Dynamic Feature Extraction Using I-Vector for Video Fire Detection

Zhongming Huang[1]*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
reavenhuang@163.com

Yuxiang Wang[1]*
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
Euson@outlook.ie

Haolan Hu
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
haolanhu_2019@163.com

Xun Liu
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
vickibaileyraymond@gmail.com

Tongzhen Liu
School of Electronic and Information Engineering
Tiangong University
Tianjin, China
liutongzhen0311@126.com

Zhanxu Zhang
School of Computer Science and Software
Tiangong University
Tianjin, China
zhangzhanxu23@outlook.com

***Abstract*—Fire detection technology has been researched and developed for decades. However, in videos and complex scenes, it still lacks fast recognition of fire's existence. The traditional model of fire recognition still needs a large number of samples and time-consuming machine learning progress. Meanwhile, the uncertain shape of fire leads to the reduction of accuracy using CNN. Based on the above problems, we establish a novel method based on I-Vector. We use an adapted I-Vector algorithm to extract the time sequence feature vector on the fire and its surroundings and train a G-PLDA classifier to recognize the dynamic occurrence of fire more quickly and accurately. This model requires fewer samples and a shorter learning time while obtaining an accuracy similar to the traditional fire recognition models, which provides a new effective solution for rapid analysis of whether there is a fire in the video scene. In addition, the algorithm has a sound universality and is easy to deploy in the application fields of video fire supervision, UAV fire inspection, and other related fields.**

Keywords—video fire detection, pixel sampling, sequence feature extraction, signal processing, I-vector

I. INTRODUCTION

In recent years, visual recognition has made great progress in both theoretical and practical applications. Among them, popular applications are progressing in face recognition [1], fingerprint recognition [2], fire recognition [3] and other related fields. Most of the methods for fire recognition in videos are based on convolutional neural networks (CNN) [4], which classify and detect images or videos input by the network, and use frame-by-frame detection methods for videos to detect whether a fire exists. However, this method cannot find out whether there is a fire in a video at the first glance, and it is followed by many confinements including an enormous dataset, time-consuming learning period, and unreliable recognition results due to the uncertainty of shapes of fire, as shown in Fig. 1. In dealing with the problems, it is necessary to develop a video fire detection algorithm that is not dependent on static fire morphology features. After long-term observation, we discover that the flickering of a fire has certain dynamic characteristics when burning. We come up with the idea of developing a feature extraction algorithm to look into the dynamic patterns of fires. When a video is separated into pixels along the time domain, each pixel will change its value as the frame goes with timeline. This means we can cut a video along the time domain, resulting in time sequences of correlating pixels. For pixels on the fire, they carry unique patterns of brightness change. Our algorithm uses the greyscale time series extracted from a video at the fire pixel points and the surrounding environment pixels to learn to differentiate unique patterns of fires. Extracting features from time sequences instead of performing frame-wise convolution enables the model to quickly find fires in videos. With a proper selection of feature extraction algorithm, our method would be lightweight and less dataset-dependent compared with CNNs, making it conducive to fire detection.

Before I-Vector is selected as our baseline feature extractor, we have compared it with other two commonly used time series feature vector extraction algorithms. But it turns out that I-Vector has the best adaptability on fire detection and is easier to use. We use the I-Vector algorithm [5] as a baseline to establish the model, where I-Vector is originally a kind of time-series feature processing algorithm commonly used in speaking verification technology and speech analysis. We first establish a dataset containing 1298 time sequences in greyscale, in which fire samples (positive samples) take up 3/4. Then an I-Vector extractor and a G-PLDA classifier are trained one after another. This method reduces dataset size and computational complexity, while keeping a performance similar to that of a CNN. And it is found that our model has a shorter learning period, faster processing speed, and decent accuracy. From the test results, our model can effectively and quickly decide whether a fire is in a video, which contributes to the rapid solution for problems caused by fire based on video records. Due to its good adaptability, our algorithm is also relatively stable.

