THE LARGE-SCALE WILDFIRE SPREAD PREDICTION USING A MULTI-KERNEL CONVOLUTIONAL NEURAL NETWORK

M. Marjani [1]*, M.S. Mesgari [1]

[1] Department of Geospatial Information Systems, Faculty of Geodesy and Geomatics Engineering, K.N. Toosi University of Technology, No. 1346, ValiAsr Street, Mirdamad Cross, Tehran, Iran 19967-15433- (Mohamad_marjani, mesgari)@email.kntu.ac.ir

Commission IV, WG IV/3

KEYWORDS: Wildfire, Multi-kernel CNN, Deep learning, Convolutional neural network, Forest fire, Machine learning

ABSTRACT:

In the last twenty years, destructive wildfires have affected the environment to the tune of billions of dollars. An accurate model is crucial for predicting the spreading of wildfires in a variety of conditions. In this study, a multi-kernel convolution neural network (CNN) deep learning model was proposed based on elevation, wind direction, and speed, minimum and maximum temperatures, humidity, precipitation, drought index, normalized difference vegetation index (NDVI), and energy release component to predict wildfire spread across the United States. Using multi-kernel CNN, it is possible to predict whether a pixel will be on fire at a future time. Compared to the model presented by other authors, the multi-kernel CNN model achieved high accuracy and F1 score. In comparison with CNNs without a multi-kernel mechanism and fixed kernel size, the proposed model predicted more accurate results based on the test data set. The multi-kernel CNN model reached an overall accuracy of 98.6 and F1 score of 70.97 on test data.

1. INTRODUCTION

United States wildfires have burned over 68,000,000 acres in the past decade. A significant economic impact resulted from this damage, which led to the repair of over $5.1 billion of infrastructural damage (Green, Kaiser et al. 2020). Moreover, wildfires in wilderness-urban interfaces pose a significant threat to the environment, physical safety, and public safety (Green, Kaiser et al. 2020).

To assist in preparing for and controlling wildfires, predictive models are increasingly vital. Forest management has become more successful as computing resources have advanced. In addition to being able to respond to wildfire events in real-time, accurate simulations of wildfires may also help inform best practices in forest management. Data-driven wildfire models have grown in popularity in the last ten years from fully physical models to models that incorporate artificial intelligence and cover an increa[1]sing number of fire events (Green, Kaiser et al. 2020).

The fundamental laws of physics and chemistry are the basis of traditional physical models. Diffusion, advection, radiation, etc. are all models of coupled dynamics. When these are presented as sets of coupled partial differential equations, numerical instability, computational complexity, and mechanistic problems can arise. For example, the US Forest Service's current models such as FARSITE require rich data input that may require field agents to document the ecology physically, (Finney 1998). Additionally, these types of models must be revalidated to be applied to other applications. In addition, (Ferragut, Asensio et al. 2007) have demonstrated problems with the numerical implementation of physical models. A physical model is a set of explicit features that are implied by data but not revised by data. To manage and control forest fires efficiently, many countries are developing databases and information systems related to forest fires today (Cheng and Wang 2008). For example China's State Forestry Administration (Zhang 2004), Canada Large Fire Database (CLFDB) (Burton, Parisien et al. 2008), and European Forest Fire Information System (EFFIS) (Yamak 2006).

