IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue-14, May 2019 E-ISSN: 2347-2693

Forest Fire Detection Using Convolutional Neural Networks

Shruthi G[1*], Disha Bhat[2], Gagana H[3], Dimple M K[4], Kavitha[5]

[1,2,3,4,5]School of Computing & Information Technology, REVA University, Bangalore, India

Corresponding Author: shruthig@reva.edu.in

DOI: https://doi.org/10.26438/ijcse/v7si14.323325 | Available online at: www.ijcseonline.org

Abstract— there have been many technologies developed recently on embedded processing that have enabled the vision based systems to detect fire using convolutional neural networks (CNN). All such methods need large memory and more computational time. In this research paper we initiate more efficient fire detection strategy with high performance. Here we are considering computational complexity and exact model for the problem by comparing other computational expensive networks. By considering the nature of problem statement, we can increase the efficiency and accuracy of the model. The results on benchmark datasets of fire shows us the efficient work of the proposed system with validation for detection of fire under cctv maintenance compared to other art of methods.

Keywords— CNN, Fire detection, stride, filters, pooling, surveillance videos.

I. Introduction

The capabilities of smarter surveillance systems in smart devices has been improved with the improvement of the processing power of the embedded systems which help in many of the accidents happening in day to day life. Fire detection in early stages has been difficult till now due to the use of old technologies and traditional alarming methods. To overcome such situations, the use of cameras and sensors will help a lot. The fire detection done using visible light cameras is the common fire detection method. There are many sub categories in this method. They are pixel-level, blob level and patch level. Pixel level makes use of colors and flickers to detect fire. But the main problem with this is its efficiency which is very low. The blob level method yields better efficiency but training their classifiers is difficult. The third level is the patch level which is better than the above mentioned two categories yet has a low efficiency which needs to be improved a lot. To improve accuracy, color and motion features for flame detection are considered and investigated. The irregularity and the dynamic behavior of the images in both HSI and RGB color spaces are considered. So we can understand that the fire detection accuracy directly depends on the efficiency of the system rather it is inversely proportional to the complexity of the working of the system. Considering the recent forest fires such as that of Bandipur in 2019 in which 10920 acres of forest had caught fire and a large destruction of flora and fauna had caused. The forest caught fire due to a small spark caused by few shepherds to chase away the tigers away from the cattle. All this can be stopped if we keep cameras around the forests so that no such incidents happen in the future. With this need to develop more efficient and accurate flame detection technologies with less computational complexity and less lost, we have been studying about Neural networks for flame detection in early stages using CCTV surveillance videos.

II. Related Work

The research work done over the past decade has been focused on traditional methods for fire detection. The major drawbacks of these methods are that they are very time consuming and have poor performance in terms of accuracy of flame detection. Such methods have also sent many false alarms due to the inaccuracy caused by different lightings and fire colored objects. To overcome these issues, we have thoroughly studied deep learning architectures for faster fire detection .We examined various CNNs to improve the fire detection accuracy and reduce the rate of false alarms. Here is an overview of our framework for fire detection.