Our novel time-series feature extraction algorithm based on I-Vector can be used as a supplement or even an alternation for

This is part of the research under the Tianjin Provincial University Student Innovation and Entrepreneurship Program (Project No.202110058107), which all the authors are affiliated to.

current fire recognition algorithms. The fire recognition model established by our algorithm can be widely used in fire spotting in videos, fire time confirmation in monitor records, and other fire detection within videos.

Fig. 1. Fire scenarios: Fire or fire does not have a fixed shape, but the dynamic feature during combustion can be extracted.

II. Related Work

CNN: Convolutional neural network has been widely used in various target recognition scenarios. The mainstream method of CNN to recognize fire is almost to build a neural network and learn the dataset which includes images labelled as fire or non-fire. After repetitive training, a model that can recognize static fire is finally obtained [6] [18]. Arpit Jadon et.al. [7] proposed a specially modified object detection CNN called FireNet for fire recognition, and Shixiao Wu et.al [8] compared performances between many prevailing CNNs tuned for fire detection. Although mainstream convolution neural network models have the potential to achieve high accuracy in recognizing fires in images, they still have limitations. To discuss drawbacks of fire detection CNNs, we classify them into two types. First, for the Object Detection Networks [9], by inputting images or videos, it will output a bounding box confining the fire. However, the accuracy of the algorithm is limited by the scale of dataset. And their work also indicates high dataset dependence and computational complexity. Second, for the Instance Segmentation Networks [10], by inputting images or videos, it will output pixel-wise segmentations for objects. However, it would take great efforts to label the training data. So far, it still lacks of dataset and practical applications in fire detection. Moreover, in practical applications, fires are continuously varying in shape. But CNNs perform frame-by-frame detection, leading to their inability to utilize dynamic features of fires.

Time series processing: To find a feature extractor best suits fire time sequence, we compared three frequently used in sequence feature extracting. First, we looked into GMM-UBM model [11], a combination of Gaussian Mixture Model and Universal Background Model. The GMM-UBM model improves upon the original GMM model by adding background information, thus enhancing the ability for the extractor to focus on the target features. Second, we evaluated the Joint Factor Analysis method as a front end for feature extraction [13]. According to experiments by Kenny P. et.al [14], JFA has better reliability when faced with noisy information and gains purer feature extraction results. Though the above methods have many advantages, they are facing the challenge of I-Vector [15], a better refined and effective method to extract and present high dimensional features in a more compact dimensionality. To absorb the advantages of those methods, it is effective for us to choose GMM-UBM as the front end of I-Vector [16]. With the help of UBM, the feature extractor can more easily spot target features, and I-Vector further carries dimension reduction to output highly compact feature vectors.

Our contribution: Based on the limitations of CNNs and advantages of I-Vector, we propose a novel method that is not dependent on huge datasets or exact shapes of a fire. We use GMM-UBM model as the front end [9], it is followed by an I-Vector feature extractor. At the back end, we choose a G-PLDA classifier [17] to differentiate fires from the background. Details are to be discussed in the next section.

III. Our Method

In this section, we will introduce the structure and principle of our algorithm in detail.

A. Sampling Method

The combustion process of a fire is dynamic. Through this feature, we consider the viability to collect the flickering pattern of each sampled pixel as a time sequence for feature extraction, thus recognizing the fires in a video.

