Predictive Models of Fire via Deep learning Exploiting Colorific Variation

Jiseong Han
Department of Applied Statistics
Konkuk University
Seoul, South Korea
wltjd802@gmail.com

GuangSu Kim
Depertment of Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
s88012@kaist.ac.kr

Chanseo Lee
Department of Statistics
Keimyung University
Daegu, South Korea
chunsang16@naver.com

Ung Hwang
Department of Electronics and Computer Engineering
Hanyang University
Seoul, South Korea
gogowinner@naver.com

Yeongkwang Han
Department of Statistics
Keimyung University
Daegu, South Korea
sundayhiphop@daum.net

SungHwan Kim*
Department of Applied Statistics
Konkuk University
Seoul, South Korea
shkim1213@konkuk.ac.kr

***Abstract*— Predictive models on fire have been increasingly popular in computer image analysis. Due to late strides of deep learning techniques, we are now unprecedently benefited from its flexible applicability. In most cases, however, the conventional algorithms are limited to only single-framed images unlike sequence data that inevitably entails heavy computational time and memory. In this paper, we propose an effective algorithm exploiting the combination of CNNs (convolution neural networks) and RNNs (recurrent neural networks) in a consecutive way so that sequence data can be allowed for the model. The LSTM (long short-term memory) is well-known to be superior to other RNN-type algorithms in accuracy, especially when applying to sequence data. In our extensive experiments, where fire videos (e.g. indoor fire, forest fire) and non-fire videos collected from a range of scenarios are taken into accounts, it is confirmed that our propose methods are found outstanding in predictive power.**

Keywords—fire detection, deep learning, video analysis, image classification,

I. Introduction

It is commonplace to utilize colorific variations in image analysis and vision-based inference (Closed Circuit Television; CCTV) known to be suited to unfortune risks. To respond to medical emergency, fight, natural disaster such as fire accidents and flood, it is critical to alarm in a timely manner at the earliest possible time. Such rapid response can potentially minimize the chances of inevitable loss, and thereby can control an irresistible events. Fire accidents occur in our daily life and importantly detection at early stages is imperative to circumvent fatal damages. [1] And yet current fire alarm systems mostly focus on preventing damages rather than detecting early fire. Generally, the systems operate based on sensors and are limited in scope to indoors. In principle, they are purposely designed to detect the presence of smoke particles revealed by ionization and a proximity to the fire. [2]

To our surprise, this system was found less efficient due to high false discovery rate and suffered from wasting social resources. Even worse, false negative (e.g., missing fire) or late detection may possibly cause enormous social and human losses. Besides, as forests, farms, buildings or oil tanks are identified at ease via current techniques. [3] In virtue of camera and video processing techniques, computer vision-based fire detection systems have been increasingly popular and superior to sensor-based systems. It is common that existing fire detection algorithms via a single image suffer from false detection[4], especially in case of feather appearance (e.g., fire like flicker lighting). This challenge makes the algorithms limited to use in practice; hence to this end, it is worthwhile to consider a fire detection algorithm that builds on multiple video sequences. As related to all concerns above, in this paper we propose an algorithm, say, “Deepfire” that exploits image sequences (to put it another way, video clips). In theory, the rule takes advantages of the cutting-edge deep learning techniques in collaboration on both recurrent neural nets (RNN) including but not limited to the convolutional neural nets (CNN) algorithm that learns on a bulk of real fire images sequences. To assess practical utility, we tested if the proposed algorithm performs robust under various scenarios (e.g., flickering electronic signs, dancing in red color top, running people, CCTVs etcs). [5]. Taken together, the proposed algorithm is found to be outstanding in accuracy, suggesting the chance of applicability. The paper is outlined as follows: In Chapter

2, we discuss the method to classify an image. Experimental results and discussions are given in Chapter 3. Finally, we conclude in Chapter 4.

II. METHOD

A. Related works

Long Short-Term Memory networks (LSTMs) are a variant of RNNs whose architecture flows learn on long-term dependencies, initially introduced by Hochreiter & Schmidhuber (1997) [6]. LSTMs are designed on purpose to avoid the long-term dependency problem. Similar to other RNN-type algorithms, the LSTMs also run on the chain structures distinctively featured with, say, the forget mechanism. This mechanism accommodates the function of sigmoid and hyperbolic tangent so that working data can add or remove in part. More precisely, LSTMs enable to remove (forget) or add (remember) information to cell states. These cells are placed on interactive in structure, making it possible to maneuver long sequence models.

CNNs lay a foundation on the sequential combination of three layers: convolution layers, pooling layers, fully-connected (FC) layer. Convolution layers serve as the core building block of the CNNs, and are designed to convolve its input with a bank of kernels. Importantly, it is known that this convolution promotes robustness to image recognition [7,8]. Pooling layers are used to sub-sample inputs, each consisting of convolution outputs. There are a range of pooling strategies (e.g., max-pooling, average pooling method and etcs). In our experiment, we adopt the max-pooling method taking the maximum value identified from an unit block of image sequences. The fully-connected layer combines the output of the topmost convolution layer into 1-dimension features, and put them together to determine class labels. [8].

