How to Cite:
Bhavani, A., Iswarya, M., & Lokesh, J. (2022). Real-time fire detection based on CNN and inception V3 algorithms. *International Journal of Health Sciences*, *6*(S2), 13513–13527. https://doi.org/10.53730/ijhs.v6nS2.8618

Real-time fire detection based on CNN and inception V3 algorithms

A. Bhavani
Assistant Professor, CSE, GMRIT, Rajam
Corresponding author email: bhavaniashapu@gmail.com

M. Iswarya
UG student, CSE, GMRIT, Rajam

J. Lokesh
UG student, CSE, GMRIT, Rajam

***Abstract*---**Fire is an abnormal event that can cause significant damage to lives and property within a very short time. The main cause of fire disaster includes a human error or system failure which results in severe loss of human life and other damages. Traditional fire alarms are based on sensors that require proximity for activation. They need human involvement to confirm a fire. To overcome these limitations, vision-based real-time fire detection has been enabled in surveillance devices. Once fire appears in any camera, the approach can detect it and send a signal to respective officers of the fire region. This work focused on an intelligent approach using the Deep Learning model for preventing fire hazards from going out of control in high-fire-risk areas. Deep Learning models are effective for fire detection. Convolutional Neural Networks outperform other algorithms in terms of accuracy. In this work Convolution, Neural Network model Inception V3 is used to detect fire indoors and outdoors and protect the surroundings and living beings.

***Keywords*---**deep learning, fire detection system, fire alarms, image processing, vision-based systems.

Introduction

Fire Accidents

Fire is an unusual occurrence that can do severe harm to people and property in a short period [9]. Human mistakes or system failure are the primary causes of such disasters, which result in significant loss of human life and other damages

Manuscript submitted: 09 March 2022, Manuscript revised: 18 April 2022, Accepted for publication: 27 May 2022

[14]. When there is low fire and bearable it is not a big deal to control the fire. But in such a case that the fire is slowly occupied over large are so that controlling fire may become tough and it takes a lot of energy, and expenditure to get it into a normal position [18]. In such major cases, people are not able to control which results in a major loss. To control the fire, traditional fire alarms were introduced earlier [1].

Traditional Fire Alarms

Traditional fire alarms use proximity sensors. They will be activated when they detect fire within a certain distance. To confirm a fire, they need human involvement. But there are limitations to traditional alarms [6]. They can't work properly without human interaction and the major disadvantage is that the accuracy depends on the distance [15]. If the distance between the alarm and the fire is large, then traditional alarms can't detect the fire accurately, also the fire may not be detected at all [7]. So, to overcome this disadvantage of traditional fire alarms, an intelligent fire detection system is needed that can work accurately in both long and short distances [3]. Researchers have proposed an early fire detection system using deep learning models.

Deep Learning

Deep learning is a machine learning and artificial intelligence (AI) technique that mimics how humans acquire knowledge. DL can process large numbers of features which makes it very powerful when dealing with any kind of data [4]. DL is most suitable for image processing and detection of objects [5]. Deep CNN works better than traditional CNN [8]. The approach improves feature extraction and classification performance [11]. Fire detection has been enabled in surveillance equipment and accuracy will be good [20]. When the fire is detected in any camera, the deep learning approach can identify it and send a signal or an alert message, and then the responsible person can be alerted in terms of fire control and damage prevention. Various deep learning models for fire detection were investigated in this study. Inceptionv3 has been experimented with, tested, and observed as a fire detection system in terms of performance and accuracy. Here the first image will be given as input to the InceptionV3 model. This model has convolutional layers, pooling layers, RELU, and SoftMax activation functions [13]. The input goes through all of these layers and finally, the model predicts whether it is a fire image or not. If it is a fire image then it will send an alert message.

Overview of the Proposed Work

In this work, first of all, we have taken a dataset that is a combination of fire and non-fire images, we have developed the inception V3 model and CNN model. Data augmentation is done concerning the techniques of zooming and rotating. Through data augmentation, the dataset size and efficiency will be increased. Then dataset training and testing are done concerning the dataset taken. Inception V3 and CNN models are implemented on a dataset to detect the fire and give the message that whether there is a fire or not. If it is a fire image then it will send the message "Fire is present" with the image of that. If it is a non-fire image then the model sends the message that "There is no fire" with the image.

Now we have developed a "Real-time fire detection system" again based on inception V3 and CNN models. Here, the camera will be activated to capture pictures from outside by using an open CV rather than giving the fire image to the model. There will be a capture option when we click to capture it captures the outside scenario and predicts that is their fire is present or not. If there is fire, it sends the message "Fire is there" along with the alarm sound. If there is no fire then send a message "Fire is not present".