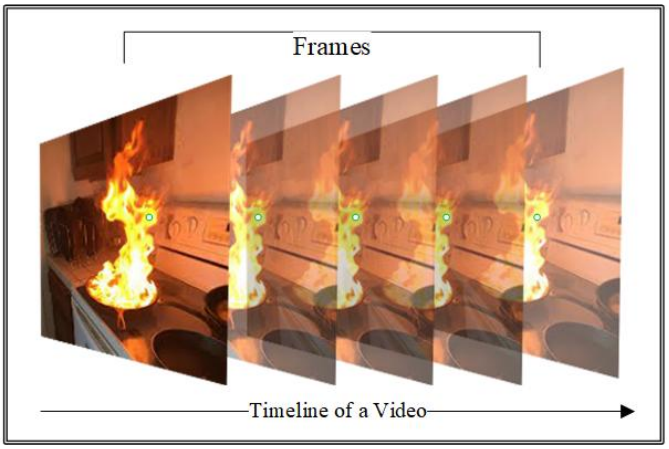

Fig. 2. Sampling method: The green dot is the pixel to be sampled. The sample pixel's brightness value of each frame will be recorded in a sequence.

As Fig. 2 shows, we first locate the pixel to be sampled. Second, considering every sample pixel in the video frames, we record each brightness value in greyscale, to obtain time sequences containing dynamic features. We sample pixels on the fires to accumulate positive instances, while sampling pixels on the surroundings or background to collect negative instances

B. I-Vector Algorithm Baseline

Inspired by [16], we consider that I-Vector can be used to represent the feature of each sample sequence. Firstly, we roughly extract the MFCC feature vectors of each fire and non-fire sample. Then, the MFCC is utilized by GMM-UBM (Gaussian Mixture Model-Universal Background Model) [11][12], and then a large number of data are collected to form further feature extractions. In GMM-UBM, the state occupancy of the Gaussian component of the signal at each time will be calculated. Input MFCC vector of non-fire sequence to GMM-UBM model, the mean value in each Gaussian classification

clutter is m and the Gaussian mean supervector of non-fire UBM can be calculated. After that, Input the MFCC vector of fire sequence to GMM-UBM model, and adaptively obtain M which is the Gaussian mean supervector of fire GMM through MAP algorithm. Meanwhile, the Baum-Welch statistics should be carried out. Assume C to be the number of Gaussian components. The algorithm formula to calculate Baum-Welch statistics shows as Eq1. And Eq2. [16]. At every moment t, $\gamma_t(c)$ is the state occupancy of γ_t in each Gaussian component c, that is, the information γ_t falls into the posterior distribution of the Gaussian component c at time t. And $N_c(s)$ represents the occupancy rate of the given information s falling into the Gaussian model c. The word information mentioned above may refer to time sequences as speaking voices or fire combustion patterns in the time domain, which will be further discussed later.

$$N_c(s) = \sum_t \gamma_t(c) \tag{1}$$

$$\gamma_t(c) = \frac{\pi_c p_c(Y_t)}{\sum_{j=1}^{C} \pi_j p_j(Y_t)} \tag{2}$$

After having Baum-Welch statistics, we collect the mean vector p_i of each Gaussian distribution:

$$m = \left[p_1^T, p_2^T, \ldots, p_C^T\right]^T \tag{3}$$

The target equation of I-Vector extraction is:

$$M(s) = m + T\omega(s) \tag{4}$$

In Eq.4, s is the extracted sequence (in NLP, a sentence; in our work, a pixel sequence of fire or non-fire). The $\omega(s)$ is the I-Vector of a given sequence s, and is also a hidden vector to be solved after having the global spatial difference matrix T. Assume D to be the dimensions of acoustic features extracted from the sequence s, and R to be the dimensions of $\omega(s)$. Hence, the dimensionality of T is $C \times D \times R$.

The global spatial difference matrix T will be obtained using EM adaptive iteration. First, initialize T randomly and bring in Eq.3 to obtain a posterior distribution of the hidden parameter $\omega(s)$. Repeat steps E and M until T meets an appropriate termination threshold. In this process, m will eliminate redundant factors by continuously reducing the dimension in the subspace. When T is obtained, the model parameters reached a global difference space matrix. Thus, the ω with a lower dimensionality is obtained, it carries the difference between classes and the difference between the channels carrying class information. The posterior mean of ω is I-Vector. I-Vector contains the difference between fires and the background environment, as well as the discriminations between sample videos. As Figure 3 shows, the fire flickering pattern is represented by the time series of the correlating pixel-wise greyscale amplitude. We then sample those time series for training and detection.