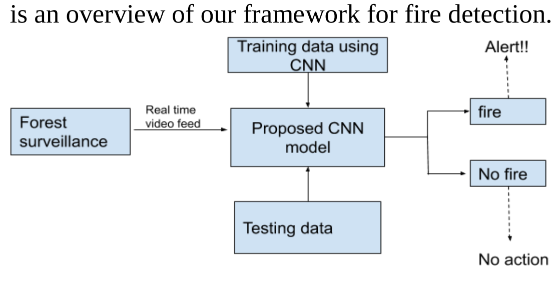


Figure 1. Framework for fire detection using CNN

A.CONVOLUTIONAL NEURAL NETWORKS

CNN is used in many areas such as image recognition, object detection, face recognition etc. CNN image classifications take an image as input, processes it and classifies it according to specific categories. Computers visualize an image as an array of pixels. It also depends on the image resolution(h*w*d where h=height, w=width, d=dimension). The features are extracted from the first layer called the convolution layer. It retains the relationship between the pixels by extracting the image features by small squares of input data. This involves a mathematical operation that takes image matrix (h x w x d) and filters or kernel(fh x fw x d) as inputs. This mathematical operation outputs a volume dimension (h - fh +1) x (w - fw + 1) x 1.The input to the CNN is an image matrix whose dimension depends on the resolution of the image and the depth of the image. For example a color image with a resolution of 100*100*3 is the matrix with a depth of 3 this gives us a total of 30000 data values to work on. The 3 indicates the image colour values RGB.The algorithm followed by the neural network is **y=w*a+b**, where "y" is the labels, "w" is the weights and "b" is the biases. For each stage of the network the value of w and b changes in par with the output of the previous layer. After training all the batches of the training set the CNN generates a model which can be used to classify the live frames from the camera monitoring the forest video feed. Additionally a softmax layer is added to convert the data from the model to a binary representation of the state of fire or not. This data can then be used to alert a response team to help prevent the widespread of such disasters. This whole system can be fully automated as after the training is done, it is only a matter of plug and play of the device.

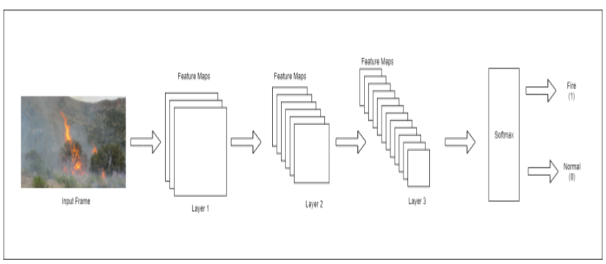


Figure 2. Main operations of a typical CNN architecture

B.FIRE DETECTION USING CNN

The data is first pre-processed. Here the data means the images with two categories. One set of data contains the images with fire and the other set of data contains the images without fire that is the normal state of the forest. The images are present in various formats and sizes. For us to compare all those images, we need to make all those images into a common format. Say we have scaled them down to 100px x 100px.The next step is to convert the data into training, testing and validation sets. The training dataset refers to the actual dataset we use to train the model. The model learns from this data. The validation dataset provides an unbiased evaluation of a model. This data set is used to examine a given model and it is used for frequent examining of the model. The model never learns from this dataset. The test dataset is used to evaluate the final model which fits the training dataset. This dataset provides the final standard for evaluating the model. It is used only after the model is completely trained. It usually contains sampled data that contains various classes of the model. The multi-dimensional image matrix is then converted to a single dimension tensor of size 1x100x100x3, because the input to the neural network can only be a single dimension tensor. The data is then normalized by dividing each vector point to 255, this sets the range of each of the vector points to the range of 0 to 1.Next stage is to build the CNN, the model of the CNN is as shown below.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 98, 98, 32)	896
max_pooling2d_1 (MaxPooling2	(None, 49, 49, 32)	0
dropout_1 (Dropout)	(None, 49, 49, 32)	0
conv2d_2 (Conv2D)	(None, 47, 47, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 24, 24, 64)	0
dropout_2 (Dropout)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 11, 11, 128)	0
dropout_3 (Dropout)	(None, 11, 11, 128)	0
flatten_1 (Flatten)	(None, 15488)	0
dense_1 (Dense)	(None, 64)	991296
dropout_4 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 133)	8645

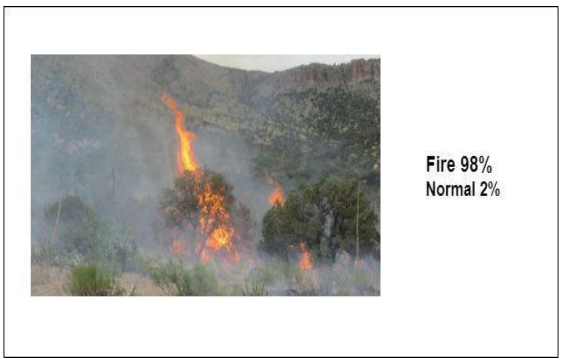
Total params: 1,093,189.0
Trainable params: 1,093,189.0
Non-trainable params: 0.0

