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

Wildland Fire Modeling Using Convolutional Neural Networks

Jonathan L. Hodges · Brian Y. Lattimer

Received: date / Accepted: date

Abstract This paper presents a novel predictive analytics approach to predicting the spread of a wildland fire using a convolutional neural network (CNN). Simulated fire perimeters for use in this process were generated at 6 hour intervals using the phenomological 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 predictions of fire perimeter 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. Although trained on predictions 6 hours apart, the CNN-based approach is shown to be capable of predicting fire perimeters further in the future by recursively using previous predictions as inputs to the model. The model was found to be primarily limited by low feature density in the input fire perimeter, typically from small fire perimeters resulting from low rates of spread in the simulations.

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

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; phenomological fire spread models are often used to predict spread across large domains [1]. Phenomological models are based on fitting experimental measurements to an expected functional form [2], such as the model of Rothermel [3,4] which uses empirical correlations for heat source and sink terms in conservation of energy [5]. Difficulties arise when modeling complex scenarios which do not fit the functional form for which the phenomonological model was developed. One of the key advantages of applying machine learning is the capacity of the model to learn an underlying functional form.

A first attempt at predicting the spatially resolved flame front of a wildland fire using predictive analytics was presented by McCormick. The model considers a 3x3 neighborhood of pixels to classify the center pixel as burned or unburned [6,7]. Although the premise of the work is interesting, the model is incomplete as the order pixels are considered by the neural network is based on the fire growth modeling by Finney which considers an ellipsoidal growth profile in the direction of wind [8]. The results show good spatial agreement between predicted and known fires; however, pixels from all 11 fires were used in training which begs the question how well the model would predict a fire which none of the pixels were used to train the network. An additional limitation of this model is the inability to predict the time-resolved fire front.

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 [9]. The objective of this study is to apply this type of framework to the predict the propagation of a wildland fire perimeter. As an analogy to image classification, data such as elevation, moisture

J. L. Hodges
Jensen Hughes
2020 Kraft Drive, Suite 3020
Blacksburg, VA 24060 USA
Tel.: +1 540-808-2800 x10611
E-mail: jhodges@jensenhughes.com

content, and wind speed can be treated as channels in an image. The features the CNN learns will be the relationships between input parameters. The CNN can then be trained on simulated data.

The objective of this study is to develop a fire spread model using a neural network which accurately predicts the spatial-temporal distribution of the fire front in a wildland fire 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 sensitivity of the network to each input parameter is examined, and the trained parameters of the network are used to infer relationships about input parameters. The work presented herein represents a proof-of-concept on a simple configuration with future work to expand the method to use experimental data with heterogenous spatial conditions.

2 Methods

The method presented herein considers each primary driver of wildland fire spread 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 a single channel corresponding to the fire perimeter after 6 hours. A schematic showing a sample image stack is presented in Fig. 1. The following subsections describe the simulation conditions and network architecture 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 [3,4,10]. In the Rothermel/Alibini phenomological model, the peak surface fire spread rate, $V_{s,peak}$ is calculated using the equation

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

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_s 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 [3,10], and 40 developed by Scott [4]. 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) [11]. Researchers have shown wildland fires grow in a generally ellipsoidal shape for homogenous spatial conditions based on $V_{s,peak}$ and wind speed [8,12,13]. For additional details on the empirical relationships and fuel models the reader is referred to the original publications.

The primary drivers in this 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 summarized 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

Custom software was developed to simulate the fire perimeter at 6 hours intervals for 10,000 different combinations of these parameters. Raster images of the fire perimeters were generated at a resolution of 1 pixel/km every 6 hours. Example fire perimeters from one simula-