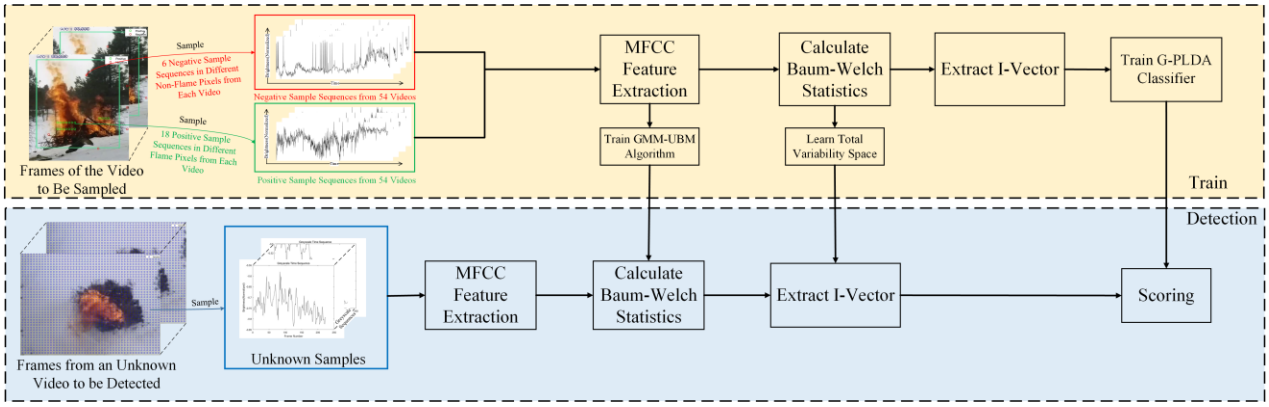

Fig. 3. Our algorithm: a) Training. Fire and non-fire pixels are respectively sampled and labelled into positive and negative samples. All the samples are in the format of time sequences shown in the above coloured squares. Features are extracted from the times sequences to train the I-Vector Extractor and G-PLDA classifier; b) Detection. Pixels of an unknown video are uniformly sampled as the blue circles indicate. Each sampled sequence will have its I-Vector. Then, the I-Vectors of the unknown pixels are scored by the G-PLDA classifier. A threshold in the classifier will determine the output class.

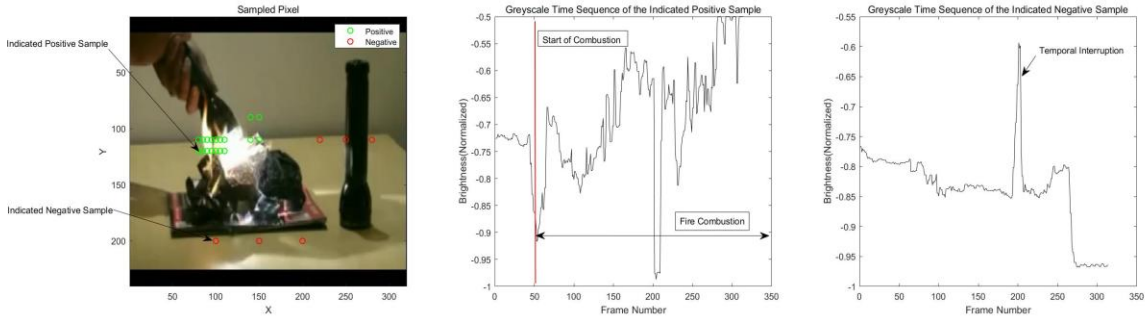

Fig. 4. The sampling of time sequences: it shows one of the 54 videos and its manual sampling. Left) select 24 sample pixels in each of the 54 videos, where 18 pixels are positive instances (coloured in green) and other 6 pixels are negative instances (coloured in red); middle) the time sequence of the arrow indicated positive sample pixel (in greyscale); right) the time sequence of the arrow indicated negative sample pixel (in greyscale).

