Fire Technology manuscript No.
(will be inserted by the editor)

Wildland Fire Spread Modeling Using Convolutional Neural Networks

Removed for Peer Review

Received: date / Accepted: date

Abstract This paper presents a novel predictive analytics approach to estimating the spread of a wildland fire using a convolutional neural network (CNN). Simulated burn maps for use in this process were generated at six hour intervals using the phenomological fire spread model of Rothermel with 10,000 different combinations of input parameters. The robustness of the approach is tested using 1,000 simulations not included when training the CNN. Overall the predicted burn maps from the CNN-based approach agreed with simulation results, with mean precision, sensitivity, and F-measure of 0.97, 0.92, and 0.93, respectively. Noise in the input parameters was found to not significantly impact the CNN-based predictions. The computational cost of the method was found to be comparable to a phenomological model in homogenous spatial conditions, and significantly better for heterogenous spatial conditions. Although trained on predictions six hours apart, the CNN-based approach is shown to be capable of predicting burn maps further in the future by recursively using previous predictions as inputs to the model. When the initial fire was small, the model tended to under-predict fire spread; however, predictions generally improved as the fire grew.

Keywords Wildland Fire · Machine Learning · Neural Network · Fire Spread · Convolutional Neural Network · CNN

1 Introduction

Wildland fire propagation is a complex process which involves the interactions of many underlying physical phenomena. Since fully resolving these processes remains a research effort researchers have developed simplified models which describe the growth rate of a wildland fire. A new wildland fire spread model which is capable of predicting the 2-D spatial-temporal flame front faster than real-time for diverse landscapes, fuel types, and weather conditions is needed. Existing wildland fire spread models generally fall into three categories: stochastic, phenomological, and

Removed for Peer Review

physical. A number of different fire spread model approaches have been proposed with each class of model having different advantages and limitations [1–3].

Stochastic fire spread models are based on statistical analysis of historical wildland fires and prescribed burns. Sullivan examined 14 stochastic models developed between 1990-2013 and found each could be summarized by the functional form chosen to describe the impact of wind speed, zero wind fire spread, and fuel moisture content [3]. The impact of wind on wildland fire spread generally follows a power law where the exponent and pre-factor are related to the fuel type. The impact of fuel moisture content on wildland fire spread is generally modeled as strongly linear or weakly exponential based on the fuel type. These models are favored by municipalities as they are capable of providing rapid reasonable estimates of the overall fire risk of the region for which they were derived. However, stochastic models fail when predicting conditions outside the historical data, such as new weather conditions, spatial regions, or different fuel types. In addition, these models generally make no attempt at resolving the spatial-temporal flame front of a wildland fire.

Phenomological fire spread models are based on conservation of energy, but use experimental measurements to develop functional forms rather than modeling from first principles [1,3]. The most widely known phenomological fire spread model is the model of Rothermel [4,5] which uses empirical correlations for heat source and sink terms [1]. The baseline rate of spread (with no wind or slope) is based on fuel density, type, and moisture content. The impact of wind and slope is modeled as a multiplier of the baseline rate of spread. Although developed to determine the rate of spread from a single ignition source in a single direction, these models have been expanded to work in 2-D using Huygens' principle [6,7] and the level set method [8,9]. These models are favored by foresters and firefighters in the field as they are capable of providing rapid estimates of the rate of spread [10]. However, phenomological models inherit similar issues to stochastic models when predicting conditions outside the historical data used in deriving the empirical models. In addition, expanding the model to 2-D to predict the spatial-temporal flame front significantly increases the computational requirements. Researchers have used reduced order modeling to decrease the computational cost of evaluating the 2-D rate of spread using phenomological models [11]. However, it is difficult to apply this framework to other models as new derivations are required.

Physical fire spread models are based on the fundamental chemistry and physics of combustion, and fire spread [2]. Physical models can be further subdivided into simple and detailed models. Simple physical models such as that of Weber consider only the transport of energy, neglecting calculation of the full flow field for computational reasons [1]. These models are more general than stochastic or phenomological models and can be close to real-time [10]. Detailed physical models such as Wildland-Urban Interface Fire Dynamics Simulator (WFDS) consider the transport of mass, momentum, and energy in addition to combustion in a multiphase approach through computational fluid dynamics (CFD) [12]. These models are favored by researchers interested in the fundamental mechanisms of wildland fire spread as the basis on first principles reduces the impact of user intuition on the results. However, the computational cost to model moderate domains ($2.25km^2$) is significant [12].

Predictive analytics uses simplified functions to represent underlying connections between data to make estimates of future events, often using machine learn-

ing. Several researchers have presented predictive analytics frameworks to estimate the total burned area of a fire based on meteorological data [13–17]. Each of these methods is capable of predicting the total burned area; however none attempts to estimate the spatial-temporal fire distribution of a wildland fire front. A predictive analytics approach to predicting the spatially-resolved final burned region of a wildland fire was presented by McCormick. The model used an artificial neural network to classify the the center pixel of a 3x3 neighborhood of pixels as burned or unburned [18,19]. The initial results are promising; however, the model relies on the assumed ellipsoidal growth profile in the direction of wind [6] and only predicts the final burned region but not the evolution of spread.