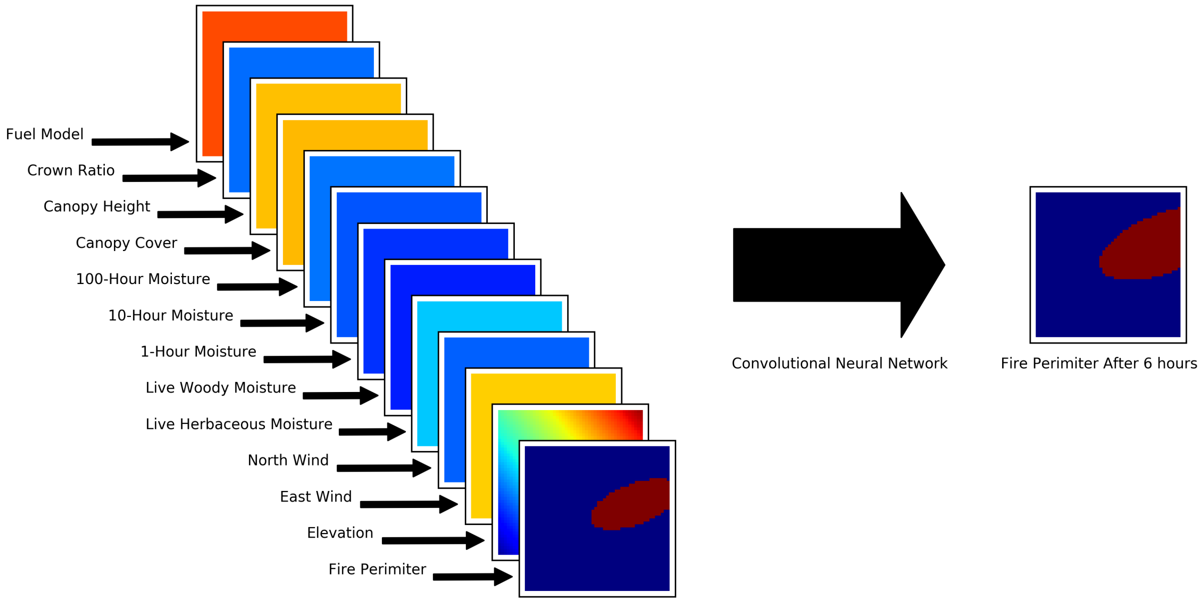

Fig. 1 Schematic of solution algorithm. The left set of images show the different channels used as inputs to the neural network. The values for each data channel are colorized based on the values shown in Table 1 and Table 2.

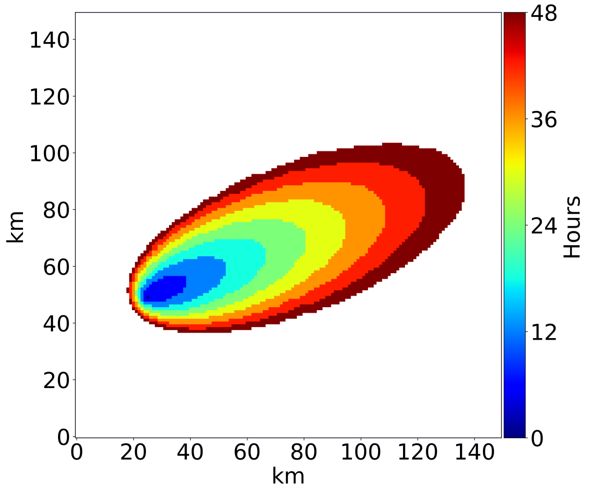

Fig. 2 Example Simulated Fire Perimeter

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

tion are shown in Fig. 2 for the parameter values shown in Table 2. The implemention of Rothermel's model in BehavePlus was used to validate the simulation framework [14].

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. This makes it possible to use deeper and more broad hidden layers without increasing computational requirements beyond what is feasible on current technology [9]. Representing the spread of a wildland fire front with a CNN is reasonable as wildland fire spread is a local phenomena [8].