2. RELATED WORKS

Recent works have used machine learning or evolutionary strategies to model wildfire dynamics to Calculate faster, more accurately, and more efficiently and take advantage of advancements in remote sensing technology. Wildfire spread predictions have improved through the use of historic fire data by (Zheng, Huang et al. 2017) and (Radke, Hessler et al. 2019). Based on satellite images and local weather patterns, (Radke, Hessler et al. 2019) proposed FireCast, an algorithm that predicts wildfire spread prediction over 24 hours. It is 20% more accurate than FARSITE. Based on artificial neural networks (ANNs), (Chetehouna, El Tabach et al. 2015) model fire behavior, including flame height and rate of spread. The best ANN architecture utilized five hidden neurons. Experimental results confirmed the accuracy of the models (Chetehouna, El Tabach et al. 2015). An agent-based model based on satellite images was used by (Ganapathi Subramanian and Crowley 2018) to learn forest fire spreading policies. In modeling wildfire spread over time, (Sønderby, Espeholt et al. 2020) of Google AI Group have done outstanding work. A study by (Liang, Zhang et al. 2019) examined backward-propagation neural networks, recurrent neural networks (RNN), and long short-term memory (LSTM) neural networks in Alberta, Canada, for predicting the wildfire scale. The highest accuracy results from the use of LSTM (90.9%). Weather, terrain, and fuel characteristics were used by (Hodges and Lattimer 2019) to predict fire spread. To model the spread of fire, convolutional neural networks were used. A CNN output shows whether a pixel is burned or not according to its probability. In comparison with FARSITE, the researchers achieved precision and sensitivity of 89% and 88% for reference six-hourly burn maps.

The main objective of this study is to develop a multi-kernel CNN model that can predict wildfire spread in large-scale observation data across the United States. To achieve the main goal of this research, there are some following sub-objectives:

* Corresponding author

https://doi.org/10.5194/isprs-annals-X-4-W1-2022-483-2023

(1) Evaluate the effectiveness of multi-kernel models to feature extraction. (2) Define proportionate hyper parameters for the model. (3) Compare proposed model results with state of art models.

3. MATERIAL AND METHODOLOGY

As part of developing a deep learning model to be able to predict wildfire spread, the appropriate variables must be gathered. Wildfire spread is highly dependent on the optimal variables set. A high-level view of the proposed deep learning model and the data used to train it is shown in figure 1. Each driver of the spread of forest fires is represented as an image channel in the proposed model. The model predicts a new image with a channel representing the likelihood of each pixel burning based on its previous training. Post-processing of the predicted image generates the future burn map.

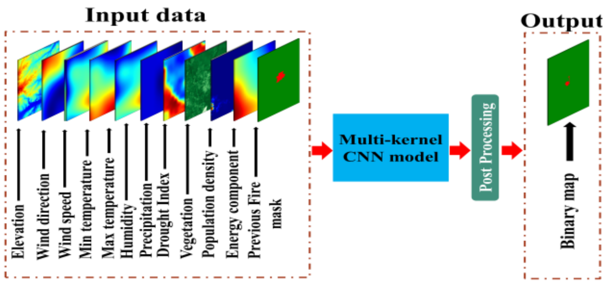

Figure 1. A high-level view of the data, model, and output.

3.1 Dataset

Remote sensing, computing resources, and machine learning have made it possible to develop more accurate methods for estimating wildfire spreading through data-driven methods (Huot, Hu et al. 2021). The next-day wildfire spread dataset was designed to analyze the potential of deep learning models for predicting wildfire spread based on observational data (Huot, Hu et al. 2021). The data was collected in different places and at different times when wildfires erupted throughout the US. Data were extracted as 64km × 64km regions at 1km resolution to capture the typical size of active fires. Using data in Google Earth Engine (GEE), they generated a fire mask for each region that showed the location of fires and no fires, as well as an additional class for uncertain labels. But in this study data which contained uncertainty in the label and the fire mask at time t-1 have been ignored. A wide variety of data are included in the data set, including elevation, wind direction and speed, minimum and maximum temperatures, humidity, precipitation, drought index, normalized difference vegetation index (NDVI), and energy release component. Figure 2 shows some samples of dataset.

