependency*: consider the spatial context such as neighbouring pixels of a given pixel [22]. This is because a wildfire spreads via near-by location.
- (R2) *Cross-band spectral patterns*: handles the correlation across spectral bands [23]. This is because the spectral bands are sensitive to various factors such as rotations of the imagery sensor, atmospheric scattering condition, and illumination condition.

Traditional machine learning models, such as fully-connected networks in which each neuron is connected to every neuron in consecutive layers, is unable to take advantage of local spatial patterns and multi-channel dependencies [24]. We leverage the advances of deep learning architectures, in which the features are generalised at multiple granularity thanks to a long sequence of small layers [23], [25], [26].

### B. Model Structure

We propose a deep neural network architecture [24] that integrates spectral and spatial information at the same time. The network features multiple modules (sub-networks): (i) *input module* – feeds the imagery data and weather information to succeeding layers, (ii) *convolutional module* – processes spatial and spectral dependencies by data pixels, (iii)

*output module* – returns the prediction result. An overview of the network can be found in Fig. 3.

1) *Input Module*: The input of the network consists of  $m \times n$  pixel matrices for each spectral band and for each weather measurement (each pixel reflects a real geolocation), where  $m \times n$  is the size of ROI processed by the aforementioned *Streaming Data Processing* component. Since we want to find the label for every pixel (Prob. 1), traditional image-wise setting is not applicable. Moreover, if using the original images directly, the training set would have a low number of data samples, making the network prone to overfit.

**Patch Normalisation.** To tackle these issues, we propose a patch normalisation layer to augment the capability of training data as well as satisfy the pixel-wise classification output. First, the image is divided in patches of  $k_p \times k_p$  pixels ( $k_p = 12$  in the experiments as the minimal burned area of fire datasets is  $\approx 144\text{km}^2$ ) with 50% overlap. Then, to ensure that the network is trained on image patches with the same domain, a normalisation is performed (separately for each spectral band) by subtracting the average value to each pixel value of the image [27].

**Pixel-wise information.** To preserve the spatial connections between patches within the same image, we also take the coordinates of the center pixel of each image patch as input to the end (fully connected layer) of convolutional module. This hints that the output of the model will indicate the score of pixels being fire-positive or not [28].

It is worth noting that we do not use the upsampling approach of semantic segmentation setting [29] for pixel-wise classification since satellite images, unlike natural images, do not involve well-defined objects such as animals. Moreover and the resulting output of upsampling layer would lose original spectral information for visualisation.

2) *Convolutional Module*: The convolution module is a sub-network designed to capture spatial dependency (R1) while simultaneously reducing the complexity of fully connected networks [24]. It involves two operations: convolution and flatten.

**3D convolutional layers.** The input is fed through three convolutional layers, in which the neurons of one layer are only connected to a few neurons of another layer within a receptive field [30]. Convolutional layers become smaller when they are at deeper level to extract more concise and abstract features. This allows the model to focus on local spatial dependencies between pixels regardless of their

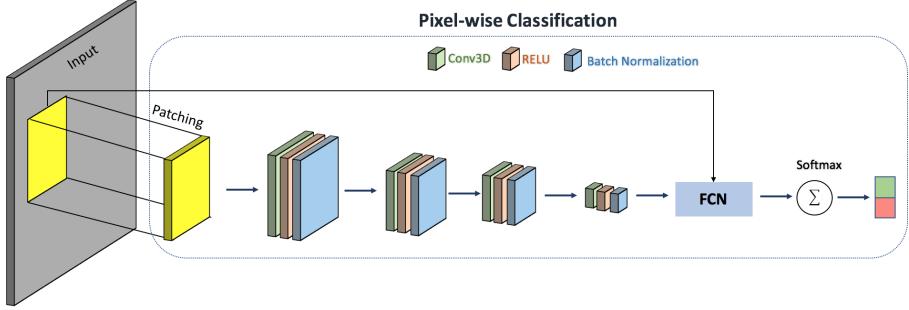


Fig. 3: Spatio-spectral deep neural network for early wildfire prediction

actual location in a 2D image, compared to fully-connected networks.