C. Fire Feature Extraction Using Adapted I-Vector

We transform the original I-Vector algorithm to be applied to fire dynamic feature extraction. In the video to be sampled, the algorithm will collect the specified pixels of a series of time sequences as shown in Fig. 3 and result in sequences $S = [s_1, s_2, \dots, s_{24}]$, which to be fed into Eq.1 and Eq.4. And each sequence s_i has the dimensionality of $1 \times F$, where F stands for the length in frames of a video. When the fire occurs on a certain pixel, its greyscale value will show a specific pattern. Though other background parts may flicker as well, they can be discriminated by the GMM-UBM model at the front end of I-Vector extractor. Thus, we use both fire and non-fire sequences for training this I-Vector extractor by using fire greyscale series as the positive samples while the others as negatives (i.e. background).

Through sampling and training this way, if the greyscale of all or almost all acquisition pixels change greatly, our algorithm will believe that the overall brightness change is caused by other factors that are not relevant to the occurrence of fire. In this way, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) of our algorithm can be greatly reduced.

D. G-PLDA Classifier

Different from the frame-by-frame processing based on a convolutional neural network, the time sequence based on the dynamic characteristics of fire takes the whole video as input. Before G-PLDA scoring, the I-Vector has already been acquired. Then, the G -PLDA classifier intakes the I-Vectors of each sequence and outputs correlating scores. After training, the G-PLDA classifier can score unknown sequences and judge whether some of them contains fires, where the judgement is based on a well-tuned threshold. According to the classification results, we know whether the unknown subject of certain corresponding pixels is burning, thus locating the fire in the video scene. This classifier is rigorous and accurate. Compared with traditional algorithms and neural networks, it not only reduces a lot of computation and speeds up the operation speed, but also greatly improves the robustness of recognition.

IV. Experiment

In this section, we will show how the proposed method is achieved and the experiment results comparing our proposed method with several mainstream approaches in fire detection.

A. Dataset

Unlike the routine of labelling bounding boxes used in CNNs, the proposed method needs time sequences for training and detection. Therefore, we do not consider the sampling and labelling in each frame of the videos, but labelling the collected time sequences. We first select 54 different videos containing at least one clutter of fire combusting at a relatively fixed location of the scene. To cover as many detection environments, the videos varies in viewpoint, lighting condition and resolution. Then, we choose pixels on obvious fires by saving their coordinate values. After that, each selected pixel will be sampled along time domain, resulting in a sequence with a length equal to the total frame number of the video. Each element value of a sampled sequence is identical to the greyscale amplitude of the correlating pixel at the matching frame. The process of manually sampling one of the 54 videos is shown in Figure 4. Through careful sampling, it results a dataset with an amount of 1296 sequences. The samples are labelled into two categories: fire or non-fire. This is different from the I-Vector applied in speaking verification where each person needs to be annotated. Because in the case of fire detection, what the algorithm faces is a binary classification problem. The total positive samples (representing fire) takes up 3/4 of the dataset, and the rest are regarded as negatives representing the environment or confusable noise.

B. Training and Detection

We split the dataset into training set and test set to train our proposed algorithm, and the two sets hold a size ratio of 3:1. During different iterations of training, we reallocate the samples that go to training or test sets randomly in order to exploit more features from different scenes. The algorithm trains I-Vector feature extractor first. At this step, the key parameters in Eq. 3 will be iterated. In this process, the algorithm learns the differences between fire and the background environment, as well as the discriminations between fires sampled in various environments. And it is a well-tuned Eq. 3 that ensures our model's feature extraction precision. By using Eq. 3, the extractor knows how to best extract features regarding fires as the target. After the I-Vector feature extractor is finely adjusted, we train the G-PLDA classifier to endow the algorithm the ability to make precise decisions. By using a DNN, the threshold at the back end of our classifier could be fine-tuned based on I-Vector features and given labels, contributing a 91% training accuracy as Fig. 5 shows.

