Group 2: Real-Time Fire & Smoke Detection

Pranjal Kumar Srivastava Mayank Devnani Vivek Kumar Gautam
Atul Kumar
Lavkush Mani Tripathi
22111046, 22111042, 22111069, 22111013, 22111034
{pranjalks22, mayankdev22, vivekg22, atulkumar22, lavkushmt22}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

Fire detection has been a hot topic in computer vision research. Historically, conventional algorithms based on rule-based approach were used to classify whether a frame contains fire or not. Similar techniques were followed for smoke detection. These features needed to be defined manually and were error prone. With the success of Deep Learning, the approach to fire and smoke detection using computer vision also changed. Deep Learning networks have the ability to perform automatic feature extraction and then use those for detection. The approach proposed here uses a Deep Learning architecture based network for fire/smoke detection multi-label problem and image processing techniques for localization of fire/smoke in the frame. Furthermore, the approach here checks for hazardous fire and sends an email to the user with the fire pixels in the frame enclosed in a rectangular box, in case hazardous fire is detected.

1 Introduction

Fire is a universal hazard. Preventing damages from fire requires fast and immediate detection. There are always sensors like smoke detectors, flame detectors available to detect the presence of fire/smoke. But using these sensors can be a costly affair. Moreover, these sensors require close proximity to fire sources and cannot provide information about fire location. Hence, there is a need for using something with minimum hardware cost and maximum efficiency.

The approach proposed here uses camera based computer vision and image processing techniques for carrying out fire/smoke detection and localization. A lot of workplaces/industries/factories and even some homes already have CCTV cameras installed to monitor the activities. The approach proposed here can use this existing infrastructure with no additional hardware cost.

This approach takes the frame from the video stream and feeds it to the trained Machine Learning classification model after pre-processing it. If the model detects the presence of fire or smoke in the frame then fire region is annotated and rate of increase of fire region is computed using successive frames. This way hazardousness of fire is detected. Detecting hazardous fire is necessary because there are places (like kitchen in a home or furnace in a factory) where controlled fire is kept continuously running. If fire is indeed hazardous, a mail is sent to the owner of the system containing the frame with fire and smoke regions annotated. Annotation is carried out using image processing via opencv library.

In the 'Related Work' section we have discussed about the related work done in this field which we have understood from the literature survey we carried out. Then we propose our idea and then describe the methodology in detail. This is followed by the results, where we have mentioned our results and how is it different from related work. Then we discuss the future work and conclude. The last section is a tabular description of the tasks carried by each individual of our team for this project.

2 Related Work

The literature we've reviewed uses the dataset comprised of thousands of images and are mostly of similar type like outdoor fire environment images. On the other hand, we have used almost 6500 images comprising of both outdoor and indoor fire environment images to make the dataset more balanced.

Earlier, fire detection methods based on image processing techniques such as using different color models and hand-designed features (eg. RGB, HSV) were developed. There was a phase in which Support Vector Machines (SVM) were seen as the best way to tackle a lot of challenges in Machine Learning. Hence, there are papers which have used SVM for fire detection task [1].

Other techniques we surveyed included using frame differences, median filters and bayes classifier to detect flame. Also, there are papers in which color [2], shape and movement was used to detect the presence of fire or smoke.

We actually reduced the "false positive rate" as well "false negative rate" compared to literature surveys[3] we had referred. We've achieved the high accuracy as well just because our dataset is much more balanced.

Some of the work[4] we've reviewed uses multiple models in order to classify fire out of the image but we've used single model for classification purpose to extract fire from image.

The literature[5] we've reviewed used the approach of "multi-class classification" in order to predict fire or not / smoke or not. Here, we used the idea of "multi-label classification", through which we can predict only fire, only smoke, fire smoke both and no fire no smoke like images.

We've also added the functionality of classifying the fire images as hazardous or not. We found very limited papers with this feature.

3 Proposed Idea

This project entails creating a system that reads frames from video feed to check if fire/smoke exists and to take action accordingly. The frames are being fed to a Machine Learning classification model, which outputs a vector indicating whether fire or smoke is present in the frame or not. To train the model, we made a dataset using fire/smoke images collected from various sources on the internet. Care was taken to create a balanced dataset, containing images of all 4 possible types:

1) Only Fire, No Smoke
2) Only Smoke, No Fire
3) Both Smoke and Fire
4) Normal (No Smoke, No Fire)

Transfer learning approach was used to train various pre-trained models on the collected data. Once the model classifies a certain frame as containing fire, it is annotated using opencv based image processing techniques to detect fire region area. This rate of change of this area is then computed to classify whether the fire is spreading and hence is in turn hazardous.
The frame containing these fire/smoke images are then sent to the owner of the system with the fire/smoke area annotated.