Different from traditional convolutional neural networks for 2D images [31], spectral images has an additional dimension of spectral bands with partial dependencies between them (R3). For this reason, we design a 3D version of convolutional layers, in which each neuron in the succeeding layer is connected only to a cube of neurons in the previous layer (i.e. 3D receptive field). The receptive field is moved across the entire input representation, and each neuron in the succeeding layer captures both spatial dependency (R1) and spectral dependency (R3) of the previous layer. The size of a receptive field  $n_c \times n_c \times n_c$  is a hyperparameter.

Convolutional networks are more computational-efficient than fully-connected ones by the weight-sharing mechanism, in which the value of the receiving neurons in the same layer share the same weights and biases in their weighted sum formation from the observed neurons in the receptive field [24]:

$$v_i = \varphi_c(b_i + \mathbf{w}_i * \mathbf{x}) \quad (2)$$

where  $*$  is the 3D convolution operation, the bias is added component-wise.  $v_i$  is the resulting values of applying  $i$ -th filter,  $\varphi$  is the neural activation function (e.g. RELU),  $b_i$  is the shared overall bias of the filter  $i$ ,  $\mathbf{w}_i = [w_{i1}, \dots, w_{ik_c}]$  is vector with shared weight and  $\mathbf{x}_j = [x_j, \dots, x_{j+n_c^3-1}]$  is the receptive field. In other words, the next layer extracts a local spatial feature from the previous layer, coined the term feature map [24].

Since there could be different types of spatial features, we need  $n_{filters}$  different filters, which results in  $n_{filters}$  feature maps. The first convolutional layer has a kernel size of  $n_{c,1} = 7$  and outputs  $n_{filters,1} = 128$  features. The second convolution layer will extract more abstract features based on the features extracted in the first layer, using a kernel size of  $n_{c,2} = 5$  and outputs  $n_{filters,2} = 64$  features. The third and final convolution layer has  $n_{c,3} = 3$  and outputs  $n_{filters,2} = 32$ . This comes from the fact that a wildfire is often bigger than a single pixel ( $1km^2$ ) of a satellite image [3].

**Flatten.** The last convolutional layer is flattened and connected to a fully-connected layer in order to project the feature space into classification score. Formally, the projection

layer consists of  $n_p$  neurons, each of which is connected to all the feature maps of the last convolutional layer. The output is a vector  $v_p \in \mathbb{R}^{n_p}$  with the  $k$ -th value computed as:

$$v_p^k = \varphi_p(\mathbf{w}_p^k v + b_p^k) \quad (3)$$

where  $\mathbf{w}^k$  and  $b^k$  are parameters and  $v$  is the value of the last convolutional layer.

3) *Pixel-wise Output Module:* The output module consists of a projection layer to transform the fusion vector into a scoring vector with the same dimension as the dimension of class label (which is 2) and a soft-max layer to normalise the scores for comparison. Formally, the projection layer is basically a linear mapping:

$$y' = \mathbf{w}v + b \quad (4)$$

where  $y' = [y'_1, \dots, y'_C]$  is the scoring vector. The soft-max layer is an activation function to non-linearise the projection:

$$y_c = \frac{\exp y'_c}{\sum_{c' \in C} \exp(Y'_{c'})} \quad (5)$$

We use the cross-entropy loss to train the network, which compares the final scoring  $y_c$  and the true label of each pixel.

### C. Training the model

**Design choices.** In order to obtain the deep learning architecture in Fig. 3, we experimented with different depths (e.g. number of convolutional layers), different patch size, different convolutional kernel sizes, and different number of convolutional filters different filter. For example, we experimented with 2D convolutional layers only but this worsened the classification results, which is indicative of spatial dependencies across spectral bands. In the end, the convolutional module has a depth of 3 layers to balance between the predictability power and the running time.

**Avoid overfitting.** Available training data for domain-specific applications such as wildfire detection is scarce due to requirement of expert knowledge (to label fire-positive pixels) and real-world observations (fires actually happened). This could lead to overfitting as the model could be too complex to be fit on a small amount of training samples.