A method to predict fully resolved spatial-temporal burn maps is desired. The fundamental principle which makes convolutional neural networks (CNNs) versatile is the capability to learn how to represent complex shapes as combinations of high level feature maps. Krizhevsky showed many of the features learned by the CNN in the ImageNet competition described the inter-relationship of the 3 color channels [20]. As an analogy to image classification, data such as elevation, moisture content, and wind speed can be treated as channels in an image. Given enough data, a CNN will be able to learn relationships between these physical parameters which can then be used to predict a future burn map.

The objective of this study is to apply a trained CNN framework to predict the spatial-temporal distribution of a wildland fire front in homogenous vegetation without relying on any other models at runtime. Data for use in training and testing the network was generated using Rothermel's phenomological model. The quantitative performance of the model on 1,000 simulations not included during training is presented, and the sensitivity of the network to noise is examined. The work presented herein demonstrates the concept on a simple configuration with future work to expand the method to use experimental data with heterogeneous spatial conditions.

2 Methods

A schematic showing a high level view of the solution algorithm is shown in Fig. 1. Each primary driver of wildland fire spread is included as a channel in an image which is input to the CNN. The CNN then uses its prior training to predict a new image with two channels corresponding to the probability a pixel has burned or not burned in the future. The two probability masks are then post-processed to output a single future burn map. The temporal resolution of six hours and spatial resolution of 1 pixel/km physically correspond to available global satellite measurements. The following subsections describe the simulation conditions, network architecture, post-processing, and performance metrics used in this work.

2.1 Wildland Fire Prediction

Data for this study was generated using the surface fire spread model presented by Rothermel/Albini [4,5,21]. In the Rothermel/Alibini phenomological fire spread

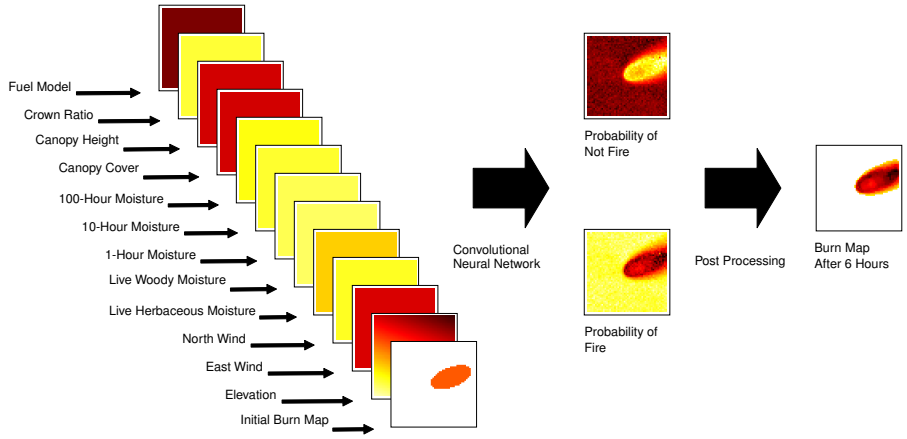


Fig. 1 Schematic of solution algorithm. The left set of images show the different channels used as inputs to the neural network. The values for each data channel are colorized based on the values shown in Table 1 and Table 2.

model, the peak surface fire spread rate, $V_{s,peak}$ is calculated using the equation

$$V_{s,peak} = \frac{Q''\zeta}{\rho\epsilon Q_{ig}}(1 + \phi_s + \phi_w) \tag{1}$$

where Q'' is the heat release rate per unit area, ρ is the fuel density, Q_{ig} is the heat of pre-ignition, ζ is the propagating flux ratio (percentage of heat released which pre-ignites fuel), ϵ is the effective heating number (percentage of fuel which is involved in ignition), ϕ_w is the wind coefficient, and ϕ_s is the slope coefficient.

Various researchers have developed empirical relationships for the different parameters in Eq. 1. A commonly used approach in the literature is to specify Q'', ρ, ζ, ϵ based on classifying the primary fuel in a region into a fuel model. A total of 53 fuel models were considered in this work including 13 developed by Rothermal/Albini [4,21], and 40 developed by Scott [5]. Rothermel presented an empirical relationship for Q_{ig} based on the fuel model and moisture content, and Scott extended the relationship to handle dynamic fuel models. Rothermel presented empirical relationships for ϕ_w and ϕ_s based on fuel model, midflame wind speed, and slope. Andrews presented an algorithm to adjust typical atmospheric wind measurements (10m or 20ft) to mindflame wind speed based on three additional parameters describing the upper story vegetation (canopy cover, canopy height, and crown ratio) [22]. Researchers have shown wildland fires grow in a generally ellipsoidal shape for homogenous spatial conditions based on $V_{s,peak}$ and wind speed [6,23,24]. The primary drivers in the fire spread model were identified as landscape (slope, aspect, and fuel model type), moisture content (1-hour, 10-hour, 100-hour, live woody, and live herbaceous), canopy type (height, ratio, and percent coverage), and 10m wind (intensity and direction). The allowable bounds used for each parameter in this work are shown in Table 1. The fuel model types were assigned indexes based on the peak rate of spread under the same spread conditions (low moisture, 10 mph wind up a 0.5 slope). Since slope and aspect can be sum-

marized as a 2-D difference in elevation, a single channel for elevation was used instead of two channels for slope and aspect in the neural network.