4 Methodology

4.1 Dataset Collection

The dataset was collected from open-source internet sources. This dataset was further augmented with images generated by us. Distribution of the complete dataset can be seen below:

Split	Only Fire	Only Smoke	Fire + Smoke	Normal	Total
Train	1098	1050	1155	1050	4353
Validation	225	225	225	225	900
Test	225	225	225	225	900
Total	1548	1500	1605	1500	6153

Table 1: Dataset Split

The images in the dataset were labeled by our group using open-source tool 'label-studio'. We also annotated the area occupied by fire pixels in each image using this tool.

4.2 Training the Classification Model

Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. It has been widely used across different domains in Machine Learning including text classification, digit recognition etc. Since we were making this model to run on a Raspberry Pi, we only used those pre-trained model available in Keras which occupied size of less than 100 MB. We made this decision thinking about the storage constraint and computational time of the model, considering that we wanted to run this system on a Raspberry Pi.

We added an output layer having 2 nodes to each of these pre-trained models. Both the nodes were put with 'sigmoid' activation function, as it is a multi-label problem. We trained these models using early-stopping based cross-validation approach. We used different optimizers and varied the value of patience parameter to get the best performance of each model. The best performance of each model is given below:

Pre-Trained Model	Optimizer	Training Accuracy(%)	Validation Accuracy(%)	Test Accuracy(%)	Model Size
ResNet-50	SGD	99.02	96.22	96.00	94.2 MB
MobileNetV1	SGD	98.41	94.33	94.11	14.3 MB
MobileNetV2	SGD	98.44	94.44	94.55	12.4 MB
EfficientNetB0	Adam	98.78	95.77	94.88	20.9 MB
InceptionV3	Adam	95.15	92.11	90.55	89.8 MB

Table 2: Trained Models

As can be seen from the table, the best accuracy was obtained with ResNet-50 model. Hence, this model was chosen as the classification model for our system. Any image/frame which we feed to this model, we first preprocess it to a size of 224 by 224 pixels, as ResNet-50 expects input image of this size.

4.3 Localization

Localization was carried out for both fire and smoke. Initially, our approach was to train a Machine Learning model to output the fire region parameters in a given image. We even annotated the complete dataset with the coordinates of the fire pixels using the open-source tool 'label-studio'. But considering the accuracy obtained and the time delay taken to carry out this task, we abandoned this approach for a much simpler, faster and more accurate image processing approach.

We converted the image to grayscale and then applied thresholding on the image with proper parameters. Then perform contour detection on the thresholded image to find the edges of fire. It was observed that only the largest contour was the area of interest so it's the only contour that is extracted. The extracted contour gives the part of frame that contains fire, which can be used to make a bounding box and sent back to the live footage. Similar approach was followed to localize smoke in an image, only thing was that different thresholding was applied here.

4.4 Hazardous/Non-Hazardous and Automatic Mail

The area of fire pixels computed in the localization step is used for checking hazardous fire. We compute moving average of fire area of previous 3 frames (if previous 3 frames contained fire) and check if the fire area in the 4th frame exceeds this moving average by 13%. This number of 13% we got from a paper.

If it did exceed, we classify the fire as hazardous. We run a SMTP based automatic mail logic at the background. As soon as hazardous fire is detected, the fire region in the frame containing fire is enclosed in a rectangular box and then sent to the owner of the system.