Now we have taken the basic CNN model to detect the fire. Now we compared the inception V3 model and the basic CNN model in terms of accuracy. Inception v3 stood out well. Inception V3 with an image dataset has an accuracy of 99%. So, finally, Inception V3 performed well in fire detection in terms of speed and accuracy when compared to the basic CNN. In the proposed work we have taken two models Inception V3 and CNN and also taken a dataset. We have done dataset augmentation to have efficient performance, we have trained two models with the dataset. While testing we have taken two scenarios, the first one is uploading the image to test and the second one is capturing real-time scenarios and predicting whether a fire is present or not. In this work we have attained 99% accuracy for inception V3 and 89% for CNN, also inception V3 is computationally low cost and executes fast so, we suggest inception V3 as a good model for detecting fire.

Literature Review

Li, P., & Zhao, W. (2020) et al, proposed a faster-RCNN, R–FCN, SSD, and YOLO v3 are advanced object detection CNN models that were used to develop a novel image fire detection algorithm. The proposed algorithms can extract complex image fire features automatically and detect fire in a variety of scenes. The author also claimed that the algorithms based on CNN are better than the traditional algorithms in terms of accuracy. Among all the CNN models the highest accurate algorithm is based on YOLO v3, With an accuracy of 83.7 percent, it detects fire the fastest, at a rate of 28 frames per second.

Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S. W. (2018) et al, proposed a cost-effective fire detection Convolutional neural network (CNN) architecture for surveillance videos. The proposed model mainly focused on computational complexity and detection accuracy. The model is inspired by Google Net architecture as it has less computational complexity compared to other computationally expensive networks such as AlexNet. The author claimed the proposed framework works better on fire datasets and is suitable for fire detection in CCTV surveillance systems in real-life applications.

Jeon, M., Choi, H. S., Lee, J., & Kang, M. (2021) et al, the author proposed a framework that strengthens the existing Convolutional neural network-based fire image classification model by focusing on the varying sizes of fires in images. The author proposed a feature-squeeze block to utilize the feature maps of various scales in the final prediction. The feature-squeeze block squeezes the feature maps spatially and channel-wise enabling efficient utilization of the information from the multi-scale prediction. The experiment of the suggested approach produced an F1-score of 97.89% and a false positive rate of 0.0227 percent.

Li, Y., Zhang, W., Liu, Y., & Jin, Y. (2022) et al, Using the lightweight Convolutional neural network model MobileNetV3 and the anchor free structure, a fast and efficient fire detection model is created. The proposed method outperforms in two ways. First, to increase network speed, and second, the proposed technology is tiny enough to be easily implemented on visual mobile devices. The accuracy of the model has been experimented on both self-built datasets and two public fire datasets. The maximum speed of the suggested framework is 29.5f/s, which can satisfy real-time detection which means suitable for fire detection systems in real-life applications.

Jadon, A., Varshney, A., & Ansari, M. S. (2020) et al, To solve the fire detection challenges, the author suggested a convolutional neural network model called MobileNetV2 architecture. The author presents a new MobileNetV2 architecture as well as a more transparent data handling mechanism that outperforms existing solutions while being computationally practical for deployment on less capable hardware. On two datasets, the metrics Accuracy, Precision, Recall, and F-Measure were used to compare this model to contemporary Convolutional neural networks models. This proposed model's accuracy is 0.99, which is the highest of all models (99 %).

Zhang, Q., Xu, J., Xu, L., & Guo, H. (2016, January) et al, The author of this research proposed a deep learning system for detecting forest fires. The author trained both a full image and fine-grained patch fire classifier in a joined deep convolutional neural network (CNN). Here the author detected the fire through the method and detection is operated in a cascaded fashion. The proposed fire patch detector obtains 97% and 90% detection accuracy on training and testing datasets respectively.

Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019, July) et al, the author focused on a forest fire detection algorithm by exploiting YOLOv3 to UAV-based aerial images. Here Firstly, a UAV platform for forest fire detection is developed. Then, using YOLOv3, a small-scale convolution neural network (CNN) is created based on the onboard hardware's available compute capability. This model shows an accuracy of 83% and the author says that This method has great advantages for real-time forest fire detection applications using UAVs.

Dua, M., Kumar, M., Charan, G. S., & Ravi, P. S. (2020, February) et al, In this paper the author says the use of transfer learning is based on a deep CNN approach to detect fire. Here the various models of CNN are tested on imbalanced datasets to imitate real-world scenarios and compared with each other. He said that deep CNN models outperform well than the traditional CNN models. The author finally said that the accuracy of MobileNet is roughly the same as VGGNet, however, MobileNet is smaller in size and faster than VGG.