Table 1 Limits of each parameter in study.

Parameter	Unit	Min	Max
Aspect	Degrees	0.0	360
Fuel Model	Index	0.0	53
Slope	Fraction	0.0	1.0
1-Hr Moisture	Percent	1.0	40
10-Hr Moisture	Percent	1.0	40
100-Hr Moisture	Percent	1.0	40
Live Herbaceous Moisture	Percent	30	100
Live Woody Moisture	Percent	30	100
Canopy Cover	Percent	0.0	1.0
Canopy Height	Feet	1.0	20
Crown Ratio	Fraction	0.1	1.0
Wind Direction	Degrees	0.0	360
Wind Velocity	Mi/Hr	0.0	30

A custom implementation of Rothermel's fire spread model was developed in Python to streamline the integration with the CNN software. The implemention of Rothermel's model in BehavePlus was used to validate the simulation framework [25]. A total of 10,000 different combinations of the parameters shown in Table 1 were simulated. The total time to run 10,000 simulations was 30.6 seconds. Raster images of the burned map were generated at a resolution of 1 pixel/km every six hours for a total of 24 hours. Three cases for use in the CNN were generated from each simulation by creating pairs of burned maps six hours apart. The zero hour to six hour pair was not considered as a case for the CNN as the zero hour burned image always contains no fire. Example burned maps from one simulation are shown in Fig. 2 for the parameter values shown in Table 2.

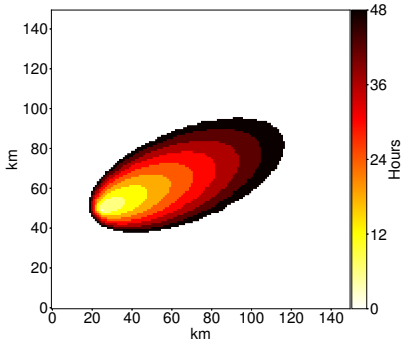


Fig. 2 Example Simulated Burn Maps

Table 2 Parameters for Example Simulation

Parameter	Unit	Value
Aspect	Degrees	130
Fuel Model	Index	FM1 (44)
Slope	Fraction	0.8
1-Hr Moisture	Percent	5.3
10-Hr Moisture	Percent	6.3
100-Hr Moisture	Percent	7.3
Live Herbaceous Moisture	Percent	69
Live Woody Moisture	Percent	49
Canopy Cover	Percent	0.7
Canopy Height	Feet	14
Crown Ratio	Fraction	0.2
Wind Direction	Degrees	34
Wind Velocity	Miles per Hour	13.5

2.2 Network Architecture

At a fundamental level, artificial neural networks are massively parallel equations which have the capability to store observed knowledge about a problem to make predictions of new inputs. A convolutional neural network (CNN) assumes the input data has distinct spatial dependence within the input parameters. Since the network assumes spatial dependence, less connections need to be made to inputs which are far from each other. This allows a CNN to contain much fewer connections and parameters than a similarly sized standard feed forward network with minimal loss in optimal performance for appropriate problems. In addition, this makes it possible to use deeper and more broad hidden layers without increasing computational requirements beyond what is feasible on current technology [20]. Representing the spread of a wildland fire front with a CNN is reasonable as wildland fire spread is a local phenomena [6].

The CNN architecture used in this work is shown in Fig. 3. The input images were 50x50 pixels with 13 image channels corresponding to the image stack shown in Fig. 1. The output image contains 50x50 pixels with two image channels corresponding to the probability the burn map has reached a pixel and the probability the burn map has not reached a pixel. A total of seven hidden layers are included in the network including two convolutional, two max pooling, one fully connected classification, one up-convolutional, and one soft max layers. The number of filters and step size in each convolutional and up-convolutional layers and number of neurons in fully connected layers were specified to steadily decrease the degrees of freedom from the 32,500 in the input layer (50x50x13 inputs) to the desired degrees of freedom of 5,000 in the output layer (50x50x2 probability of fire and not fire).

All layers in the network presented in this work used leaky rectified linear unit (ReLU) activation functions except the fully connected layer which used hyperbolic tangent (TanH) activation functions and the output layer which used a softmax activation function. Researchers have shown deep neural networks learn more quickly and more accurately using leaky or parametric ReLU than logistic or TanH for activation [26, 27]. However, since the activation is unbounded, ReLU is not appropriate for classification layers. Since the intent of the fully connected

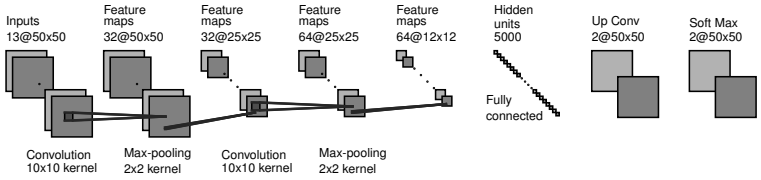


Fig. 3 Convolutional Neural Network Architecture.

layer is to classify the feature maps into high level descriptions of the fire and environment, a TanH activation function is appropriate. Since the desired output is a probabilistic estimate of whether or not each pixel would be burned, a soft max activation function is appropriate to scale the activations of the fire and non-fire probability masks.