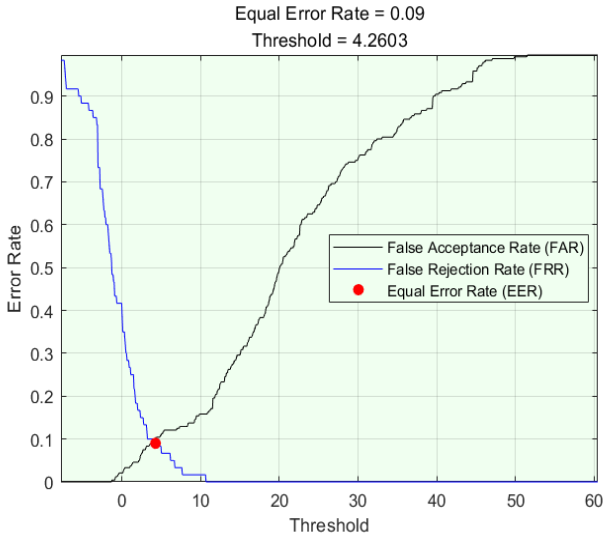

Fig. 5. G-PLDA classifier threshold tuning: After iterating the threshold using DNN, EER reaches 0.09. At this point, recall rates for acceptance or rejection are identical, the G-PLDA classifier reaches best overall performance.

In detection, the unknown video is sampled into sequences in the same approach as the training process does. But the pixels are sampled at a regular adjustable interval as Fig. 6 indicates, resulting in a multi-dimensional matrix. The algorithm will then process all the input sequences in parallel using matrix

manipulation. Each input sample will have an I-Vector and an output classification. By applying the output classes back onto the unknown video, the position and approximate shape of fires would be indicated. It is understandable that the resolution and accuracy would increase as the sampling interval narrows.

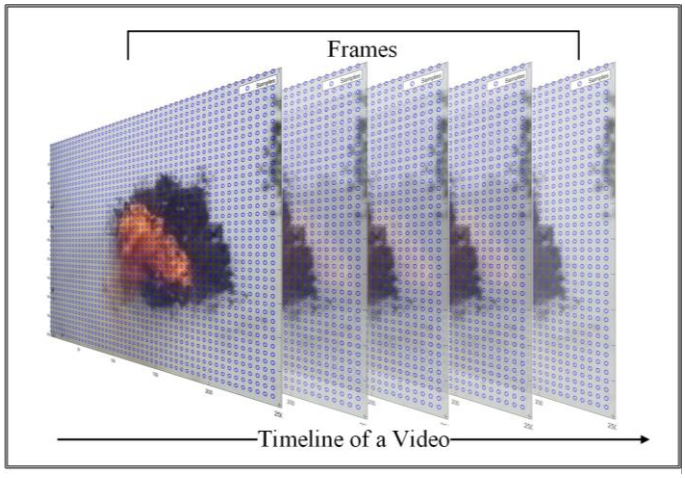

Fig. 6. Sampling of unknown videos before detection: Pixels are sampled at a regular adjustable interval (sampled pixels are spotted by blue circles). Sampled sequences are stored in a multi-dimensional matrix.

V. Results

A. Speed and Accuracy

Usually, smaller parameters are prone to reduce the accuracy of recognition. However, the proposed algorithm not only speeds up the FPS, but also increases the accuracy of recognition.