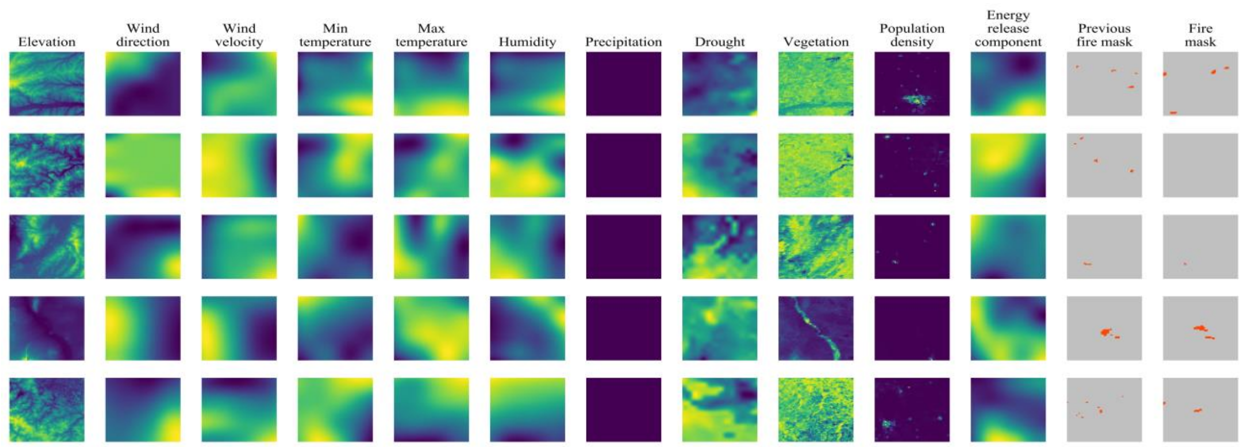

Figure 2. Examples from dataset. Each row corresponds to a fire event and the columns represent the different environmental variables. The label image on the last column indicates fire at time t which is going to be predicted.

3.2 Pre-processing

Pre-processing refers to the transformations applied to the data before feeding it to the algorithm. Features are not on the same scale; therefore, the model will give more weightage to the ones that have bigger values which is not the ideal scenario as the other features are important for building the model. To avoid this issue, Z-score standardization was performed for each feature individually according to equation (1) to quantify the variables and transform the original variable ranges into new ranges.

$$Z_i = \frac{x_i - mean_i}{\sigma_i} \tag{1}$$

where Z_i is the output value, x_i is the value of the variables, and mean represents the average value of the variable and σ_i is the standard deviation of the variable.

3.3 Multi-Kernel CNN

The construction of CNNs is based on multi-layer interconnected neural networks that combine low-level, intermediate-level, and high-level features (Mahdianpari, Salehi et al. 2018). CNN frameworks typically contain two main layers, the convolution layer, and the pooling layer. The CNN method extracts both spectral and spatial features, which makes it superior to the other deep learning algorithms for image datasets (Jamali, Mahdianpari et al. 2021). A summary of the proposed multi-kernel CNN model is presented in Table 1. This multi-kernel CNN model consists of 32 layers and 33 connections, where 15 convolutional layers are followed by a non-linear activation function of ReLU (Agarap 2018). As seen in table 1 and figure 3, there are five max-pooling layers followed by a concatenation layer in the proposed CNN model. The stride is set one by one, and the padding is the same for all the convolution layers. As shown in table 1 multi-kernel CNN model uses 5 different kernel sizes 3,5 and 7 in encoder blocks.

https://doi.org/10.5194/isprs-annals-X-4-W1-2022-483-2023 | © Author(s) 2023. CC BY 4.0 License.

The number of filters starts at 32 in block 1 and has been increased in the following blocks since the last encoder block uses 256 filters. In the encoder part, features are extracted and passed to the flattened layer to be inserted into a fully connected layer. Multi kernel CNN models in the decoder consist of three dense layers. In the first dense layer, there are 128 neurons, and in the second layer, 256 neurons, and the ReLU activation function is used in both of them. Dropout layers are used after the first two fully connected layers with a probability value of 0.3. The third and last dense layer contains 4096 neurons with sigmoid activation functions. 4096 neurons are required because labels are 64×64, and the sigmoid function is chosen to estimate burn probability for each pixel. There is a reshape layer at the end of the model which reshapes a vector of 4096 elements into a 64×64 tensor. In this study, the setting was determined empirically. Several objectives were pursued through the examination of various settings. To minimize the computation cost, it is most important to keep the number of layers and complexity low. Further, the proposed model should have an accuracy level comparable to deep CNNs.

