International Journal of Progressive Research in Science and Engineering
Volume-1, Issue-5, August-2020
www.ijprse.com

Fire Detection Using Deep Learning

Suhas G[1], Chetan Kumar[1], Abhishek B S[1], Digvijay Gowda K A[1], Prajwal R[1]

[1]Student, Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore, Karnataka, India.
Corresponding Author: chetanbhat34@gmail.com

Abstract: From sprawling urbans to dense jungles, fire accidents pose a major threat to the world. These could be prevented by deploying fire detection systems, but the prohibitive cost, false alarms, need for dedicated infrastructure, and the overall lack of robustness of the present hardware and software-based detection systems have served as roadblocks in this direction. In this work, we endeavor to make a stride towards detection of fire in videos using Deep learning. Deep learning is an emerging concept based on artificial neural networks and has achieved exceptional results in various fields including computer vision. We plan to overcome the shortcomings of the present systems and provide an accurate and precise system to detect fires as early as possible and capable of working in various environments thereby saving innumerable lives and resources.

Key Words: - ***Fire accidents, Fire detection, Surveillance video, Machine learning, Deep Learning, Transfer Learning.***

I. Introduction

Fire accidents pose a serious threat to industries, crowded events, social gatherings, and densely populated areas that are observed across India. These kinds of incidents may cause damage to property, environment, and pose a threat to human and animal life. According to the recent National Risk Survey Report [1], Fire stood at the third position overtaking corruption, terrorism, and insurgency thus posing a significant risk to our country's economy and citizens. The recent forest-fires in Australia reminded the world, the destructive capability of fire and the impending ecological disaster, by claiming millions of lives resulting in billions of dollars in damage.

Early detection of fire-accidents can save innumerable lives along with saving properties from permanent infrastructure damage and the consequent financial losses. In order to achieve high accuracy and robustness in dense urban areas, detection through local surveillance is necessary and also effective. Traditional opto-electronic fire detection systems have major disadvantages: Requirement of separate and often redundant systems, fault-prone hardware systems, regular maintenance, false alarms and so on. Usage of sensors in hot, dusty industrial conditions is also not possible. Thus, detecting fires through surveillance video stream is one of the most feasible, cost-effective solution suitable for replacement of existing systems without the need for large infrastructure installation or investment. The existing video-based machine learning models rely heavily on domain knowledge and feature engineering to achieve detection therefore, have to be updated to meet new threats.

We aim to develop a classification model using Deep learning and Transfer Learning to recognise fires in images/video frames, thus ensuring early detection and save manual work. This model can be used to detect fires in surveillance videos. Unlike existing systems, this neither requires special infrastructure for setup like hardware-based solutions, nor does it need domain knowledge and prohibitive computation for development.

II. Related Work

Among the different computer-based approaches to detect fire, the prominent approaches we found were using Artificial Neural network, Deep Learning, Transfer learning and convolutional neural network. Artificial Neural Network based approaches seen in paper [2] uses Levenberg-Maraquardt training algorithm for a fast solution. The accuracy of the algorithm altered between 61% to 92%. False positives ranged from 8% to 51%. This approach yielded high accuracy and low false positive rate, yet it requires immense domain knowledge.

In this paper [3], The author says that the present hardware-based detection systems offer low accuracy along with high occurrence of false alarms consequently making it more likely to misclassify actual fires. It is also not suitable for detecting fires breaking out in large areas such as forests, warehouses, fields, buildings or oil reservoirs. The authors used a simplified YOLO (You Only Look Once) model with 12 layers. Image augmentation techniques such as rotation, adjusting contrast, zooming in/out, saturation and aspect ratio were used to create multiple samples of each image, forming 1720 samples in total. It aims to draw a bounding box around the flame region. It outperformed existing models when the color features of the flames varied from those in training set.

Paper [4] provides two approaches First approach is to perform training on the data set using Transfer Learning and later fine tune it. The next approach was to extract flame features, fuse them and classify it using a machine learning classifier. The transfer learning algorithms used were Xception, Inception V3, ResNet-50, trained in ImageNet. In the first approach, accuracy up to 96% was achieved. The second approach, stacking Xgboost and lightbgm achieved an AUC of 0.996. Transfer learning models greatly reduces the training time required for our model. It requires comparatively smaller data set. Both approaches don't require any sort of domain knowledge. In works [7] and [8], Deep CNN approach was taken to detection and localization of fires. The accuracy obtained was between 90 to 97% in both of these papers. This approach is time consuming and training was performed using Nvidia GTX Titan X with 12 GB of onboard memory.