We use a 480 × 460 size video consisting of 3491 frames to test speed performance on four different algorithms. We operated the following test data under the hardware test conditions of an Intel 10875H CPU and an NVIDIA RTX 2070 MaxQ GPU. The operation results are shown in Table II. As can be seen from the table, the shortest time consumption is 17s which is performed by YOLOv5-S on the GPU. However, our algorithm can complete the same task using only 23s under GPU mode, while obtaining over 1.4 times higher the accuracy of YOLOv5-S, despite a slight cost of speed. Compared with other methods, the proposed algorithm greatly reduces the time consumption of calculation, showing great process capacity.

For the comparison of accuracy, the first three networks are trained based on pretraining models with the same dataset. This dataset includes the Kaggle data set of 1000 images and 500 images about fire we collected from the Internet (to be described later). Our proposed algorithm uses smaller datasets containing 1296 sequences. After training, those four methods are tested on the same unknown labelled video. Although the accuracy of Inception-V4-OnFire is the highest among the three mainstream networks, reaching 83% our algorithm achieves 91% accuracy in the case of smaller data set training. This means that our algorithm achieves higher dataset utilization and computational efficiency, while remaining relatively high accuracy.

TABLE I. Speed and Accuracy of the Four Models

Detection Methods	Speed	Accuracy (%)
FireNet	61s	51
InceptionV4-OnFire	108s	83
YOLOv5-s	17s	67
Our Method	23s	91

B. Parameters

The number of parameters of the four models is shown in Table I. It can be seen that the parameter size of our method is between FireNet and YOLOv5-S, which is less than that of Inception-V4-OnFire. For our method, the parameter size is 14.3 million, which is about 75% of the number of parameters in the Inception-V4-OnFire network. Although the parameters size of our method is not minimal, it is still applicable for deployments on small mobile platforms.

Although the parameter size of our method is larger than that of FireNet and YOLOv5-S, its accuracy is much higher than those of the two models

TABLE II. Parameters of the Four Models

Detection Methods	Parameters
FireNet	3.1M
InceptionV4-OnFire	Over 20M
YOLOv5-s	7.01M
Our Method	14.3M

C. Datasets

We compared the datasets only detecting fire or no fire (i.e. binary classification) with our dataset. In the first two rows of Table III, the dataset sizes for training the corresponding network for fire detection are shown. Arpit Jadon et.al. [7] used a dataset of 2.4K images to train the FireNet, and Shixiao Wu et.al [8] used 1.1K images for YOLOv3 transfer learning towards fire detection. Our dataset is much smaller than the common ones, but our method can still perform equally well when feeding unknown data, compared with the former two studies.

TABLE III. Datasets Comparison

Datasets	Contents
FireNet Dataset [7]	2.4K(image)
YOLOv3 Dataset [8]	1.1K(image)
Our Dataset	1K(greyscale sequence)

VI. Discussion

In this paper, we sample the video into greyscale sequences, and to extract the dynamic features along the time domain. The algorithm makes full use of the brightness patterns of fire in greyscale and achieves decent accuracy. However, in some cases, the occurrence of fire cannot be well determined only from the fluctuation of the object. Therefore, we will add color channels to assist analysis in our future research. Through the time series acquisition of RGB channels, the changes of the target on the three color elements can be analyzed respectively.

Based on specific colors, amplitude difference between color channels could be further researched to discover the relative patterns. Our method proposed in this research can also be used an auxiliary algorithm of the existing ones, providing additional judgements about the existence of fire.

VII. Conclusion

In this paper, we propose an algorithm based on I-Vector to detect the occurrence of fire in videos by using time series. Having set off from fire dynamic features, our algorithm achieves a unique approach using a lighter model while only requiring smaller datasets. The main advantage of our algorithm is that it recognizes fires within video frames at the first glance, which counters the low recognition efficiency of the current CNN networks which process frame by frame. The application of this model has great potentials varying from video fire monitoring and UAV fire inspection. Compared with mainstream convolutional neural networks, our method is easier for both training and deployment. Containing fewer parameters also means lower performance requirements for computing platforms.