3.4 Post-Processing

Due to the sigmoid function in the last layer of the model architecture, the output of the multi-kernel CNN model is a probability layer. A threshold value based on the probability of fire is applied to determine whether a pixel has been burned. Post-processing was carried out using an experimentally determined threshold. The threshold value in this study was 0.8.

Task	Block Number	Layer Type	Kernel Size	Filters
	B1	Convolution	(3,3)	16
	B1	Convolution	(5,5)	16
	B1	Convolution	(7,7)	16
	B1	Concatenation	~	~
	B1	Max pooling	~	~
	B2	Convolution	(3,3)	32
	B2	Convolution	(5,5)	32
	B2	Convolution	(7,7)	32
	B2	Concatenation	~	~
	B2	Max pooling	~	~
	B3	Convolution	(3,3)	64
	B3	Convolution	(5,5)	64
Encode	B3	Convolution	(7,7)	64
	B3	Concatenation	~	~
	B3	Max pooling	~	~
	B4	Convolution	(3,3)	128
	B4	Convolution	(5,5)	128
	B4	Convolution	(7,7)	128
	B4	Concatenation	~	~
	B4	Max pooling	~	~
	B5	Convolution	(3,3)	256
	B5	Convolution	(5,5)	256
	B5	Convolution	(7,7)	256
	B5	Concatenation	~	~
	B5	Max pooling	~	~
Flat	B6	Flatten	~	~
	B7	Dense	128	~
	B7	Dropout	~	~
Decode	B7	Dense	256	~
	B7	Dropout	~	~
	B7	Dense	4096	~
	B7	Reshape	(64,64)	~

Table 1. The structure of the multi-kernel CNN model.

3.5 Accuracy assessment

F1-score and overall accuracy (OA) were used to evaluate the proposed model performance. To calculate the F1 score, recall, and precision metrics are combined (see equation (2)), also the calculation formulas corresponding to the OA are shown in equation (5).

$$F1(\%) = \left(\frac{2 \times precision \times recall}{precision + recall}\right) \times 100 \quad (2)$$

$$precision(\%) = \left(\frac{TP}{TP + FP}\right) \times 100 \quad (3)$$

$$recall(\%) = \left(\frac{TP}{TP + FN}\right) \times 100 \quad (4)$$

$$OA(\%) = \left(\frac{TP + TN}{TP + FP + FN + TN}\right) \times 100 \quad (5)$$

where FN is the false negative rate, TN is the true negative rate, FP is the false positive rate, and TP is the true positive rate.

4. RESULTS AND DISCUSSION

This section explains the experimental settings, illustrates the implementation details, presents the experimental results, and discusses each aspect.