The network architecture was built using the Python 3 bindings for TensorFlow [28]. The models were trained using stochastic gradient descent with a batch size of 100 samples. All weights and biases were initialized from a uniform distribution between -1 and 1. The learning rate was fixed for all layers throughout training at 0.0001. The cost function used in training was based on sum square error. Overfitting was reduced by using 50% dropout on the input layer and shuffling the order of the samples during training. The network was trained using 9,000 simulations (27,000 pairs of burn maps) for 50,000 cycles using a single NVIDIA Quadro K620. The total time to train the network was 18 hours 7 minutes. The total time to evaluate the network with all 10,000 simulations (30,000 pairs of burn maps) was 38.1 seconds.

2.3 Performance Metrics

The metrics used to quantify the performance of the CNN in this study were precision, sensitivity, and F-measure. For each metric, the range of possible values is zero to one, with a perfect score being one. The precision, P, is a measure of commission errors (predicting a fire where there was not fire) and is defined as

$$P = \frac{t_p}{t_p + f_p} \tag{2}$$

where t_p is the number of correctly identified fire pixels, and f_p is the number of falsely identified fire pixels. The sensitivity, S, is a measure of omission errors (predicting no fire where there was a fire) and is defined as

$$S = \frac{t_p}{t_p + f_n} \tag{3}$$

where f_n is the number of fire pixels which were identified as non-fire pixels. F-measure, F, is an overall measure of performance defined as the harmonic mean of P and S,

$$F = 2 \cdot \frac{P \cdot S}{P + S}. \tag{4}$$

2.4 Post Processing

The output layer of the CNN contains two normalized probability masks, one for fire and one for not fire. The normalized probability mask for fire is post-processed to convert the probabilistic estimate of burned or unburned to a single contour. A 3x3 median filter is applied to smooth the image. A threshold value on probability of fire is used to determine whether or not each pixel has been burned.

The optimal threshold to use in post-processing was determined by calculating the mean F-measure for CNN predictions of the 9,000 training simulations (27,000 pairs of burn maps) with thresholds ranging from 0.01 to 0.99. The mean F-measure was found to be mostly independent of the post-processing threshold in the range of 0.2-0.6 as shown in Fig. 4. The maximum mean F-measure of the training data was calculated with a threshold of 0.41. This threshold was fixed and used for post-processing throughout this work. An example neural network prediction before and after post-processing is shown in Fig. 5.

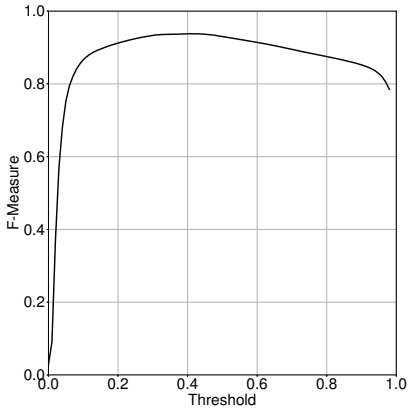
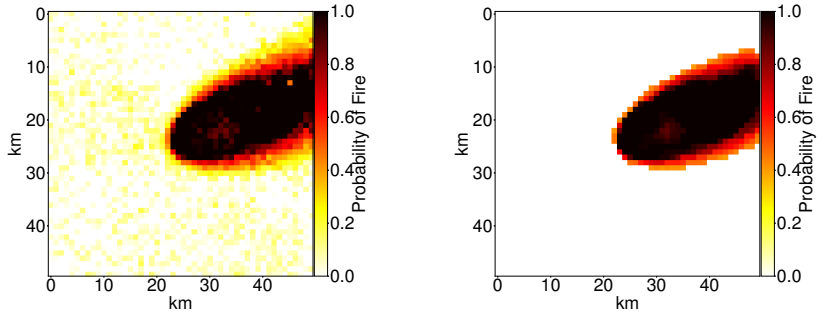


Fig. 4 Mean F-Measure of CNN predictions of 9,000 training simulations (27,000 pairs of burn maps) for different threshold values.

3 Results

The robustness of the neural network to predict new fires was examined by considering 1,000 simulations (3,000 pairs of burn maps) which were not included when training the network. Sample CNN predictions from five of these test cases are compared with simulation predictions in Fig. 6. Figure 6a shows the initial and next burn map from the simulation. Figure 6b shows the final burn map predicted by the CNN. Figure 6c highlights pixels which the CNN prediction did not match the simulation predictions. Pixels shown as black represent commission errors (false positive of fire), and pixels shown as orange represent omission errors (false negative of fire).