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

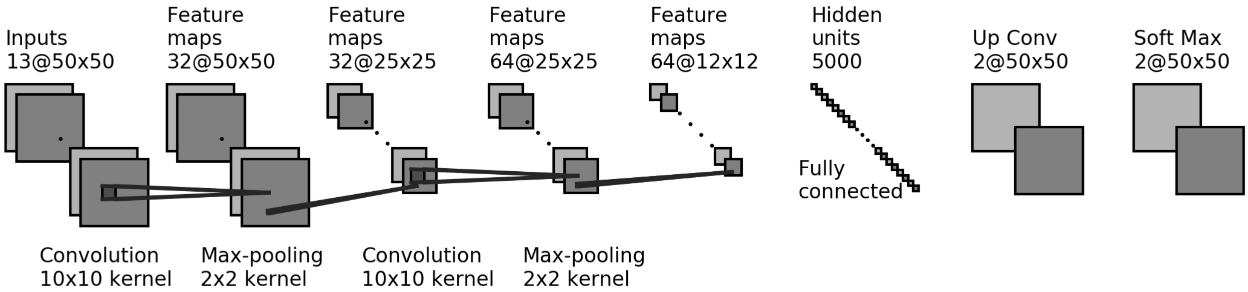

Fig. 3 Convolutional Neural Network Architecture.

in Fig. 1. The output image contains 50x50 pixels with two image channels corresponding to the probability the fire perimeter has reached a pixel and the probability the fire perimeter has not reached a pixel. A total of 6 hidden layers are included in the network including 2 convolutional, 2 max pooling, 1 dense classification, 1 up-convolution, and 1 soft max layers. The number of filters and step size in each convolutional and up-convolutional and number of neurons in dense layers were specified to steadily decrease the degrees of freedom from the 32,500 (50x50x13) in the input layer to the desired degrees of freedom of 2,500 (50x50,1) in the output layer. All hidden layers used a leaky rectified linear unit activation function except the fully connected layer which used a hyperbolic tangent activation function. The output layer used a soft max activation function to estimate the probability that each pixel would contain a fire. Over-fitting was reduced by using 50% dropout on the input layer. The cost function used in training was based on sum square error.

The network architecture was built using the Python 3 bindings for TensorFlow [15]. The network was trained using 27,000 samples (3 per parameter set) for 50,000 epochs using a single NVIDIA Quadro K620. The total time to train the network was 18 hours 7 minutes.

2.3 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 fire perimeter 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 is part of the fire perimeter. An example neural network prediction before and after post-processing is shown in Fig. 4. The impact of this threshold on the performance of the model is discussed in Section 4.

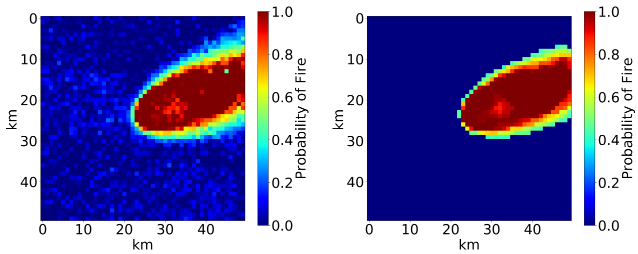

Fig. 4 Example CNN prediction of fire perimeter probability (a) Directly predicted from CNN (b) After post-processing

2.4 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 0 to 1, with a perfect score being 1. The precision, P, 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 defined as

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

where f_n is the number of fire pixels which were identifed as non-fire pixels. F-measure is, F, is the harmonic mean of P and S,

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

3 Results

The robustness of the neural network to predict new fires was examined by considering 3,000 test cases which were not included when training the network (1,000 parameter sets with predictions 6 hours apart). Sample CNN predictions from 5 of these test cases are compared with simulation predictions in Fig. 5. Figure 5a