To alleviate this problem, we experimented with several strategies:

- *Semantic-preserving augmentation*: Since the image classification is rotation invariant, e.g., users can investigate fire-positive images from different orientations without altering the decision. Modifications such as rotation and mirroring help to increase training data without compromising label quality. Formally, we transform each data sample into eight different samples by combining  $k \times \pi/2$  rotations, with  $k = 0..3$ , and vertical reflections. Each of modified samples is considered to have the same class label as the original sample [27].
- *Regularisation*: We put several pooling, batch normalisation, and drop-out layers in between consecutive layers in the network to avoid overfitting. For example, max-pool layer is used between two consecutive convolutional layers. Batch normalisation is used before and after fully-connected layer of the convolutional module.
- *Temporal information*: We use satellite images in different time points as training data, increasing input variation to avoid model overfitting.

**Parameter optimisation.** We trained the network using Adaptive Moment Estimation (Adam) optimizer. The final network requires fine-tuning of 3 hyperparameters: the learning rate  $\eta$  and the momentum coefficient  $\mu$  and the regularization parameter  $\lambda$  for the  $L_2$  weight regularization layers.

## V. STREAMING DATA VISUALISATION

### A. Dashboard Design

Through integrating and codifying satellite images and weather data, a visualisation dashboard is developed on top of the proposed model to autonomously predict and monitor wildfires for regions of interest in real-life. It offers two main types of visualisation: *monitoring view* and *focus view*. While the former gives an overview status of ROIs on the whole world-map, the latter has the ability to zoom-in a particular hot spot and monitor it over time. Besides, weather data and satellite images are also provided at the selected moment to help users examine the accuracy of the prediction model.

### B. Monitoring View

In the monitoring view, the system shows an instant overview of the selected region of interest. User can switch between ROIs by using the drop-down toolbox in the left panel. The hot-spots are highlighted automatically by the system, allow users to easily access those locations for investigation. There is also a bar chart which gives the latest predictions and other supporting information (long-lat coordinates, last visited time, the proportion of fire) for a target location. If users click on any point in the graphical figure, all of the above information will be presented according to the new position. This feature helps user to investigate any suspicious area at their convenience even alarms have not been raised.

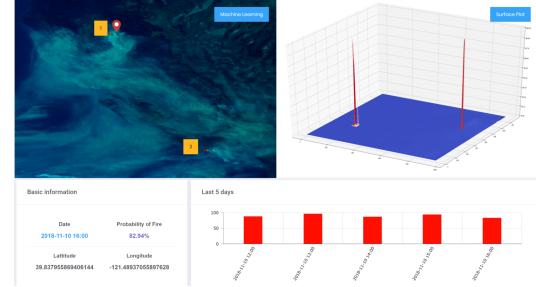


Fig. 4: Monitoring View

### C. Focus View

In the focus view, one location is more focused. The bar chart not only gives the timeline for the proportion of wildfires for a target point but also contains integrated weather data for that location. Moreover, their spread inside the ROIs is also presented visually in the map. Combining with the zoom feature of the maps, users can easily follow the growth of a hot-spot from the beginning until its end. Other supporting information like image per time, longitude, latitude and captured time are also provided for cross-checking purposes.

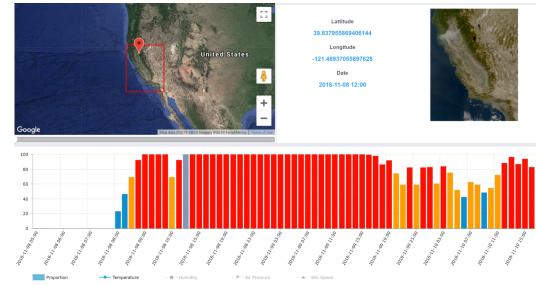


Fig. 5: Focus View

### D. Benefits of the software application

The developed data visualisations help users to monitor and detect wildfires granularly and seamlessly. Combining all the components of detection framework, including streaming data processing and early wildfire prediction, as a whole, our visualisation dashboard performs an end-to-end process from raw satellite streams to wildfire prediction outputs in near real-time. Our current prototype will pave the way for online applications used by governments and organisations to timely prevent the spread of wildfires such as connecting with urban emergency systems in the target ROI.