B. Proposed Algorithm

The convolution neural networks (CNNs) is one of the most powerful predictive rule, in particular specialized in classifying images. And yet, this is limited in scope to non-sequential data, originally not allowing for temporal domain. To address this challenge, we take a forte of the recurrent neural networks (RNNs) that decipher temporal effects in the midst of sequential information. In principle, the RNN can accept not only an immediate input vector fed in at present but also ones that lie time ahead. [9] In our research, we integrate both the special type of RNN, namely LSTM and novel CNNs: (Step 1) Use CNNs to generate learning data: we use CNNs to make layers, and they are passed to RNNs. We use the Nielsen net [10] to make CNNs layers.

TABLE I. THE PROPOSED ARCHITECTURE OF SIMPLIfiED CNN MODELS.

type	patch size / stride	input size
conv	10x10 / 5	100x100x3
Max-pool	2x2 / 4	25x25x10
conv	5x5 / 2	24x24x5
Max-pool	2x2 / 2	12x12x5
Fully-connected	-	1x720
softmax	classifier	1x1x2

We then split a video running over ten seconds into individual frames (commonly, 30 frames per second), and equivalently obtain 300 frames in total). In an effort to generalization, we carve out images of 100×100 as size of video sequences is different one another. Putting all sequence together, we generate the convolution layers, pooling layers and fully connected layer. Training data include 71 totally different videos of fire, 71 non-fire videos (but it appears to be fire like flickering electric light) and 156 fire videos whose condition changes on the way from non-fire to fire. Integrating up all, we glean 37,500 images training, and randomly divide 32,500 images for training and 5,000 for test images to circumvent over-fitting. (Step 2) Transfer our train and test data to CNNs: we can pass the fully-connected layers of the CNN, the convolution features at preceding stage to the last, to the RNN. It is important to note that this step is not a terminal prediction. (Step 3) Convert the output of individual frames into sequences of frames: we aim that RNNs exploit the sequence features of video that CNNs produce in spirit of a specific feature of images. (Step 4) Train the new RNN on the train set: we generate a CNNs feature data on which RNNs learn to model. (Step 5) Evaluate result: Toward this end, we can evaluate the model inference via test images. To catch a glimpse, Table 1 encapsulates the outline of the Deepfire workflows.

TABLE II. THE WORKFLOWS OF THE DEEPFIRE ALGORITHM

1. Initialize model configurations:
   1) Convolution: five consecutive convolutions layers.
   2) Pooling: sample the maximum values aligned onto filters.
   3) Fully connected network: create a CNN feature vector X_i of 720 elements from a fully-connected layers, and $\left\{C_i^{(r)}\right\}_{125\times 720}$ that combines X_i across all video frames for $i = 1,2,\dots,125$ and $r = 1,2,\dots,596$.
2. Input $C_i^{(r)}$ into the LSTM algorithm, where the number of cells is 128.
3. Predict the labels of the sequence array $C_i^{(r)}$ either fire or non-fire.

III. RESULT

In this section, we assess our predictive models using video data, each containing videos of fire, 71 videos of non-fire, and 156 videos of non-fire to fire as in Fig 1 and

2. We learned a training model, and applied the entire data to compute validation accuracy. Regarding training models, we adopted three difference algorithms; only CNNs, LSTMs, and CNNs with G variation. In case of LSTMs, we use deep neural network model to make LSTMs implemented by R-packages ("tflearn"). Shown in Table III, CNNs shows 7.89% of false positives and 8.47% of false negatives. Total accuracy is 92.08%. It takes 60.195 seconds per epoch on average. On the other hand, LSTMs perform with higher accuracy than CNNs method, producing 8.05% of false negatives and 7.38% of false positive. Total accuracy is 92.28%. Importantly, this method run faster than CNNs method. The average computing time per epoch is 6.73 seconds. It is confirmed that G layers come into play with distinguishable features[5]. With that known prior, we make 4-channel CNNs architecture that contains G-variation to compare to our algorithm. It has 93.75% of accuracy, more precisely with 5.78% of false positives and 7.20% of negatives. However, it takes 242.67 seconds per epoch on average.