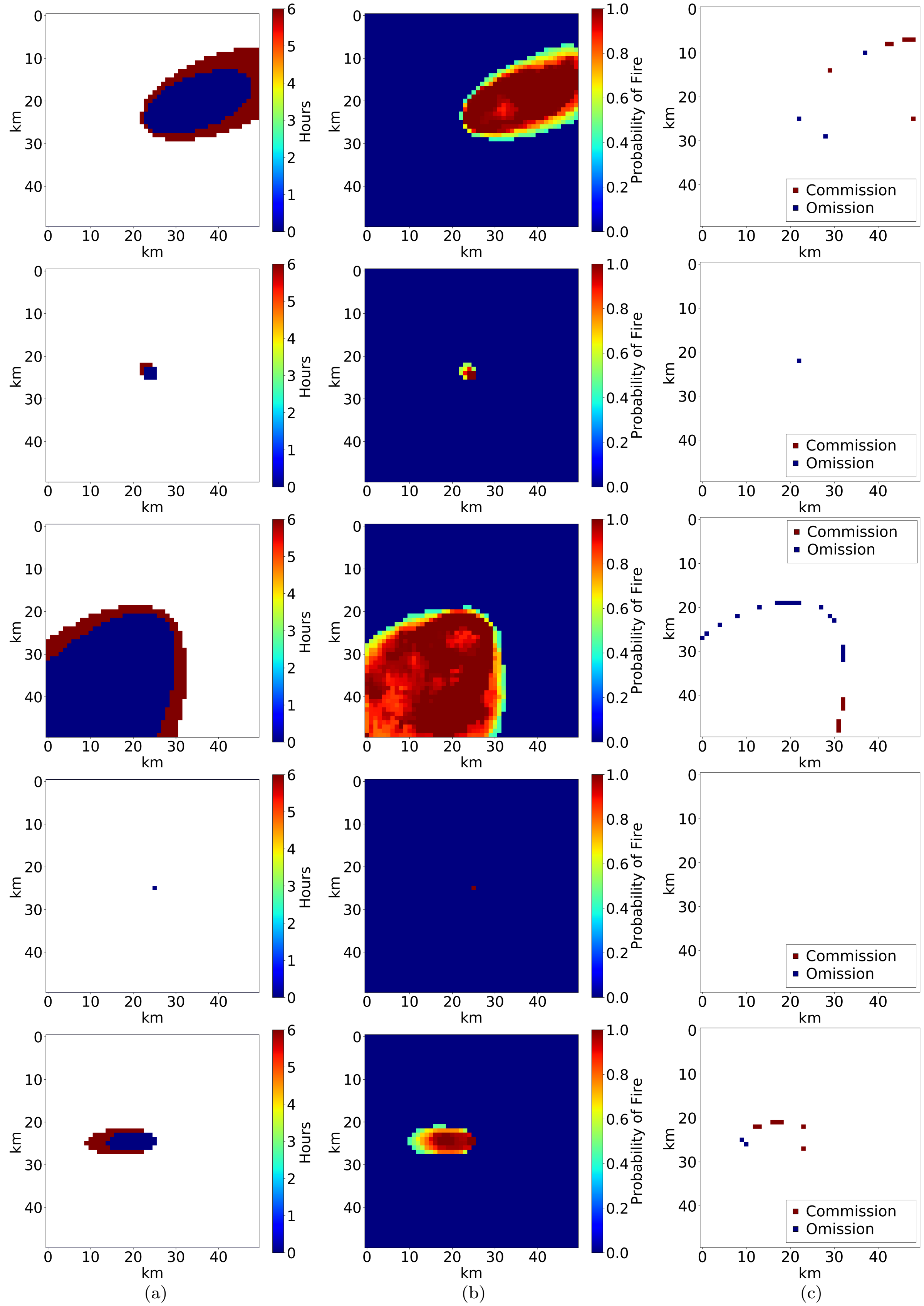

Fig. 5 Example CNN prediction of fire perimeters (a) Simulation fire perimiter (blue) input to CNN (red) expected fire perimeter after 6 hours (b) CNN prediction of fire perimeter (c) Classification error

shows the initial and final fire perimeter from the simulation. Figure 5b shows the final fire perimeter predicted by the CNN. Figure 5c highlights pixels which the CNN prediction did not match the simulation predictions. Pixels shown as red represent commission errors (false positive of fire), and pixels shown as blue represent omission errors (false negative of fire).