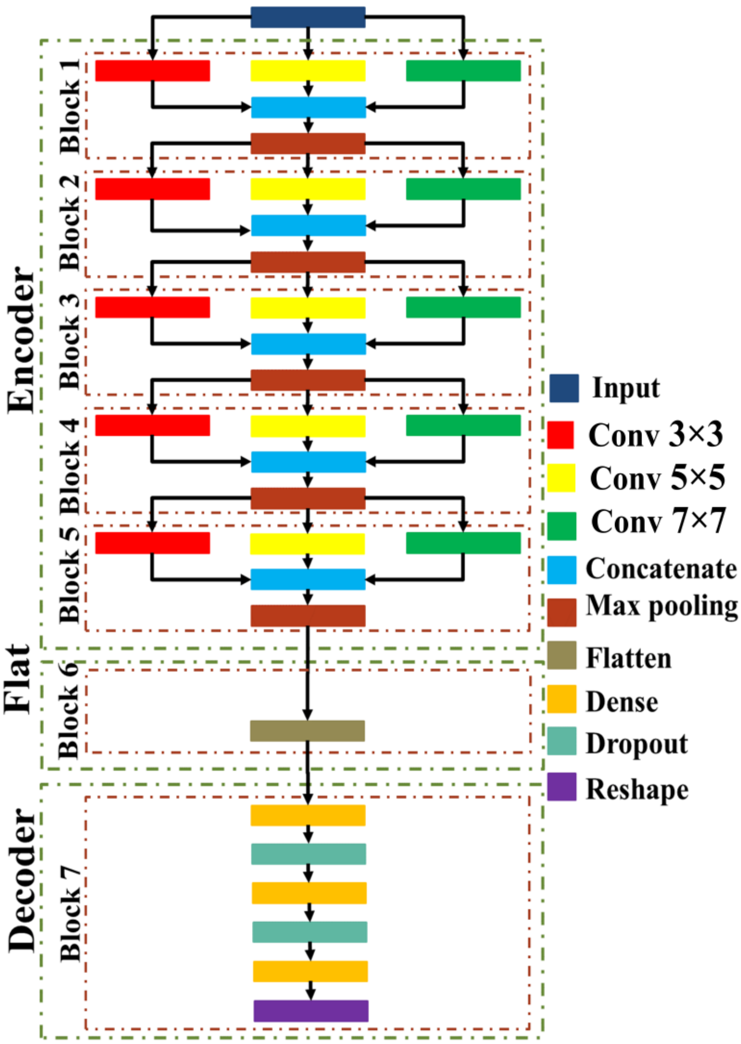

Figure 3. The proposed CNN architecture in this study.

4.1 Experimental Results

Tensorflow and Keras were used to develop the proposed multi-kernel CNN model. Training and testing were performed on a machine with NVIDIA Geforce GTX 1650Ti GPU. A total of 8616 samples were used for training, 685 samples for validation, and 1002 samples for testing. The Dice coefficient (DC) (Sudre, Li et al. 2017) is used as a loss function that maximizes the overlap between the predicted and ground truth images. Equation (6) shows how to compute the similarity between the predicted image and the ground truth image. batch size during the training phase set to 4 empirically. The multi-kernel CNN model was trained using a learning rate value of 0.0005, and 250 iterations from scratch.

$$DC = 1 - \frac{2|\Upsilon \cap X|}{|\Upsilon| + |X|} \tag{6}$$

where X is the predicted map, Y is the ground truth input, and ∩ is the intersection of the ground truth Y and the predicted map X.

4.2 Quantitative results

The training and validation loss over 250 epochs for wildfire spread prediction using multi-kernel CNN models are shown in Figure 4. Validation and training loss in the first epochs validation loss is almost constant without any effective change and just fluctuated but after some iteration, the model started to learn and validation loss decreased. According to figure 4, we notice that across the board, the loss function value of training and validation gradually decreases between the 200th and 250th iterations, and the model approaches convergence steadily.
Using the 64×64×12 image, this model can predict whether a pixel is on fire in the next time step. Table 2 shows the precision, recall, and OA of the multi-kernel CNN model based on the training set, validation set, and test set. The recall is lower than the precision in Table 2, indicating a slight preference for false negatives. Therefore, the model predicts there is no fire when there is one. It is also important to compare results between CNN with and without the multi-kernel mechanism as the results obtained with the multi-kernel mechanism are generally better in all metrics.