Traditional Machine learning using feature extraction yielded high accuracy and low false positive rate, yet it requires immense domain knowledge i.e., about color-model, color-space, patterns and motion vectors of flames. when the object changes, the models need to be rebuilt for the new objects. The traditional approach to feature engineering [12] is manual in nature. It involves handcrafting features incrementally using domain knowledge, a tedious, time-consuming, and error-prone process. The resultant model is problem dependent and might not perform well on new data. Automated feature engineering ([3][5]) improves upon this inefficient workflow by automatically extracting useful and meaningful features from data with a framework that can be applied to any problem. It not only cuts down on the time spent, but creates features that can be interpreted and prevents data leakage. With transfer learning, instead of developing a model from scratch, we can start from a pre-trained model with necessary fine-tunings. These models can be imported directly from Keras. The use of pre-trained models saves a lot of computational work, which otherwise, would require high end GPU's. Inception V3, Inception-ResNet-V2 were found to be ideal algorithms for feature extraction as they showed promising results with high accuracy ([3]). With transfer learning, instead of developing a model from scratch, we can start from a pre-trained model with necessary fine-tunings. These models can be imported directly from Keras. The use of pre-trained models saves a lot of computational work, which otherwise, would require high end GPU's. Inception V3, Inception-ResNet-V2 were found to be ideal algorithms for feature extraction as they showed promising results with high accuracy.

III. System Architecture

The passive components of the system include data preprocessing, feature engineering, model selection scripts which were used to train and develop machine learning model.

Source/input data which is in the form of videos is split into frames and preprocessed to convert it into a format that is suitable to be fed as input to pre-built models for feature extraction. The deep learning model returns a feature vector which is also known in transfer learning terminology as bottleneck features.

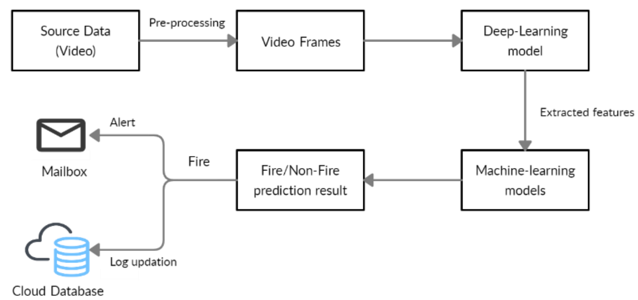

Fig.1: System Architecture

In the next stage, the bottleneck features are passed through a classification model to obtain the result, which may be either fire of Non-fire. The classification model was built through training using the training data set.

The result of the classification is displayed to the user, and depending on the result, further actions are taken. If the result is a fire then, an email is sent to the concerned stakeholders along with the video frame and date-time stamp to alert them. The email to which the mail is delivered can be changed by the user. An entry will also be made in the cloud database for the purpose of analysis.

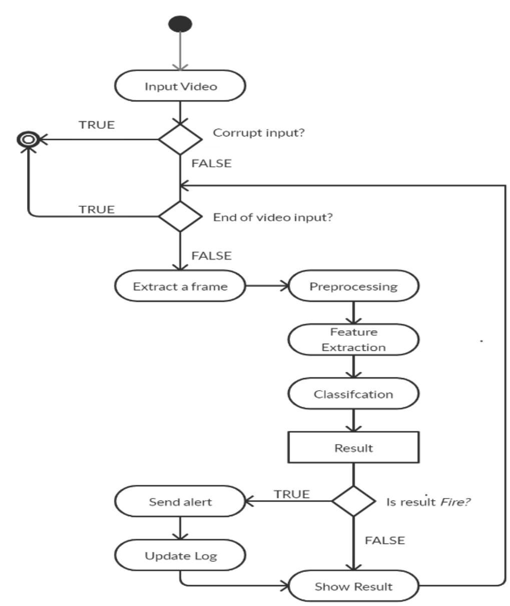

Fig.2: Activity Diagram

IV. Methodology

The model is divided into two parts

1. Data Collection and Pre-processing.
2. Building fire detection model by Transfer Learning.

The first step is to gather video frames for the problem statement. The dataset has 2 classes - fire and non-fire. Positive samples consist of images with real fire. False Positives consists of images which have objects that look like fire but are not. False positives are easier to collect. Thus, we need to collect diverse video frames which will help better fire detection. The collected dataset is divided into train and test video frames. The dataset currently has 1678 fire images/video frames and 1368 that of non-fire sourced from google since there is no standard data set available.