The mean and 80, 90, and 95-Percentile (X% of observations above) precision, sensitivity, and F-measure of the 3,000 test cases are shown in Table 3. The distribution of F-measure for the 3,000 test cases are shown in Fig. 6.

Table 3 Performance Metrics of CNN Predictions of Test Cases

Parameter	Mean	80%	90%	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

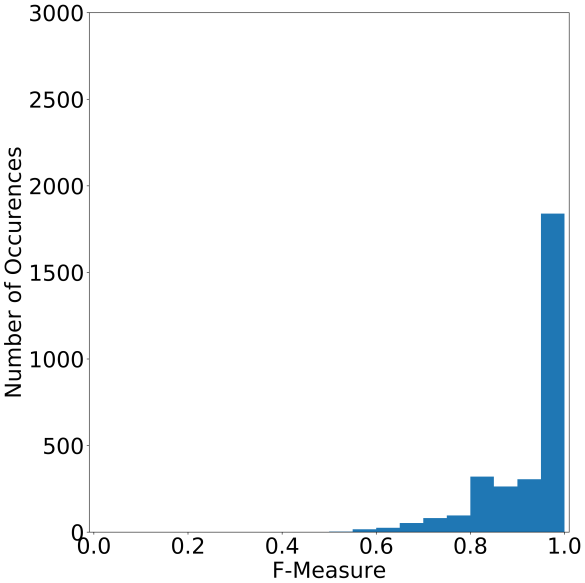

Fig. 6 F-Measure distribution of CNN Predictions of Test Cases.

4 Discussion

The overall shape of the fire perimeter predicted by the CNN is aligned with the simulations for the 3,000 test cases examined in this work. The fire perimeters predicted by the CNN do not contain non-physical holes or spotting. The direction of maximum growth is captured well. Figure 6 shows a sharp drop in number of occurrences at $F < 0.8$. Examining the cases where $F < 0.8$, it was found 82% had an initial fire size of 9 pixels or less, as shown in Fig. 7. Since a CNN relies on feature recognition, low feature density in the inputs leads to a decrease in the accuracy of the model.

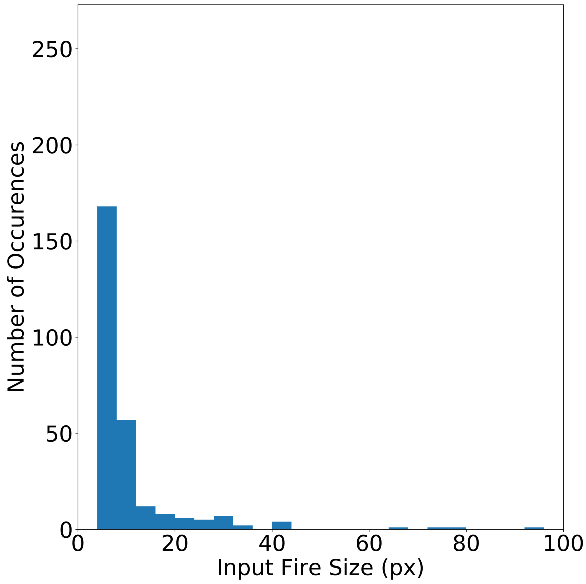

Fig. 7 Distribution of final fire size for all cases where $F < 0.8$.

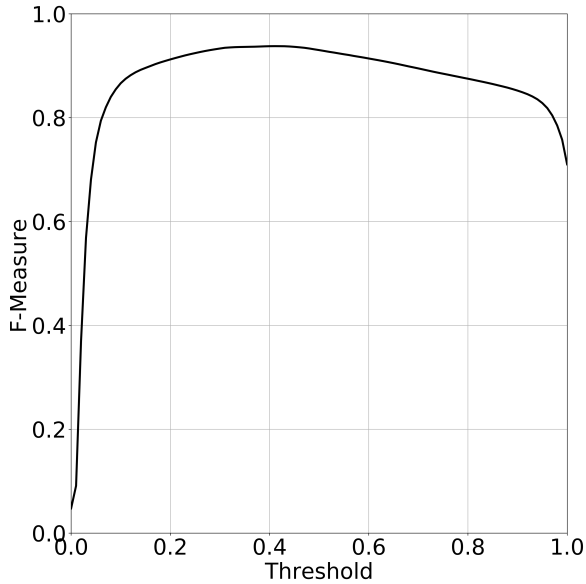

Fig. 8 Mean F-Measure of CNN predictions of 27,000 training data for different threshold values.