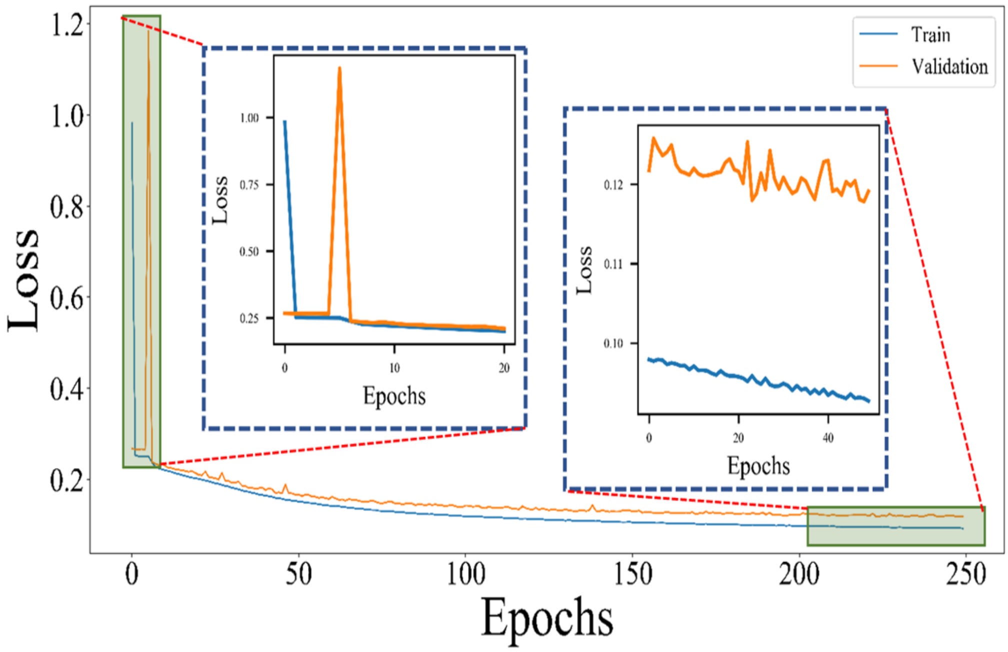

Figure 4. The training and loss function value of the multi-kernel CNN.

Data set	Models							
	CNN without the multi-kernel mechanism				Multi kernel CNN			
	OA	F1	Precision	Recall	OA	F1	Precision	Recall
Training set	94.15	79.52	89.32	71.67	99.9	87.36	93.1	83.9
Validation set	82.48	58.17	61.91	54.88	99.3	71.63	88.3	60.2
Test set	82.12	57.66	62.38	53.62	98.6	70.97	87.5	59.7

Table2. The model demonstrated highly accurate predictions of pixels' states when tested on validation and set.

4.3 Qualitative results

Based on weather conditions, fuel conditions, or topography, the proposed model can identify burned and unburned pixels in the next time step. The proposed multi-kernel CNN model accurately predicted wildfire spread prediction, as demonstrated in the evaluation results. An example of the qualitative results generated by the multi-kernel CNN model is shown in Figure 5.

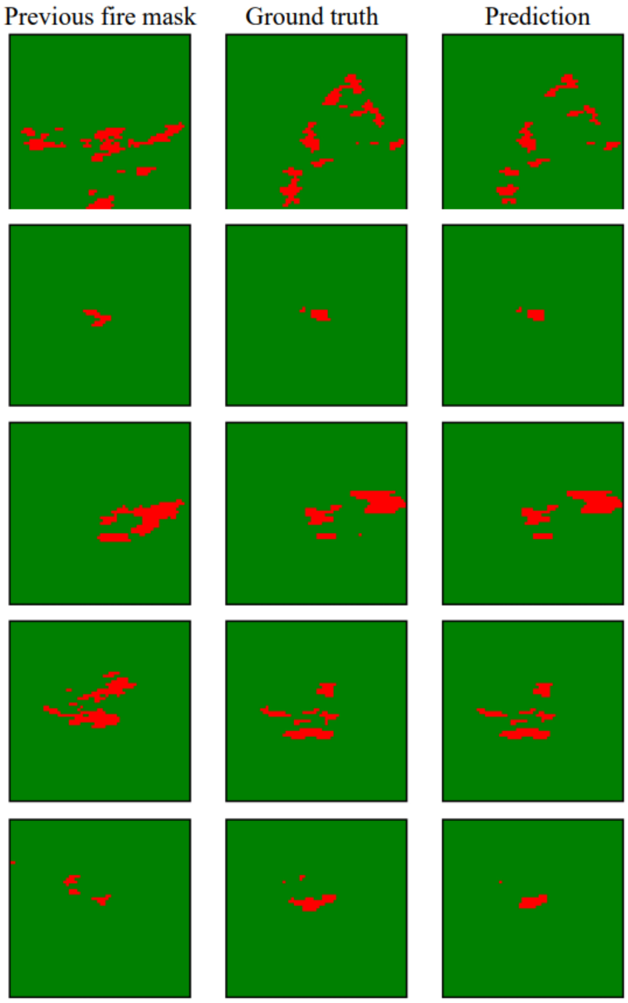

Figure 5. Qualitative results generated by the multi-kernel CNN model.