4.5 Complete Workflow

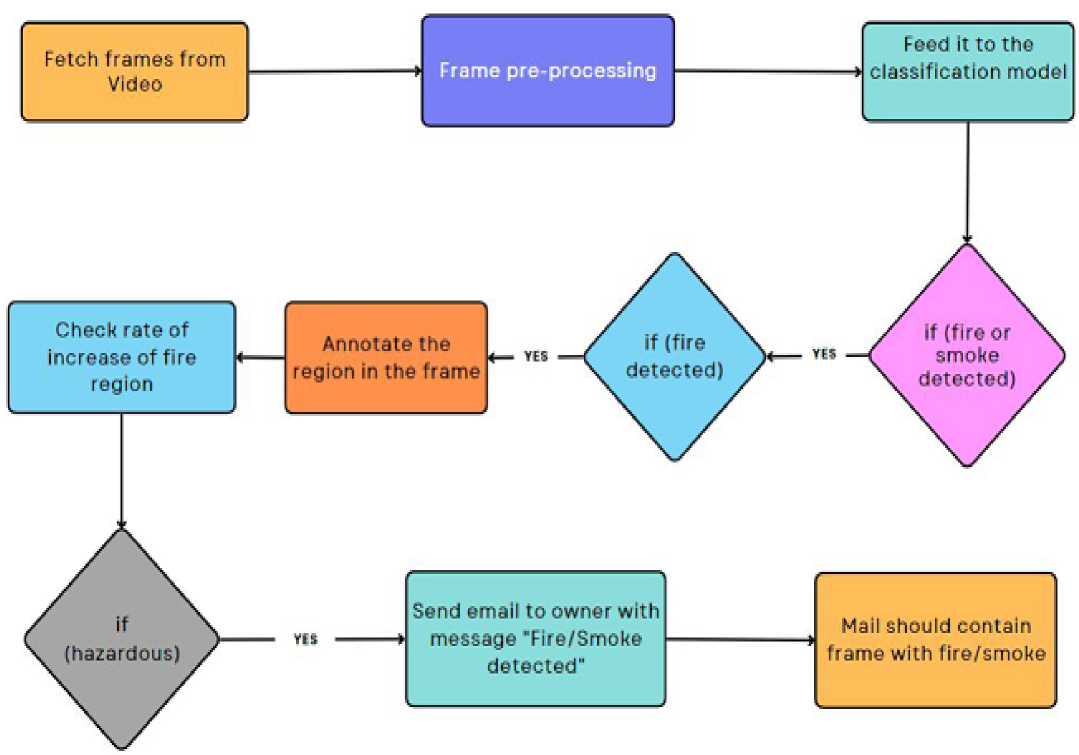

Figure 1: Workflow

5 Results

1) Our system is able to offer real-time prediction at a speed of 15 frames per second.

2) The papers we read as part of literature survey were doing the fire/smoke classification task as a multi-class problem. We have tried doing the same as a multi-label problem, and thus unlike most other systems, our model is able to detect the presence of both fire and smoke together, which is not able if the problem was modeled as a multi-class problem.

3) The confusion matrices and values of precision, recall f1-score are given in the below tables.

	Actual Positive	**Actual Negative**
Predicted Positive	441	9
Predicted Negative	9	441

Table 3: Fire Confusion Matrix

	Actual Positive	**Actual Negative**
Predicted Positive	442	8
Predicted Negative	14	436

Table 4: Smoke Confusion Matrix

	Precision	**Recall**	**F1-Score**
Fire	0.98	0.98	0.98
Smoke	0.98	0.97	0.98

Table 5: Precision, Recall & F1-Score

4) Compared to [3], our model gives lesser False Negatives and a good False Positive. This is achieved using a much lighter model comparatively as well.

	False Positive Rate	**False Negative Rate**
Fire	1 %	1 %
Smoke	0.88 %	1.55 %

Table 6: False Positive/Negative Rate

6 Discussion and Future Work

While the results obtained in this project are encouraging, we feel they can be further improved.

1) Any model is as good as the data. With a more diverse training data, model performance can be further improved to generalise to real-world tasks.

2) Using more features to perform better localization, especially for smoke.

3) Having a dataset available with fire/smoke regions annotated, a Machine Learning based strategy can be explored to carry out the localization task also using a Machine Learning model. This approach, can in theory give better results than any image processing based techniques.

4) We can interface our system with fire-alarms.

5) This system can be scaled to be used from simple home applications to industrial settings and even in outdoor settings like forest fires.

6) The email part sometimes takes quite a while to finish because of the authentication bottleneck

that is not in our control. This is also a scope where our system can be improved by exploring more notification alternatives.

7 Conclusion

This project tackled the problem of providing a cheap and efficient method to detect the presence of fire or smoke. It provided an opportunity to explore small sized less complex models that would work on any IoT board like a Raspberry Pi.

Furthermore, it has to be ensured that the overall program runs fast enough to be called real-time. This means that different components including detection, localization and upto sending of email have to work smoothly with as much low latency as possible.

Overall, this system offers a lightweight and efficient implementation for which it was intended.

8 Individual Contributions

Name	Contributions
Lavkush Mani Tripathi	1) Data Annotation 2) Email Automation
Atul Kumar	1) Data Annotation 2) Fire localization
Pranjal Kumar Srivastava	1) Data Annotation 2) Classification model development
Mayank Devnani	1) Data Annotation 2) Classification model development
Vivek Kumar Gautam	1) Detect hazardous fire 2) Code Integration

Table 7: List of task contribution

References

[1] Z. Wang, Z. Wang, H. Zhang, and X. Guo, "A novel fire detection approach based on cnn-svm using tensorflow," in *International conference on intelligent computing*, pp. 682–693, Springer, 2017.

[2] T. Zaman, M. Hasan, S. Ahmed, and S. Ashfaq, "Fire detection using computer vision," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 356–359, IEEE, 2018.

[3] H. Wu, D. Wu, and J. Zhao, "An intelligent fire detection approach through cameras based on computer vision methods," *Process Safety and Environmental Protection*, vol. 127, pp. 245–256, 2019.

[4] P. Foggia, A. Saggese, and M. Vento, "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE TRANSACTIONS on circuits and systems for video technology*, vol. 25, no. 9, pp. 1545–1556, 2015.

[5] N. Chowdhury, D. R. Mushfiq, and A. E. Chowdhury, "Computer vision and smoke sensor based fire detection system," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–5, IEEE, 2019.