## VI. EMPIRICAL EVALUATION

In this section, we conduct experiments with the aim of answering the following research questions:

- (RQ1) Does our proposed framework outperform the baseline methods of spectral image classification for wildfire detection?
- (RQ2) Is the model robust to noises such as cloud effects, missing values, outliers, and ambient background (e.g. day or night)?

(RQ3) Are the model outcomes interpretable?

In the remaining of the section, we first describe our experimental settings (§VI-A). Then we present our empirical evaluations to verify the above research questions, including end-to-end effectiveness test (§VI-B), robustness to adversarial conditions (§VI-C), and qualitative showcases (§VI-D).

### A. Setup

**Datasets.** We study real-world wildfire datasets happened in different seasons and different regions in USA, allowing to examine the flexibility of our model. Key characteristics of the datasets are described in Table I.

- **UTE Park Fire:** The Ute Park Fire start after 2pm local time on May 31, 2018. It was reported burning on the east of Ute Park, New Mexico in the United States. The fire burned a total of about  $149 \text{ km}^2$  [32].
- **Ferguson Fire:** The Ferguson Fire is a wildfire in the Sierra National Forest, Stanislaus National Forest and Yosemite National Park in California in the USA. The fire was reported on July 13, 2018 and burned above  $392 \text{ km}^2$  [33].
- **Carr Fire:** The Carr Fire happened in Shasta and Trinity Counties, California, United States. The fire started at about 1pm local time of July 23, 2018 and burned above  $929 \text{ km}^2$  [34].
- **Camp Fire:** The Camp Fire is one of the deadliest wildfire in California. The fire started on November 8, 2018 in Butte Country in Northern California and covered an area of  $620 \text{ km}^2$  [35].

TABLE I: Statistics of experimental datasets

| Dataset  | Genesis (2018) | #Img | #Img of Fires | #Pixels          | Distribution <sup>1</sup> |
|----------|----------------|------|---------------|------------------|---------------------------|
| UTE Park | 14:00 31.05    | 48   | 25            | $181 \times 222$ | 1927560 : 1176            |
| Ferguson | 21:36 13.07    | 48   | 48            | $252 \times 247$ | 2986943 : 769             |
| Carr     | 13:00 23.07    | 48   | 40            | $414 \times 573$ | 11383759 : 2897           |
| Camp     | 06:33 08.11    | 72   | 55            | $290 \times 300$ | 6252876 : 11124           |

<sup>1</sup> Ratio between #non-fire pixels and #fire pixels

**Baselines.** The performance of our wildfire detection model is evaluated against representative baselines in the literature.

- **MODIS-Terra:** is the active fire product (MOD14/MYD14) algorithm of MODIS satellite system. It detects fire pixels by thresholding on background characteristics, via masking mechanisms to identify cloud, land, and water. False alarms are further reduced by regression tests such as desert boundary rejection and land-pixel coastal rejection [8].
- **AVHRR-FIMMA:** is the active fire mapping algorithm used in near real-time AVHRR imagery data [6]. It uses only three spectral channels (3,4,5) to eliminate clouds, isolate potential pixels from warm background, bright scenes, and non-forest scene. Then, only potential pixels occurring within the land type with some tree cover are regarded as fire-positive (urban or agriculture lands are not considered).
- **VIIRS-AFP:** is the active fire detection algorithm used in VIIRS satellite system. It is built on top of MODIS-

Terra algorithm to customise for both day-time and night-time detections [9].

- **GOES-AFP:** is the state-of-the-art detection method [7] developed for GOES-11 and GOES-12 data. It uses new masking mechanisms to remove clouds and distinguish water-land complex. However, it greedily focuses on recall and thus introduces more false alarms.

We deliberately omit existing object detection methods [36] from our evaluation, since they are specialized in well-defined shapes, whereas wildfires' are arbitrary.