The optimal threshold to use in post-processing was determined by calculating the mean F-measure for CNN predictions of the 27,000 training cases with thresholds ranging from 0.01 to 0.99. The mean F-measure was found to be mostly independent of the post-processing

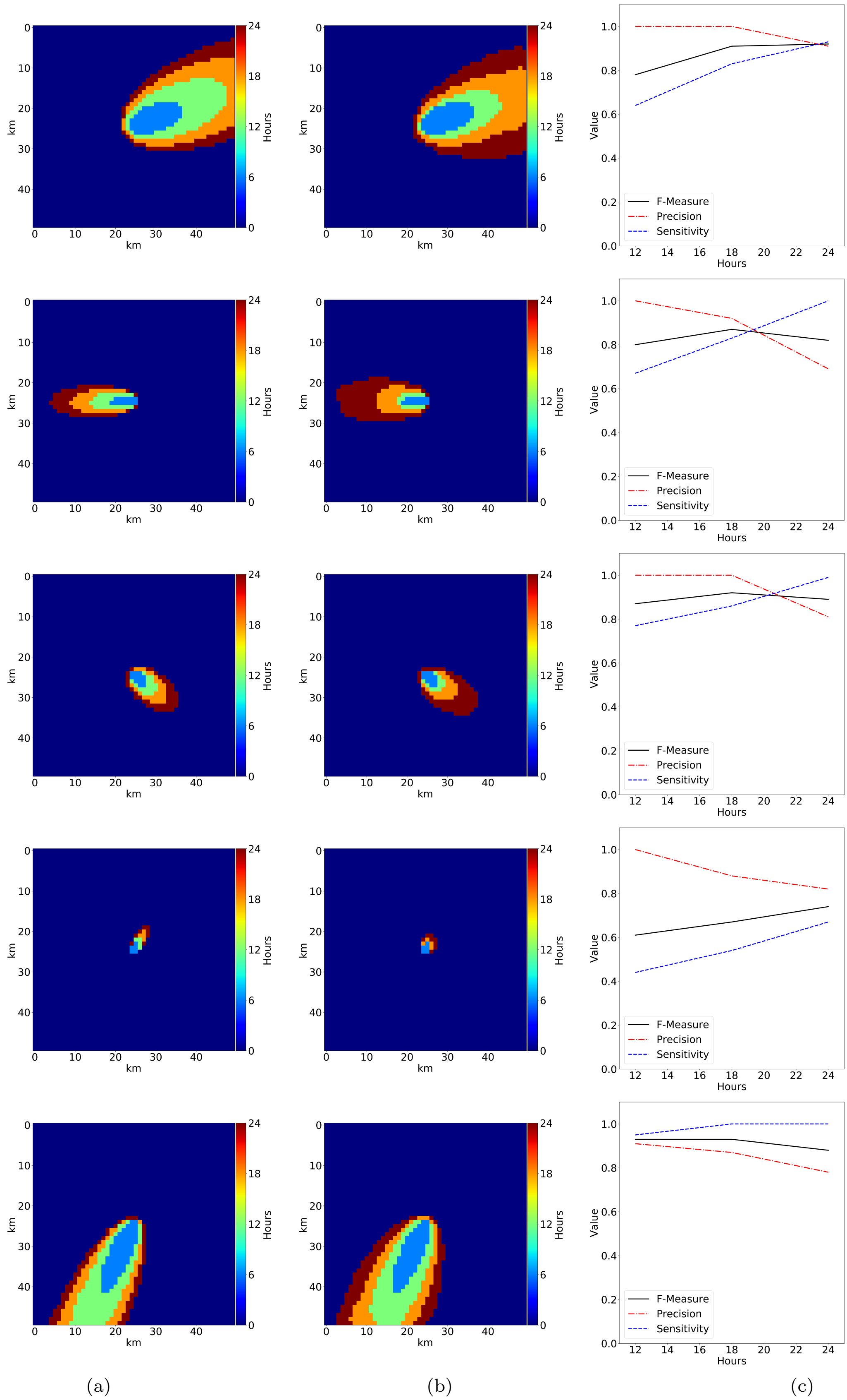

Fig. 9 Example CNN prediction of time resolved fire perimeters (a) Simulation fire perimiter (b) CNN prediction of fire perimeter (c) F-Measure at 6 hour intervals

threshold in the range of 0.2-0.6 as shown in Fig. 8. The maximum mean F-measure of the training data was calculated with a threshold of 0.41. This threshold was fixed and used when analyzing the test cases.

The model was trained using a 6 hour time interval between the input and output fire perimeters. It is possible to obtain predictions at points further in the future at 6 hour intervals by recursively using the previous prediction as an input to the CNN. Figure 9 shows 5 example cases where this process was used to predict fire perimeters up to 24 hours from ignition based on an input fire perimeter 6 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. Since precision describes commission errors and sensitivity describes omission errors, this shows early in the progression of the fire the model under-predicts the rate of spread, but later on the model over-predicts the rate of spread. This highlights the difficulty the CNN can have when dealing with low feature density.

5 Conclusion

A novel predictive analytics approach to predicting the spread of a wildland fire using a convolutional neural network (CNN) was presented. The robustness of the approach was tested using 3,000 test cases which were not included when training the network. The predictions of fire perimeter from the CNN based approach agreed with simulation results, with a mean F-measure of 0.93. Cases where F-measure was observed to be less than 0.80, 82% contained less than 9 pixels in the input fire perimeter. Although trained on predictions 6 hours apart, the CNN-based approach is capable of predicting fire perimeters further in the future by recursively using previous predictions as inputs to the model. The model was found to be primarily limited by low feature density in the input fire perimeter, typically from small fire perimeters resulting from low rates of spread in the simulations.