Acknowledgement

We here sincerely thank Tiangong University and our group members of Tianjin Provincial University Student Innovation and Entrepreneurship Program (No.202110058107). We would also like to express our appreciation to Mr. Yukuan Sun and Mr. Zijing Zhang for their inspirations and professional suggestions.

References

[1] Valizadeh, M., & Wolff, S. J. (2022). Convolutional Neural Network applications in additive manufacturing: A review. *Advances in Industrial and Manufacturing Engineering*, 100072.

[2] Maheswari, B. U., Rajakumar, M. P., & Ramya, J. (2022). Dynamic differential annealing-based anti-spoofing model for fingerprint detection using CNN. *Neural Computing and Applications*, 1-17.

[3] Zhu Yan, Zhang Bin, Zhang Yaping, Li Weimin, & Cai Likang. (2018). Research on multi-point distributed fire monitoring method based on visual recognition. *Measurement and Control Technology, 37(10), 5*

[4] Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S. W. (2018). Convolutional neural networks based fire detection in surveillance videos. *IEEE Access*, *6*, 18174-18183.

[5] Glembek, O. , Burget, L. , P Matějka, M Karafiát, & Kenny, P. . (2011). Simplification and optimization of i-vector extraction. *2011 IEEE International* Conference *on Acoustics, Speech and Signal Processing (ICASSP). IEEE*

[6] Kim, Y. J. , & Kim, E. G. . (2017). Fire Detection System using Faster R-CNN. *International Conference on Future Information & Communication Engineering.*

[7] Arpit Jadon, A. , Omama, M. , Varshney, Ansari, M. S. , & Sharma, R. . (2019). Firenet: a specialized lightweight fire & smoke detection model for real-time iot applications. *arXiv:1905.11922v2 [cs.CV] 4 Sep 2019*

[8] Wu, S. , & Zhang, L. . (2018). Using Popular Object Detection Methods for Real Time Forest Fire Detection. *2018 11th International Symposium on Computational Intelligence and Design (ISCID). IEEE.*

[9] ZHAO Baojun, ZHAO Boya, TANG Linbo, WANG Wenzheng, WU Chen.(2019) Multi-scale object detection by top-down and bottom-up feature pyramid network. *Journal of Systems Engineering and Electronics,2019,30(01):1-12.*

[10] Pi, L., & Wu, J. (2021, May). FPNet: Fusion Attention Instance Segmentation Network Based On Pose Estimation. *In 2021 33rd Chinese Control and Decision Conference (CCDC) (pp. 2426-2431). IEEE.*

[11] Mclaughlin, J. , Reynolds, D. A. , & Gleason, T. P. . (1999). A study of computation speed-UPS of the GMM-UBM speaker recognition system. *European Conference on Speech Communication & Technology. DBLP.*

[12] Reynolds, D. A. (2009). Gaussian mixture models. *Encyclopedia of biometrics, 741(659-663).*

[13] Bond, S. R. , Hoeffler, A. , & Temple, J. . (2001). Gmm estimation of empirical growth models. *Cepr Discussion Papers, 159(1), 99-115.*

[14] Kenny, P. , Boulianne, G. , Ouellet, P. , & Dumouchel, P. . (2007). Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing.*

[15] Kenny, P. , Stafylakis, T. , Ouellet, P. , & Alam, M. J. . (2014). JFA-based front ends for speaker recognition. *IEEE International Conference on Acoustics. IEEE.*

[16] Garcia-Romero, D. , & Espy-Wilson, C. Y. . (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems. *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011.*

[17] Matejka, P. , Glembek, O. , Castaldo, F. , Alam, M. J. , Kenny, P. , & Burget, L. , et al. (2011). Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. DBLP.

[18] Ying Liu, Luyao Geng, Weidong Zhang, Yanchao Gong, and Zhijie Xu (2021), Survey of video based small target detection," Journal of Image and Graphics, Vol. 9, No. 4, pp. 122-134. doi: 10.18178/joig.9.4.122-134.