The mean precision, sensitivity, and F-measure of the 1,000 test simulations (3,000 burn map pairs) are shown in Table 3. Each metric ranges from zero to one



(a) (b)

Fig. 5 Example CNN prediction of burn map probability (a) Directly predicted from CNN (b) After post-processing

with a perfect score being one. The distribution of F-measure for the 3,000 burn map pairs are shown in Fig. 7. Percentiles of each metric are shown in Table 3 to quantify the spread of each score. Here a percentile is defined as

$$X = \frac{1}{N_{total}} \int_{Y_{min}}^{1} N(Y)dy \tag{5}$$

where X is the percentile, $N(Y)$ is the number of burn maps achieving a specific score, N_{total} is the total burn maps Y_{min} is the minimum score. For example, the F-measure of 0.86 for $X = 80\%$ shown in Table 3 means 80% (2400/3000) of the burn map pairs had an F-measure of 0.86 or higher.

Table 3 Performance Metrics of CNN Predictions of Test Cases

Parameter	Mean	$X = 80\%$	$X = 90\%$	$X = 95\%$
Precision	0.97	0.95	0.85	0.79
Sensitivity	0.92	0.80	0.67	0.59
F-Measure	0.93	0.86	0.80	0.73

4 Discussion

The overall shape of the burn maps predicted by the CNN are consistent with the simulations for the 1,000 test simulations (3,000 burn map pairs) examined in this work, as shown in the examples in Fig. 6. The burn maps predicted by the CNN do not contain non-physical holes or excessive noise. The direction of maximum growth is captured well. Figure 6 shows the CNN is able to predict the growth of small ($1km^2$, rows 2 and 4), intermediate ($10km^2$, rows 1 and 5), and large ($100km^2$ row 3) fires.

Across all test cases the was not significant bias observed in over-predicting or under-predicting fires, as is shown by the comparable mean sensitivity and precision shown in Table 3. However, the spread in sensitivity is higher than the

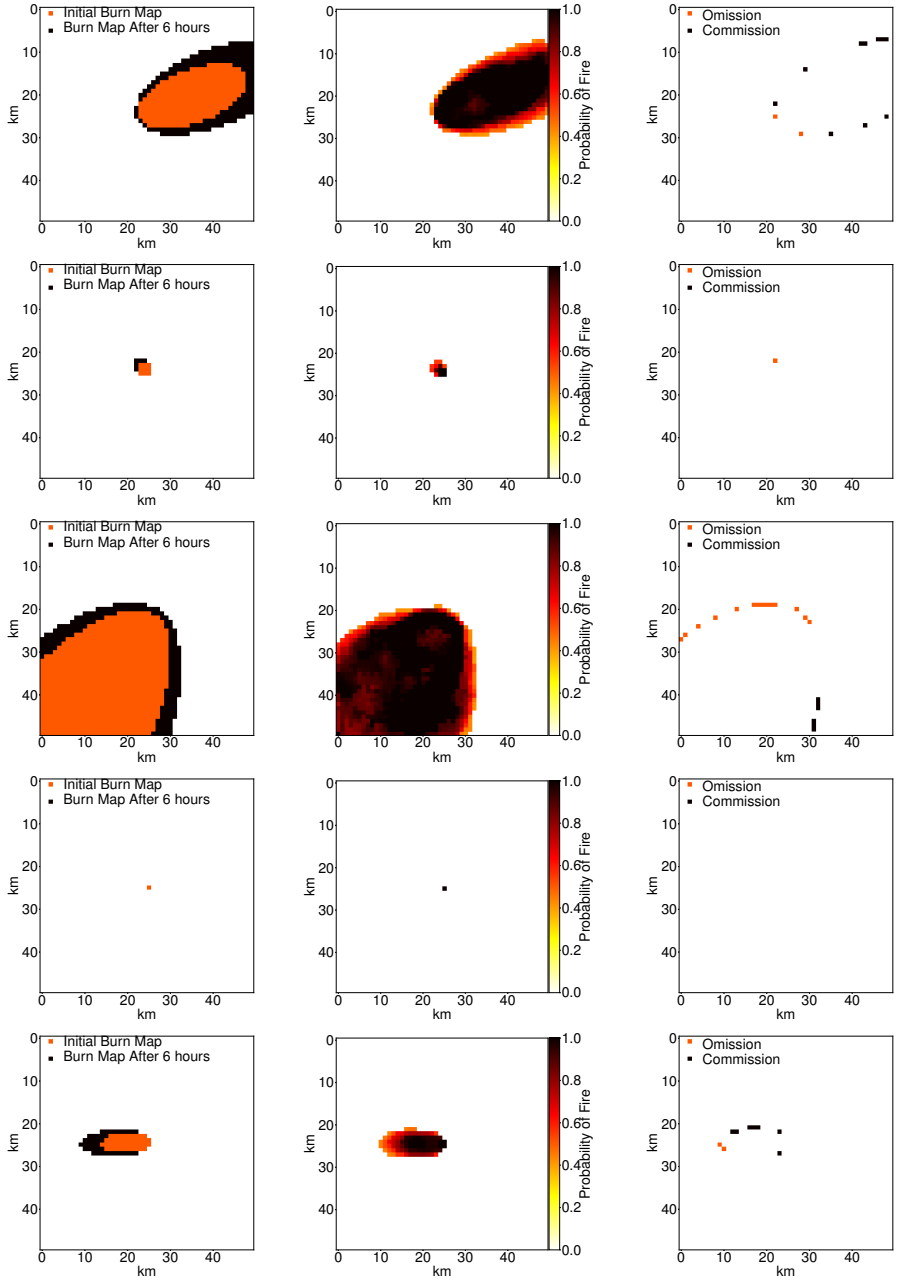


Fig. 6 Example CNN prediction results (a) Simulation burn maps (b) CNN predicted burn map (c) Classification error