**Metrics.** A number of standard statistical measures are employed for evaluating deep learning models, including: (1) *Precision* – the number of true positive data samples (i.e. classified correctly as fire-positive) divided by the number of positively classified data samples; (2) *Recall* – the number of true positive data samples divided by the number of fire-positive ground truth; 3) *Weighted F1-score* – a harmonic mean of *Precision* and *Recall*, calculated for each class and the average is weighted by the number of true instances in each class. *Weighted F1-score*, like *Accuracy* (ratio of correctly classified instances over the total number of instances), reflects the capability of identifying true positives as well as avoiding false positives; but is more useful in case of imbalance class distribution (e.g. true positives are more important than true negatives) [37].

In the context of wildfire detection, we measure *Lag time to detection*, which is the time difference between the genesis of a wildfire and its detection (i.e., the first data sample is labelled as fire-positive).

**Training procedure.** We use  $k$ -fold cross validation to ensure fairness in splitting the data into training set and test set.  $k = 10$  is commonly used in practice to achieve a good trade-off between having enough data for training and having enough unseen samples for a fair evaluation. To avoid over-fitting, the training data is further randomly split into a learning set (consisting of  $k - 2$  subsets) and a tuning set (1 subset). The model is then trained only on the learning set and the tuning set is used as a reference performance. It allows an optimal setting to be chosen where model is guaranteed to perform well on previously unseen data (via tuning set) and hence prevents over-fitting when it comes to the test set. In sum, the labelled data is divided into 80% for training, 10% for validating and 10% for testing.

**Hyperparameter tuning.** The aforementioned three hyperparameters needs to be estimated for training. These include the learning rate  $\eta$ , the momentum coefficient,  $\mu$  and the regularization parameter  $\lambda$ , which are calculated using a random search technique as proposed by [24]. This method is reported to be more efficient for hyperparameters optimization than a traditional grid search.

**Reproducibility environment.** The model is implemented in Python v3.6 using Keras API (v2.2.4), a high-level neural networks library focusing on efficiency. Our model was trained and tested on GPU GeForce GTX 108, CPU AMD Ryzen Threadripper 1900X 8-Core Processor and 62 GB

RAM. Due to space limitation, further evaluations such as running time will be reported in our extended paper.

### B. End-to-end Comparisons

In this experiment, we will answer (RQQ) by comparing our model against different baselines, including MODIS-Terra and VIIRS-AFP. The comparisons between all detection methods summarised in Table II, whose results are averaged over all datasets.

TABLE II: End to end comparisons

|                  | Precision     | Recall        | Weighted F1   | Lag time(h) |
|------------------|---------------|---------------|---------------|-------------|
| MODIS-Terra      | 94.87%        | 88.61%        | 91.60%        | 3.8         |
| AVHRR-FIMMA      | 79.03%        | 80.01%        | 79.52%        | 11.2        |
| VIIRS-AFP        | 94.62%        | 86.12%        | 90.12%        | 5.8         |
| GOES-AFP         | 83.92%        | 89.23%        | 86.52%        | 4.3         |
| <b>Our model</b> | <b>96.05%</b> | <b>91.87%</b> | <b>93.89%</b> | <b>2.6</b>  |

It can be observed that our model performs better than all the baselines with at least 2.5% more accurate (F1-score) and 1.5 times earlier (lag time). Another key finding is that all methods have a large lag time in Ferguson dataset due to extremely adversarial conditions: the fire had small size and happened in night-time. An interesting observation is that the recall of GOES-AFP is often higher than its precision, due to its greedy design in capturing all potential wildfire signals with the sacrifice with false alarms [7].

### C. Robustness to Adversarial Conditions

This set of experiments validate (RQ2) regarding the robustness of our model against adversarial conditions.

**Day vs. night.** We compare the overlapping of the detected region of pixels and the actual fire-positive region of pixels in different times of the day: day-time and night-time. The result is depicted in Fig. 6, in which our model outperforms the best baseline (MODIS, due to aforementioned experiments) across all datasets. An interesting finding is that the performance of our model degrades in night-time but still  $\geq 0.8$ . The Ferguson dataset suffers the smallest performance of wildfire detection since it was a small fire happened at midnight, which still poses challenge for all detection methods.