TABLE III. COMPARISION OF THREE METHODS

Method	***Epochs***	***Accuracy (%)***	***False Positives (%)***	***False Negatives (%)***
CNNs	200	92.08	7.89	8.47
LSTMs	200	92.28	7.38	8.05
CNNs 4-Ch	200	93.75	5.78	7.20

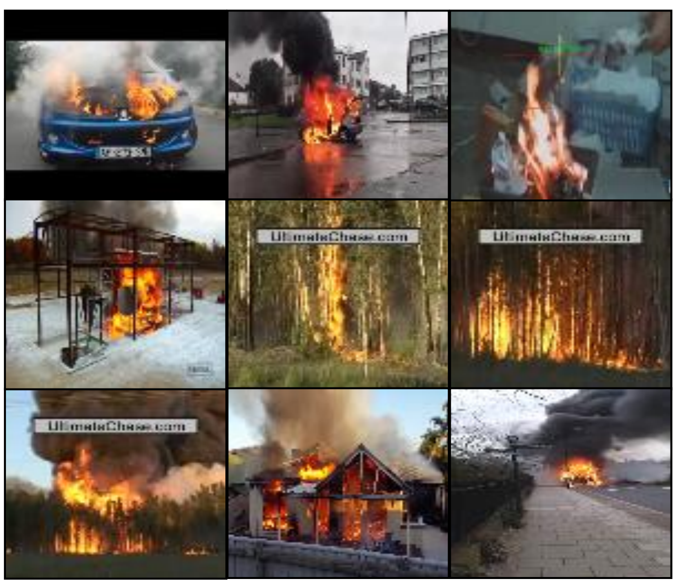

Fig. 1. The thumb nails of fire videos

IV. CONCLUSION AND DISCUSSION

Up to date, research on fire detection have been increasingly prevalent and successful in many ways. And yet, it common that conventional algorithms fail to effective prediction on video sequence data. In this paper, we propose the algorithm 'Deepfire' to detect fire in video sequence in collaboration with both convolution neural networks (CNN) and recurrent neural networks (RNN) simultaneously. The proposed methods outperform a variant of CNNs in accuracy and computing cost. In terms of False negative, CNNs exploiting G variance perform best, and computing cost was 4 times slower than CNNs on average. It is important to see that LSTMs, however need high volumes of video data to refine classifier, but still suffer from false fire alarms to some extent. In future work, (1) we can compare convolution neural networks building on 3D data, allowing to handle with video in context of CNNs. (2) inspired by significance of G values, we improve Deepfire algorithms that adopt in the model diverse color variations of fire.

REFERENCES

[1] Khan Muhammad, (Student Member, IEEE), Jamil Ahmad, (Student Member, IEEE), Irfan Mehmood , (Member, IEEE), Seungmin Rho, (Member, IEEE), and Sung Wook Baik, (Member, IEEE), Convolutional Neural Networks Based Fire Detection in Surveillance Videos

[2] Sebastien Frizzi, Rabeb Kaabi, Moez Bouchouicha, Jean-Marc Ginoux, Eric Moreau, Farhat Fnaiech, Convolutional Neural Network for Video Fire and Smoke Detection

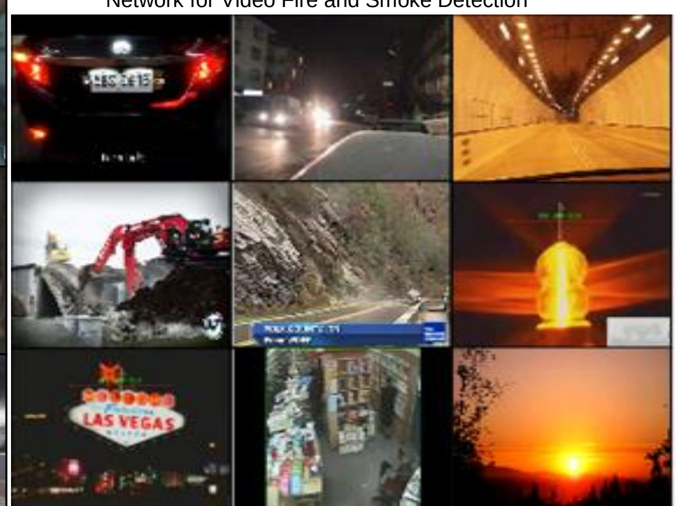

Fig. 2. The thumb nails of non-fire videos

[3] Dongqing Shen, Xin Chen, Minh Nguyen, Wei Qi Yan, Flame Detection Using Deep Learning, 2018 4th International Conference on Control, Automation and Robotics

[4] Celik, T. Fast and efficient method for fire detection using image processing. ETRI, 32(6), 881-890, 2010.

[5] Ung Hwang, Jechang Jeong and SungHwan Kim, Computer Vision-based Method to Detect Fire Using Color Variation in Temporal Domain

[6] Sepp Horchreiter, Jürgen schmidhuber, Long Short-Term Memory, 1997

[7] Sebastien Frizzi, Rabeb Kaabi, Moez Bouchouicha, Jean-Marc Ginoux, Eric Moreau, Farhat Fnaiech, Convolutional Neural Network for Video Fire and Smoke Detection

[8] DanC.Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, JürgenSchmidhuber, Flexible, High Performance Convolutional Neural Networks for Image Classification, Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence

[9] Zachary C. Lipton, John Berkowitz , A Critical Review of Recurrent Neural Networks for Sequence Learning, June 5th, 2015

[10] M. A. Nielsen, “Neural networks and deep learning,” 2015.