Figure 3. The model of CNN

We generally insert a pooling layer in between successive convolutional layers in a CNN architecture. Its aim is to gradually reduce the spatial size of the representation to reduce the amount of parameters and computation in the network which helps in preventing over fitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would in this case be taking a max over 4 numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged. There are two parameters used to modify the behaviour of each layer. These are stride and padding. Stride is the controlling unit that decides how the filter should convolve around the input volume. The amount by which the filter shifts is the stride. Stride is normally set in a way so that the output volume is an integer and not a fraction. By regularly applying filters to the convolutional layers, the spatial size decreases. It causes a loss of the low level features of the convolutional layer. To preserve these low level features, we can apply padding to these convolutional layers and obtain them.

III. RESULTS AND DISCUSSION

We have to stream a video of the forest from which 30 frames are extracted each second and all these frames go through the above process of fire detection using convolutional neural networks. The test data and the training data is used to train the CNN model to learn how to detect if an image contains fire or not. If the fire is detected in the image, an alert is sent to the controlling dept of the forest who will take necessary actions. If no fire is detected, no alert is sent and the video continues streaming.



IV. CONCLUSION AND FUTURE SCOPE

The recent improvements in the technologies have developed many fire detection and alarming systems. The need to detect fire in the forest due to many forest fire incidents recently has motivated us to do this project. The forest fires have caused a great loss to the environment as they imbalance the ecosystem and damage the flora and fauna. Therefore in this research article we have proposed a cost effective fire detection model using surveillance videos. We have tried to reduce the computational complexity and improve the accuracy compared to the previously active models. Although we have tried to improve the accuracy, yet the alerts given are sometimes false. We can further improve this project by training the model to detect smoke and improve the accuracy of the fire detection.

REFERENCES

[1] KHAN MUHAMMAD 1 , (Student Member, IEEE), JAMIL AHMAD 1 , (Student Member, IEEE),IRFAN MEHMOOD 2 , (Member, IEEE), SEUNGMIN RHO 3 , (Member, IEEE), SUNG WOOB BAIK 1 , (Member, IEEE) " *Convolutional Neural Networks based Fire Detection in Surveillance Videos*".

[2] K. Muhammad, J. Ahmad, and S. W. Baik, "*Early Fire Detection using Convolutional Neural Network during Surveillance for Effective Disaster Management*," Neurocomputing, 2017/12/29/ 2017.

[3] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin,"*Computer vision based method for real-time fire and flame detection*," Pattern recognition letters, vol. 27, pp. 49-58, 2006.

[4] L. Rossi, M. Akhloufi, and Y. Tison, "*On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics*," Fire Safety Journal, vol. 46, pp. 9-20, 2011.

[5] P. V. K. Borges and E. Izquierdo, "*A probabilistic approach for vision-based fire detection in videos,*" IEEE transactions on circuits and systems for video technology, vol. 20, pp. 721-731, 2010.

[6] 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, pp. 1545-1556, 2015.

[7] Vinay Chowdary and Mukul Kumar Gupta "*Automatic Forest Fire Detection and Monitoring Techniques: A Survey* " pp. 2018.

[8] Pasquale Foggia ; Alessia Saggese ; Mario Vento "*Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion"* Volume: 25 , Issue: 9 , Sept. 2015

[9] M. V. D. Prasad, G. Jaya Sree, K. Gnanendra, P. V. V. Kishore and D. Anil Kumar *"Fire Detection Using Computer Vision Models in Surveillance Videos"* Vol. 12, No. 19, pp. October 2017