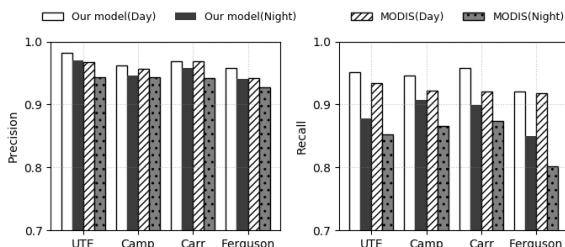


Fig. 6: Wildfire detection in day-time vs. night-time

**Cloud effects.** We divide the dataset into images with cloud and images without cloud. We measure the performance of

pixel-wise classification since clouds can alter the neighbourhood effect of image pixels. The result is presented in Fig. 7, in which MODIS is used as representative of all baselines against our model due to its superior performance (see experiments above). A key finding is that all the baselines (not only MODIS) do not consider cloudy images and have no detection (performance = 0). Another interesting observation is that although our model degrades with the presence of clouds, the overall performance is still acceptable for all datasets ( $\geq 0.8$ ). It is worth noting that in UTE dataset, there is no cloud image and thus the measurement is not applicable (N/A).

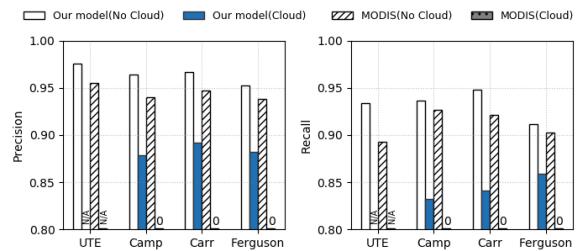


Fig. 7: Wildfire detection in cloudy conditions

### D. Qualitative showcases

We answer the (RQ3) regarding the interpretability of our prediction model using the developed visualisation dashboard.

**Showcase on Camp Fire dataset.** We compare our prediction model against MODIS-Terra (which is chosen for showcase due to its superior performance against other baselines above) on Camp Fire dataset. Fig. 8 shows the visualisation of MODIS-Terra on: (a) the true color image captured by MODIS on November, 8th 2018, (b) Terra prediction at 14:00 UTC, (c) Terra prediction at 20:00 UTC. The visualisation is only shown at these two moments since the Terra platform only acquires MODIS data twice per day at mid-latitudes [38]. Although its result is quite good at 20:00 UTC, the fire, in fact, already started at about 14:00 UTC, resulting in a 6-hour lag time.

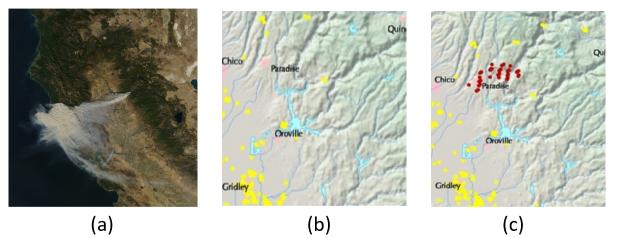
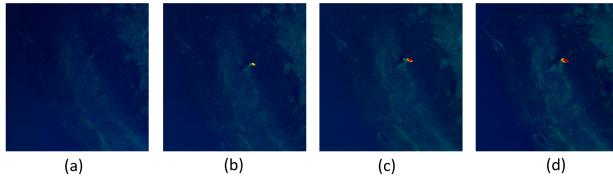


Fig. 8: Camp Fire Prediction by MODIS. (a) True color Terra satellite image. (b) 14:00. (c) 20:00.

Fig. 9 shows the visualisation of our prediction model on the same event. In contrast with MODIS, our model on top of GOES-16 satellite shows its temporal advantage.

The visualisation is captured every 15min from 15:15 UTC to 16:00 UTC. Our model can detect the fire from 15:30 UTC, resulting in 1.5 hour lag time, resulting in a significant improvement over MODIS-Terra. Moreover, with the short alarm interval (15min), users can monitor the growth of fire in near real-time.