The second step is to use various available pre-trained models in Keras to extract the video frame features. The pre-trained models are trained on very large-scale video frames classification problems. The convolutional layer's act as feature extractor and the fully connected layers' act as Classifiers. Since these models are very large and have seen a huge number of images, they tend to learn very good, discriminative features. In order to do extract, the video frames feature we remove the last layer i.e. fully connected layer. This provides us with a feature vector. The feature vector sizes differ from model to model. The central concept of Transfer Learning is to use a more complex but successful pre-trained DNN model to transfer its learning to our more simplified problem. Instead of creating and training deep neural nets from scratch (which takes significant time and computing resources), we use the pre-trained weights of these deep neural net architectures (trained on ImageNet) and use it for our own dataset. We have used ResNet-50, InceptionV3 and InceptionResNetV2. models to extract the features and various ML algorithms [SVM, Logistic Regression, Naive Bayes and Decision Tree] on the extracted features to detect fire in video frames.

All the possible feature-extractor and classifier combinations were evaluated using stratified K-fold validation. Table 1 shows the obtained performance metric values for different combinations of deep learning networks and classifier algorithms. The best performance was observed with ResNet-50 as the deep learning feature extraction model and Support Vector Machine as the ML classifier and the Accuracy, Precision and Recall values for this combination was 97.8%, 97.46% and 97.66% respectively. Hence this would be used in our application.

Table.1: Performance evaluation results of different model combinations

Network	Algorithm	Accuracy	Precision	Recall
Resnet 50	Decision Tree	94.78%	93.98%	94.44%
	Naïve Bayes	93.17%	91.92%	92.98%
	Logistic Regression	97.73%	97.30%	97.66%
	SVM	97.80%	97.46%	97.66%
Inception-Net V2	Decision Tree	90.51%	89.59%	89.25%
	Naïve Bayes	94.32%	91.75%	95.97%
	Logistic Regression	96.81%	96.23%	96.71%
	SVM	96.25%	94.71%	97.07%
Inception V3	Decision Tree	92.58%	92.07%	91.37%
	Naïve Bayes	94.71%	92.63%	96.19%
	Logistic Regression	97.17%	96.53%	97.22%
	SVM	96.51%	95.36%	97.00%

V. Results And Discussion

The aim of our work was to develop to an application capable of detecting fire in videos and images, which is robust and works in any environment. In this regard, we have experimented with various deep learning models and classification models and have selected ResNet-50-SVM combination for implementation as it offered the best performance metric values (Accuracy, Precision and Recall values for this combination was 97.8%, 97.46% and 97.66% respectively). An email alert feature has also been incorporated to our application to provide real time alerts to the concerned stakeholders along with a logging system, which is implemented using Firebase. The GUI offers a user-friendly experience and allows user with non-technical background to make use of the application. The application performed exceptionally well during testing. It was able to identify fires in all of the twelve test fire videos but misclassified some instances of non-fire videos.

Compared to existing hardware solutions, our application is affordable, robust, reliable and provides high performance without the need for setting up a dedicated infrastructure. Due to the use of deep learning and transfer learning techniques, our model is easier to build, alter, upgrade, requires fewer computing resources and offers better performance than existing software solutions that make extensive use of feature engineering, domain knowledge.

Table.2: Non-fire videos testing results.

Test Case Description	**Expected Output**	**Actual Output**	**Test Status (P/F)**
Video with visuals of mountains, pools, indoors, driving	Non-fire	Non-fire	P
Headlight Glare during night	Non-fire	Fire	F
Kid playing with toys	Non-fire	Non-fire	P
Man riding bicycle	Non-fire	Non-fire	P
Close up view of light through of fabric	Non-fire	Non-fire	P
Daytime Traffic	Non-fire	Non-fire	P
Camera focused on sun in a garden setting	Non-fire	Non-fire	P
Man walking with an orange bag in an office environment	Non-fire	Non-fire	P
Lake with yachts and windmill	Non-fire	Non-fire	P
Yellow heavy-duty vehicles moving on a flyover	Non-fire	Non-fire	P
People waiting at train terminal with a train approaching	Non-fire	Non-fire	P
People moving inside an airport, subway terminal	Non-fire	Non-fire	P

Table.3: Fire videos testing results

Test Case Description	**Expected Output**	**Actual Output**	**Test Status P/F**
Bus on Fire	Fire	Fire	P
Kitchen sink fire	Fire	Fire	P
Twigs and leaves on fire	Fire	Fire	P
Fireplace	Fire	Fire	P
Automobile on Fire	Fire	Fire	P
Piece of cloth set on fire by a firefighter	Fire	Fire	P
Trash can set on Fire	Fire	Fire	P
Close-up in view of fire	Fire	Fire	P
Huge Campfire	Fire	Fire	P
Forest Fire	Fire	Fire	P
Bush fire	Fire	Fire	P
Fire on mattress	Fire	Fire	P