This work represents a first step in creating a framework to predict wildland fire spread without physics based models. Although the data used to train the CNN in this work was generated using a phenomological model, the model does not contain any concept over whether the data is from a computational fluid dyanmics model, phenomological model, or even experimental measurements of fire perimeter. Additionally, although the simulations used were generated using 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. A. Sullivan, A review of wildland fire spread modelling, 1990-present 1: Physical and quasi-physical models, arXiv:0706.3074v1[physics.geo-ph] (2007)
2. A. Sullivan, A review of wildland fire spread modelling, 1990-present 2: Empirical and quasi-empirical models, arXiv:0706.4128[physics.geo-ph] (2007)
3. R.C. Rothermel, et al., A mathematical model for predicting fire spread in wildland fuels. Tech. rep., USDA Forest Service (1972)
4. 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)
5. R. Weber, Modelling fire spread through fuel beds, Progress in Energy and Combustion Science **17**(1), 67 (1991)
6. 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)
7. R.J. McCormick, in *ForestSAT Symposium, Edinburgh* (Citeseer, 2002)
8. 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)
9. A. Krizhevsky, I. Sutskever, G.E. Hinton, in *Advances in neural information processing systems* (2012), pp. 1097–1105
10. F.A. Albini, Estimating wildfire behavior and effects. Tech. rep., USDA Forest Service (1976)
11. 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)
12. C.V. Wagner, A simple fire-growth model, The Forestry Chronicle **45**(2), 103 (1969)
13. D. Green, A.M. Gill, I. Noble, Fire shapes and the adequacy of fire-spread models, Ecological Modelling **20**(1), 33 (1983)
14. 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)
15. 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