**Fig. 9: Camp Fire prediction by GOES-16 satellite. (a) 15:15. (b) 15:30. (c) 15:45. (d) 16:00.**

## VII. CONCLUSIONS

In this paper, we propose a novel remote wildfire detection framework using streams of satellite images. The framework consists of (i) a *streaming data processing* component to clean-up and scrutinise raw image data for regions of interest, (ii) an *early wildfire prediction* component using deep learning architectures to capture spatial and spectral patterns for more accurate and robust detection, (iv) a *streaming data visualisation* dashboard for wildfire mitigation specialists to receive timely alerts of potential wildfires. The empirical evaluations highlight that our techniques outperform the baselines with 94% F1-score and  $1.5\times$  faster detections.

## REFERENCES

- [1] B. Teague, R. McLeod, and S. Pascoe, “Victorian bushfires royal commission final report,” *Melbourne: Parliament of Victoria*, 2009.
- [2] D. J. Wuebbles, D. W. Fahey, and K. A. Hibbard, “Climate science special report: fourth national climate assessment, volume i;” *US Global Change Research Program*, 2017.
- [3] USGCRP, “Impacts, risks, and adaptation in the united states: Fourth national climate assessment, volume ii,” *US Global Change Research Program*, 2018.
- [4] P. E. Dennison, D. A. Roberts, and L. Kammer, “Wildfire detection for retrieving fire temperature from hyperspectral data,” *Journal of Scientific and Engineering Research*, vol. 4, no. 7, pp. 126–133, 2017.
- [5] V. Slavkovikj, S. Verstockt, S. Van Hoecke, and R. Van de Walle, “Review of wildfire detection using social media,” *Fire safety journal*, vol. 68, pp. 109–118, 2014.
- [6] Z. Li, Y. J. Kaufman, C. Ichoku, R. Fraser, A. Trishchenko, L. Giglio, J. Jin, and X. Yu, “A review of avhrr-based active fire detection algorithms: Principles, limitations, and recommendations,” *Global and regional vegetation fire monitoring from space, planning and coordinated international effort*, pp. 199–225, 2001.
- [7] W. Xu, M. Wooster, G. Roberts, and P. Freeborn, “New goes imager algorithms for cloud and active fire detection and fire radiative power assessment across north, south and central america,” *Remote Sensing of Environment*, vol. 114, no. 9, pp. 1876–1895, 2010.
- [8] L. Giglio, W. Schroeder, and C. O. Justice, “The collection 6 modis active fire detection algorithm and fire products,” *Remote Sensing of Environment*, vol. 178, pp. 31–41, 2016.
- [9] W. Schroeder, P. Oliva, L. Giglio, and I. A. Csizar, “The new viirs 375 m active fire detection data product: Algorithm description and initial assessment,” *Remote Sensing of Environment*, vol. 143, pp. 85–96, 2014.
- [10] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, “An evaluation of aggregation techniques in crowdsourcing,” in *WISE*. Springer, 2013, pp. 1–15.
- [11] M. Rashid, N. Sulaiman, M. Mustafa, S. Khatun, and B. S. Bari, “The classification of eeg signal using different machine learning techniques for bci application,” in *RiTA*. Springer, 2018, pp. 207–221.
- [12] S. Ren, Y. Zhou, and L. He, “Multi-object tracking with pre-classified detection,” in *RiTA*. Springer, 2017, pp. 503–513.
- [13] X. Le, J. Jo, S. Youngbo, and D. Stantic, “Detection and classification of vehicle types from moving backgrounds,” in *RiTA*. Springer, 2017, pp. 491–502.
- [14] M. Z. Nezhad, N. Sadati, K. Yang, and D. Zhu, “A deep active survival analysis approach for precision treatment recommendations: Application of prostate cancer,” *ESWA*, vol. 115, pp. 16–26, 2019.
- [15] A. Albert, J. Kaur, and M. C. Gonzalez, “Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale,” in *KDD*, 2017, pp. 1357–1366.