VI. CONCLUSION

The present decade is marked by huge strides in areas of processing, computation and algorithms. This has enabled great progress in many fields including processing of surveillance video streams for recognizing abnormal or unusual events and actions. Fire accidents have caused death and destruction all over the world, consuming countless lives and causing billions in damages. This implies that developing an accurate, early, affordable fire-detection system is imperative Therefore, we have proposed a fire detection model for videos/video frames using transfer learning for deep learning. The models make use of ResNet-50, InceptionV3 and Inception-ResNet-V2 models to extract the features and various ML algorithms such as SVM, Logistic Regression, Naive Bayes and Decision Tree on the extracted features to detect fire in video frames. Looking at the results, ResNet-50 with SVM works best for our problem statement. Coming to the application on the whole, it works in real-time and has the ability to send alert emails along with offering a user-friendly graphical interface. It's cost-effective, reliable, robust, accurate compared to existing opto-electronic hardware and software-based systems in the market.

VII. FUTURE SCOPE

The application can be enhanced by training the model with a larger dataset consisting of fires at various stages and dimensions. With higher GPU memory, we could use two deep learning models for feature extraction, whose output feature vectors are concatenated and classified to offer more robustness. An R-CNN model can be used to implement fire localization along with classification. We can also expect better deep learning architectures to emerge in the future, offering better feature extraction. The application will also offer a considerably better performance when run on machines having better processing power compared to existing one of which it has been developed.

References

[1]. National Risk Survey Report - Pinkerton, FICCI (2018).

[2]. Janku P., Kominkova Oplatkova Z., Dulik T., Snopek P. and Liba J. 2018. “Fire Detection in Video Stream by Using Simple Artificial Neural”. Network. MENDEL. 24, 2 (Dec. 2018), 55–60.

[3]. Shen, D., Chen, X., Nguyen, M., & Yan, W. Q. (2018). “Flame detection using deep learning”. 2018 4th International Conference on Control, Automation and Robotics (ICCAR).

[4]. Li, C., & Bai, Y. (2018). “Fire Flame Image Detection Based on Transfer Learning”. 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS).

[5]. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho and S. W. Baik, “Convolutional Neural Networks Based Fire Detection in Surveillance Videos," in IEEE Access, vol. 6, pp. 18174-18183, 2018.

[6]. S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[7]. [Zhang, Qingjie & Xu, Jiaolong & Xu, Liang & Guo, Haifeng. (2016). “Deep Convolutional Neural Networks for Forest Fire Detection”.

[8]. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang and S. W. Baik, “Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications,” in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 7, pp. 1419-1434, July 2019.

[9]. Z. Jiao et al., “A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3,” 2019 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 2019.

[10]. Faming Gong ,1 Chuantao Li,1 Wenjuan Gong ,1 Xin Li,1 Xiangbing Yuan,2 Yuhui Ma and Tao Song. “A Real-Time Fire Detection Method from Video with Multi-feature Fusion”. Hindawi, Computational Intelligence and Neuroscience, Volume 2019.

[11]. L. Shao, F. Zhu and X. Li, “Transfer Learning for Visual Categorization: A Survey,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, pp. 1019-1034, May 2015.

[12]. S. Mohd Razmi, N. Saad and V. S. Asirvadam, “Vision-based flame detection: Motion detection & fire analysis,” 2010 IEEE Student Conference on Research and Development (SCOReD), Putrajaya, 2010, pp. 187-191.

[13]. T. Qiu, Y. Yan and G. Lu, “A new edge detection algorithm for flame image processing,” 2011 IEEE International Instrumentation and Measurement Technology Conference, Binjiang, 2011, pp. 1-4.

[14]. M. Ligang, C. Yanjun and W. Aizhong, “Flame region detection using color and motion features in video sequences,” The 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, 2014, pp. 3005-3009.

[15]. Toulouse, Tom & Rossi, Lucile & Celik, Turgay & Akhloufi, Moulay. (2015). “Automatic fire pixel detection using image processing: A comparative analysis of Rule-based and Machine Learning-based methods. Signal, Image and Video Processing”.

[16]. K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[17]. S. Wilson, S. P. Varghese, G. A. Nikhil, I. Manolekshmi and P. G. Raji, “A Comprehensive Study on Fire Detection,” 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, 2018, pp. 242-246.

[18]. X. Wu, X. Lu and H. Leung, “An adaptive threshold deep learning method for fire and smoke detection,” 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, 2017, pp. 1954-1959.

[19]. Chenebert, Audrey & Breckon, Toby & Gaszczak, Anna. (2011). “A nontemporal texture driven approach to real-time fire detection”.

[20]. Seebamrungsat, Jareerat et al. “Fire detection in the buildings using image processing.” 2014 Third ICT International Student Project Conference (ICTISPC) (2014): 95-98.