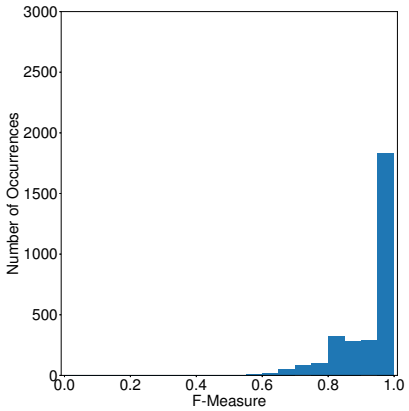


Fig. 7 F-Measure distribution of CNN predictions of test cases.

spread in precision as is shown by $X = 80\%$, $X = 90\%$, and $X = 95\%$ shown in Table 3. Examining the histogram of F shown in Fig. 7, there is a sharp drop in number of occurrences at $F < 0.8$. The initial fire size of 82% of the cases where $F < 0.8$ was found to be nine pixels or less, as shown in Fig. 8. Combined with the percentiles shown in Table 3, this shows the sensitivity is impacted to a greater extent than precision for small fire sizes. Since a CNN relies on feature recognition, low feature density in the inputs (such as fires less than 9 pixels in size) can lead to a decrease in the accuracy of the model. If smaller fires are of primary interest, this effect could be reduced by increasing the spatial resolution of the model.

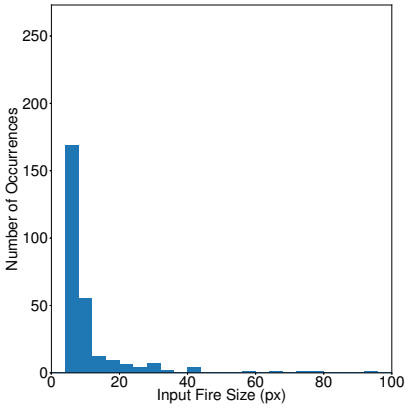


Fig. 8 Distribution of input fire size for all cases where $F < 0.8$.

The sensitivity of the model to noise in the input parameters was examined by adding noise to the 5 examples shown in Fig. 6 and examining the impact on S, P, and F. Noise was added by multiplying each input channel by a random value from a log-normal distribution with a mean of 0 and a standard deviation of 1. Note the input and output burn maps were not changed. A total of 3,000

iterations for each of the five test cases were predicted by the CNN. The zero noise, noise mean, and noise 80% scores for each case for each metric are shown in Table 4. The largest impact was observed in the last test case where the mean and 80% sensitivity dropped from the baseline score by 0.07 and 0.14, respectively. This shows the addition of noise resulted in an increase in the under prediction of the burn map. However, the minimal overall impact on the mean and 80% scores shows the general shape of the predicted burn maps are still in agreement with simulation results.

Table 4 Impact of Noise on Performance Metrics of CNN Predictions of Test Cases

Sensitivity			Precision			F-Measure		
Base	Noise	80%	Base	Noise	80%	Base	Noise	80%
0.99	0.98	0.99	0.98	0.98	0.98	0.98	0.98	0.98
0.93	0.91	0.93	1.00	1.00	1.00	0.96	0.94	0.96
0.98	0.97	0.95	0.99	0.99	0.99	0.98	0.98	0.97
1.00	1.00	1.00	1.00	0.93	1.00	1.00	0.95	1.00
0.98	0.91	0.84	0.92	0.90	0.85	0.95	0.90	0.88

One of the key benefits of the CNN architecture presented in this work is the scalability in heterogeneous spatial conditions. Recall the total time to run 10,000 simulations using the model of Rothermel was 30.6s compared to the 38.1s required for the CNN. The phenomological model will likely perform faster whenever evaluating the spread of a fire from a single point in homogenous conditions. However, when using Huygen's principle to evaluate Rothermel's model in heterogeneous spatial conditions, the fire perimeter must be discretized and evaluated for each point along the perimeter. This leads to a significant increase in computational time as the size of the fire increases. Since the CNN is already analyzing all the input channels as 2-D image channels, there is no increase in computational cost to add heterogeneous conditions. As an example, a 10km fire perimeter with a 30m grid will require 333 evaluations of the phenomological model, with a total computational time of 1.0s. The CNN approach will require a single evaluation, with a total computational time of 0.004s.

Although the model was trained using a six hour time interval between the input and output burn maps, it is possible to obtain predictions at points further in the future at six hour intervals by recursively using the previous prediction as an input to the CNN. Figure 9 shows five example cases where this process was used to predict burn maps up to 24 hours from ignition based on an input burn map six hours after ignition. The results for each case show the general direction of spread is captured well, with $F > 0.8$ in all cases except the fourth case where the input and output fires are small. The sensitivity of the predictions generally increases with time, whereas the precision generally decreases with time. This shows early in the progression of the fire the model under-predicts the rate of spread, which highlights the difficulty the CNN can have when dealing with low feature density. Later in the progression of the fire, the model begins to over-predict the rate of spread. Since the network was trained on constant rates of spread for each configuration, the over-prediction is likely a result of the network over-correcting for the initial under prediction of rate of spread.