5. CONCLUSION

A CNN model based on the multi-kernel has been proposed to predict the spread of wildfires. Each pixel in the image is represented by a binary value generated by the model based on a 64×64×12 tensor input. 12 bands include elevation, wind direction and speed, minimum and maximum temperatures, humidity, precipitation, drought index, normalized difference vegetation index (NDVI), and energy release component. As a result of applying new wildfire spread datasets, some interesting conclusions have been drawn. The multi-kernel CNN model achieved high accuracy compared to the model reported by (Huot, Hu et al. 2021) as a publisher of the wildfire dataset. Secondly, multi-kernel CNN can extract more high-level features and assist in learning wildfire spread patterns. Multi-kernel CNN network is used to extract the feature of the data by considering the complex variables. This may give us some new ideas on how to deal with massive data and collect relevant information. With the use of the multi-kernel mechanism, the model's performance has increased significantly when the iterations over a certain step. Third, compared with a CNN model without a multi-kernel mechanism and fixes kernel size, the proposed model produces a better prediction result using the test dataset. Next, we will build a general model using global fire records and apply transfer learning to fine-tune the model for particular regions. Upsampling or convolution transpose layers can also be used to reduce model parameters and complexity.

6. REFERENCES

Agarap, A. F. (2018). "Deep learning using rectified linear units (relu)." arXiv preprint arXiv:1803.08375.

Burton, P. J., et al. (2008). "Large fires as agents of ecological diversity in the North American boreal forest." International Journal of Wildland Fire 17(6): 754-767.

Cheng, T. and J. Wang (2008). "Integrated spatio-temporal data mining for forest fire prediction." Transactions in GIS 12(5): 591-611.

Chetehouna, K., et al. (2015). "Predicting the flame characteristics and rate of spread in fires propagating in a bed of Pinus pinaster using Artificial Neural Networks." Process Safety and Environmental Protection 98: 50-56.

Ferragut, L., et al. (2007). "A numerical method for solving convection–reaction–diffusion multivalued equations in fire spread modelling." Advances in Engineering Software 38(6): 366-371.

Finney, M. A. (1998). FARSITE, Fire Area Simulator--model development and evaluation, US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Ganapathi Subramanian, S. and M. Crowley (2018). "Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images." Frontiers in ICT 5: 6.

Green, M. E., et al. (2020). "Modeling Wildfire Perimeter Evolution using Deep Neural Networks." arXiv preprint arXiv:2009.03977.

Hodges, J. L. and B. Y. Lattimer (2019). "Wildland fire spread modeling using convolutional neural networks." Fire technology 55(6): 2115-2142.

Huot, F., et al. (2021). "Next Day Wildfire Spread: A Machine Learning Data Set to Predict Wildfire Spreading from Remote-Sensing Data." arXiv preprint arXiv:2112.02447.

Jamali, A., et al. (2021). "Wetland Mapping Using Multi-Spectral Satellite Imagery and Deep Convolutional Neural Networks: A Case Study in Newfoundland and Labrador, Canada." Canadian Journal of Remote Sensing 47(2): 243-260.

Liang, H., et al. (2019). "A neural network model for wildfire scale prediction using meteorological factors." IEEE Access 7: 176746-176755.

Mahdianpari, M., et al. (2018). "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery." Remote Sensing 10(7): 1119.

Radke, D., et al. (2019). FireCast: Leveraging Deep Learning to Predict Wildfire Spread. IJCAI.

Sønderby, C. K., et al. (2020). "Metnet: A neural weather model for precipitation forecasting." arXiv preprint arXiv:2003.12140.

Sudre, C. H., et al. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. Deep learning in medical image analysis and multimodal learning for clinical decision support, Springer: 240-248.

Yamak, Ç. (2006). Investigation over a national meteorological fire danger approach for Turkey with geographic information systems, Middle East Technical University.

Zhang, G. (2004). Study on forest fire dynamic monitoring in Guangzhou City, Central South Forestry University Zhuzhou.

Zheng, Z., et al. (2017). "Forest fire spread simulating model using cellular automaton with extreme learning machine." Ecological Modelling 348: 33-43.