- [16] T. Oh, M. J. Chung, and H. Myung, “Accurate localization in urban environments using fault detection of gps and multi-sensor fusion,” in *RiTA*. Springer, 2017, pp. 43–53.
- [17] N. Q. V. Hung, D. C. Thang, M. Weidlich, and K. Aberer, “Minimizing efforts in validating crowd answers,” in *ACM SIGMOD*. ACM, 2015, pp. 999–1014.
- [18] N. Q. V. Hung, H. Jeung, and K. Aberer, “An evaluation of model-based approaches to sensor data compression,” *IEEE TKDE*, vol. 25, no. 11, pp. 2434–2447, 2012.
- [19] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, “Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review,” *IJAC*, vol. 14, no. 5, pp. 503–519, 2017.
- [20] Lambert azimuthal equal-area projection. [https://en.wikipedia.org/wiki/Lambert\\_azimuthal\\_equal-area\\_projection](https://en.wikipedia.org/wiki/Lambert_azimuthal_equal-area_projection), visited 2019.
- [21] P. Liu, H. Zhang, and K. B. Eom, “Active deep learning for classification of hyperspectral images,” *IEEE J-STARS*, vol. 10, no. 2, pp. 712–724, 2017.
- [22] B. Hally, L. Wallace, K. Reinke, S. Jones, and A. Skidmore, “Advances in active fire detection using a multi-temporal method for next-generation geostationary satellite data,” *IJDE*, pp. 1–16, 2018.
- [23] J. Yue, W. Zhao, S. Mao, and H. Liu, “Spectral-spatial classification of hyperspectral images using deep convolutional neural networks,” *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, 2015.
- [24] R. S. Andersen, A. Peimankar, and S. Puthusserypady, “A deep learning approach for real-time detection of atrial fibrillation,” *ESWA*, vol. 115, pp. 465–473, 2019.
- [25] N. Q. V. Hung, H. H. Viet, N. T. Tam, M. Weidlich, H. Yin, and X. Zhou, “Computing crowd consensus with partial agreement,” *IEEE TKDE*, vol. 30, no. 1, pp. 1–14, 2017.
- [26] N. Q. Hung, D. C. Thang, N. T. Tam, M. Weidlich, K. Aberer, H. Yin, and X. Zhou, “Answer validation for generic crowdsourcing tasks with minimal efforts,” *VLDBJ*, vol. 26, no. 6, pp. 855–880, 2017.
- [27] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polónia, and A. Campilho, “Classification of breast cancer histology images using convolutional neural networks,” *PloS one*, vol. 12, no. 6, p. e0177544, 2017.
- [28] Y. Wang, Y. Qiu, T. Thai, K. Moore, H. Liu, and B. Zheng, “A two-step convolutional neural network based computer-aided detection scheme for automatically segmenting adipose tissue volume depicting on ct images,” *Computer methods and programs in biomedicine*, vol. 144, pp. 97–104, 2017.
- [29] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.
- [30] S. Thirumuruganathan, N. Tang, and M. Ouzzani, “Data curation with deep learning [vision]: Towards self driving data curation,” *arXiv preprint arXiv:1803.01384*, 2018.
- [31] S. Cha and T. Moon, “Udlr convolutional network for adaptive image denoiser,” in *RiTA*. Springer, 2018, pp. 55–61.
- [32] Ute park fire. [https://en.wikipedia.org/wiki/Ute\\_Park\\_Fire](https://en.wikipedia.org/wiki/Ute_Park_Fire), visited 2019.
- [33] Ferguson fire. [https://en.wikipedia.org/wiki/Ferguson\\_Fire](https://en.wikipedia.org/wiki/Ferguson_Fire), visited 2019.
- [34] Carr fire. [https://en.wikipedia.org/wiki/Carr\\_Fire](https://en.wikipedia.org/wiki/Carr_Fire), visited 2019.
- [35] Camp fire. [http://cdfdata.fire.ca.gov/incidents/incidents.details.info?incident\\_id=2277](http://cdfdata.fire.ca.gov/incidents/incidents.details.info?incident_id=2277), visited 2019.
- [36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *CVPR*, 2016.
- [37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.
- [38] Modis thermal anomalies/fire. <https://modis.gsfc.nasa.gov/data/dataprod/mod14.php>, visited 2019.