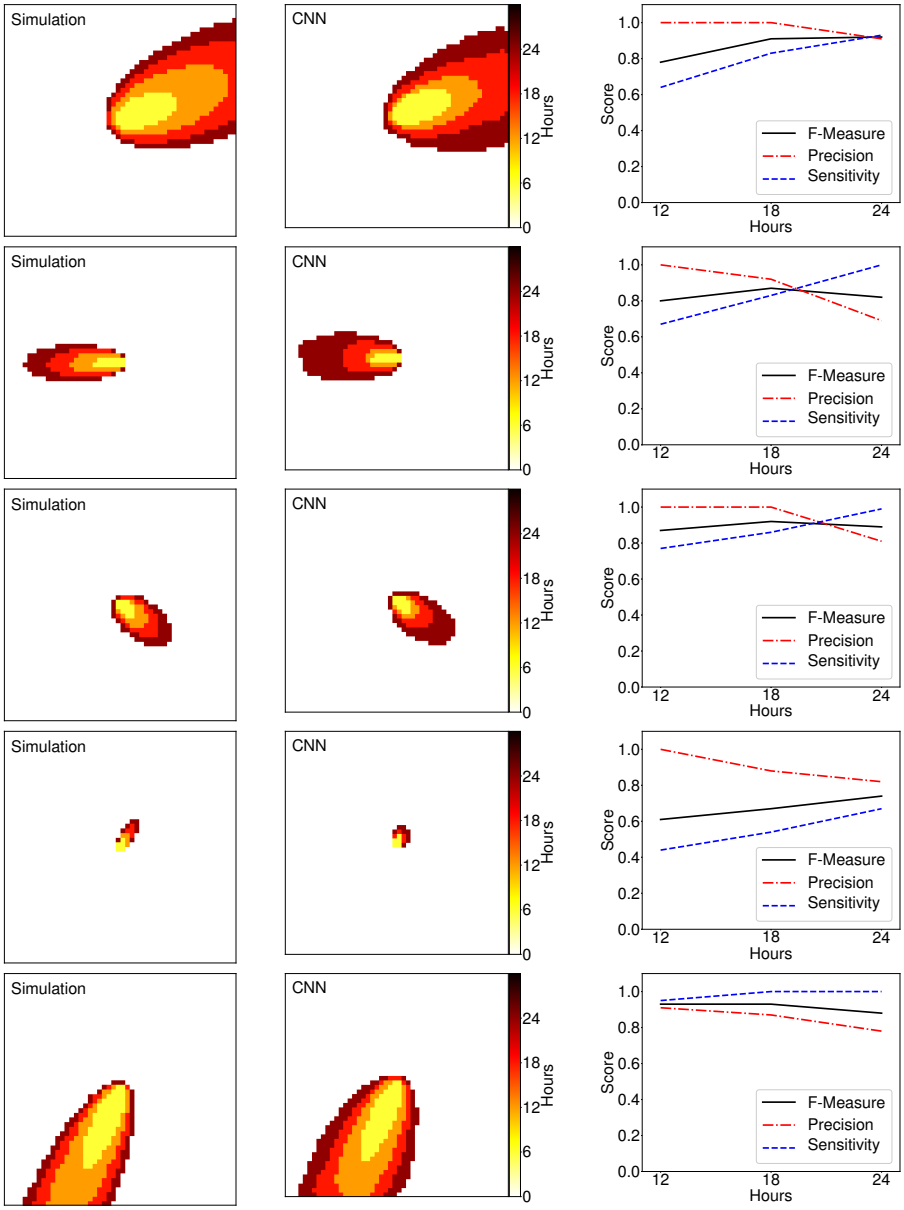


Fig. 9 Example CNN prediction of time resolved burn maps (a) Simulation burn maps (b) CNN prediction burn maps (c) Performance metrics at six hour intervals

5 Conclusion

A novel predictive analytics approach to estimating the spread of a wildland fire using a convolutional neural network (CNN) was presented. The robustness of the approach was tested using 1,000 simulations (3,000 burn map pairs) not included when training the network. The predictions of burn maps from the CNN-based approach agreed with simulation results, with mean precision, sensitivity, and F-measure of 0.97, 0.92, and 0.93, respectively across a diverse set of input parameters. Noise in the input parameters was found to have a minimal affect on the CNN predictions. The computational cost of the method was found to be comparable to simulating a phenomological model for homogenous spatial conditions, and significantly better for heterogeneous spatial conditions. Although trained on predictions six hours apart, the CNN-based approach is capable of predicting burn maps further in the future by recursively using previous predictions as inputs to the model. The cases where F-measure was observed to be less than 0.80 were found to have input burn maps of less than nine pixels. This indicates as the fire continues to grow, the predictions will continue to improve.

This work represents a first step in creating a framework to predict wildland fire spread based on physics based models and historic data. Although the data used to train the CNN in this work was generated using a phenomological model, the model does not have any information about whether the data is from a computational fluid dynamics model, phenomological model, or even experimental measurements of burn maps. Additionally, although the simulations used homogenous vegetation and landscapes, the feature learning aspect of the CNN-based method is well posed to learn heterogenous spatial conditions. The next step in this process is to incorporate data from simulations with spatially varying environmental conditions in the training and test sets.

References

1. R. Weber, Modelling fire spread through fuel beds, Progress in Energy and Combustion Science **17**(1), 67 (1991)
2. A. Sullivan, A review of wildland fire spread modelling, 1990-present 1: Physical and quasi-physical models, arXiv:0706.3074v1[physics.geo-ph] (2008)
3. A. Sullivan, A review of wildland fire spread modelling, 1990-present 2: Empirical and quasi-empirical models, arXiv:0706.4128[physics.geo-ph] (2013)
4. R.C. Rothermel, et al., A mathematical model for predicting fire spread in wildland fuels. Tech. rep., USDA Forest Service (1972)
5. J.H. Scott, R.E. Burgan, Standard fire behavior fuel models: a comprehensive set for use with rothermel's surface fire spread model. Tech. rep., USDA Forest Service (2005)
6. M.A. Finney, Mechanistic modeling of landscape fire patterns, Spatial Modeling of Forest Landscapes: Approaches and Applications. Cambridge University Press, Cambridge pp. 186–209 (1999)
7. M.A. Finney, et al., *FARSITE, Fire Area Simulator–model development and evaluation*, vol. 3 (US Department of Agriculture, Forest Service, Rocky Mountain Research Station Ogden, UT, 1998)
8. R.G. Rehm, R.J. McDermott, *Fire-front propagation using the level set method* (US Department of Commerce, National Institute of Standards and Technology, 2009)
9. C. Lautenberger, Wildland fire modeling with an eulerian level set method and automated calibration, Fire Safety Journal **62**, 289 (2013)
10. A. Simeoni, in *SFPE handbook of fire protection engineering*, ed. by M.J. Hurley, D.T. Gottuk, J.R. Hall Jr, K. Harada, E.D. Kuligowski, M. Puchovsky, J.M. Watts Jr, C.J. WIECZOREK, et al. (Springer, 2015), chap. 87, pp. 3283–3302

11. A. Lattimer, J. Borggaard, S. Gugercin, K. Luxbacher, B. Lattimer, in *Proceedings of the 14th International Fire Science & Engineering Conference* (2016), pp. 305–315
12. W. Mell, M.A. Jenkins, J. Gould, P. Cheney, A physics-based approach to modelling grassland fires, International Journal of Wildland Fire **16**(1), 1 (2007)
13. Y. Safi, A. Bouroumi, Prediction of forest fires using artificial neural networks, Applied Mathematical Sciences **7**(6), 271 (2013)
14. M. Castelli, L. Vanneschi, A. Popovič, Predicting burned areas of for-est fires: an artificial intelligence approach, Fire ecology **11**(1), 106 (2015)
15. J. Storer, R. Green, in *Neural Networks (IJCNN), 2016 International Joint Conference on* (IEEE, 2016), pp. 676–683
16. H. Naganathan, S.P. Seshasayee, J. Kim, W.K. Chong, J.S. Chou, Wildfire predictions: Determining reliable models using fused dataset, Global Journal of Computer Science and Technology (2016)
17. Y. Cao, M. Wang, K. Liu, Wildfire susceptibility assessment in southern china: A comparison of multiple methods, International Journal of Disaster Risk Science **8**(2), 164 (2017)
18. R.J. McCormick, T.A. Brandner, T.F. Allen, Toward a theory of meso-scale wildfire modeling: A complex systems approach using artificial neural networks. Ph.D. thesis, University of Wisconsin–Madison (2001)
19. R.J. McCormick, in *ForestSAT Symposium, Edinburgh* (Citeseer, 2002)
20. A. Krizhevsky, I. Sutskever, G.E. Hinton, in *Advances in neural information processing systems* (2012), pp. 1097–1105
21. F.A. Albini, Estimating wildfire behavior and effects. Tech. rep., USDA Forest Service (1976)
22. P.L. Andrews, Modeling wind adjustment factor and midflame wind speed for rothermel's surface fire spread model, Gen. Tech. Rep. RMRS-GTR-266. Fort Collins, CO: Department of Agriculture, Forest Service, Rocky Mountain Research Station. 39 p. **213** (2012)
23. C.V. Wagner, A simple fire-growth model, The Forestry Chronicle **45**(2), 103 (1969)
24. D. Green, A.M. Gill, I. Noble, Fire shapes and the adequacy of fire-spread models, Ecological Modelling **20**(1), 33 (1983)
25. P.L. Andrews, Behaveplus fire modeling system, version 5.0: Variables, Gen. Tech. Rep. RMRS-GTR-213 Revised. Fort Collins, CO: Department of Agriculture, Forest Service, Rocky Mountain Research Station. 111 p. **213** (2009)
26. A.L. Maas, A.Y. Hannun, A.Y. Ng, in *Proc. icml*, vol. 30 (2013), vol. 30, p. 3
27. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1026–1034
28. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems (2015). URL https://www.tensorflow.org/. Software available from tensorflow.org