Saeed, F., Paul, A., Karthigaikumar, P., & Nayyar, A. (2020) et al, the author proposes the Adaboost-LBP model and a convolutional neural network for fire detection utilizing videos and images captured by surveillance cameras. The Adaboost-LBP model produces good results, with a 99 percent accuracy rate. False alarms are extremely uncommon. The Adaboost-LBP model's goal is to discover emergencies in an image and build an ROI for that object. Because that model's false alarm rate is significant, we tried these ROIs in our CNN model to reduce false alarms.

Gotthans, J., Gotthans, T., & Marsalek, R. (2020, April) et al, Based on the computational performance of the NVIDIA Jetson Nano, they provide a low-cost deep CNN architecture for video fire detection. The measured accuracy of the fire detection network model using embedded hardware was 79.66 percent in dataset 1 and 88.0 percent in dataset 2. On dataset1, the AlexNet model with 81.66 percent accuracy and the Squeezenet model with 81.66 percent accuracy were both tested. We limited ourselves to fire and smoke detection because these are the most common events where early detection can prevent fire disasters and potentially reduce the number of victims and costs.

Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Moreau, E., & Fnaiech, F. (2016, October) et al, the author of this research proposes using a convolutional neural network to identify a fire in videos (CNN).In the field of object classification, convolutional neural networks outperform quite well. The approach improves feature extraction and classification performance. It is also mentioned that scanning the feature map directly instead of the whole original picture during the detection test could reduce the time cost by a factor of 6 to 60. Our model can only identify red fire; to detect other colors of fire, we must expand our training set to include additional fire colors such as blue, etc.

Iqbal, M., Setianingsih, C., & Irawan, B. (2020, August) et al, backpropagation and Convolution neural networks (CNN) for object recognition and fire patterns were proposed in this paper. Using the HOG feature extraction and backpropagation algorithm as a classification approach, it is possible to classify fire photos effectively. The backpropagation technology used in the fire detection system has a 95 percent accuracy rate. CNN, on the other hand, has a 97 percent accuracy rate. The CNN algorithm, which employs the Inception-v3 model, bottleneck features, and high-resolution pictures, has a 97 percent accuracy rate.
Zhang, Q., Ge, L., Zhang, R., Metternicht, G. I., Liu, C., & Du, Z. (2021) et al, Using the Sentinel-2 imager, presents an automated active fire detection methodology. The framework's active fire detection module was built on a specially constructed DCPA+HRNetV2 network. On active fire detection, the DCPA and HRNetV2 combination beat DeepLabV3 and HRNetV2 models, according to the results. This dataset can be used as a baseline for other deep-learning-based active fire detection systems to enhance accuracy. The framework consists of three main modules: data collecting and preprocessing, deep-learning-based active fire detection, and final product development.

Kim, B., & Lee, J. (2019) et al, The proposed method uses Faster Region-based Convolutional Neural Network (R-CNN) to detect the suspected regions of fire Computer vision-based fire detection was focused on the color of a fire within the framework of a rule-based system. To train this model the author used a large fire dataset that contains images and video clips that enhance the data from wellknown datasets. The method is experimentally proven that excellent fire detection accuracy by reducing the false detections and misdetections. Exploring the static and dynamic features of various flame and smoke to be utilized in a visual system is not an easy process in general, since it necessitates a substantial amount of subject expertise.

Xu, R., Lin, H., Lu, K., Cao, L., & Liu, Y. (2021) et al, a novel ensemble learning method is proposed to detect forest fires in different scenarios. In this Yolov5 and EfficientDet are integrated to accomplish the fire detection process. The successful application of convolutional neural networks significantly improves the performance of object detection. EfficientNet is introduced to guide the detection process to reduce false positives. Forest fires are notorious for spreading quickly and being difficult to put out in a short amount of time. As a result, spotting an early forest fire before it spreads is critical, yet typical detection technologies have apparent shortcomings in open forest environments. Sensor-based detection systems operate well in the indoor environment, but they are challenging to install outside due to the high cost of coverage.

Zhong, Z., Wang, M., Shi, Y., & Gao, W. (2018) et al, In this paper, a new color fire feature detection and recognition scheme based on a convolutional neural network has been proposed. One of the most important jobs of a modern surveillance system is computer vision-based fire detection. Because of its high accuracy recognition rate in a wide range of applications, the convolutional neural network (CNN) has become a hot topic. They are classified using a CNN-based deep neural network model. Finally, the classification results yield the associated alert signal.

Rahmatov, N., Paul, A., Saeed, F., & Seo, H. (2021) et al, the author proposed Convolutional Neural Network which is a state space navigational model for fire detection. Here the author compared the approach with both uniformed and informed search algorithms. This method is used for finding the best way to the target site by following suitable navigational procedures and arriving at an optimal solution. The Multi-Agent based route discovering search algorithm is used by the State Space Cooperative exploration agent for disaster rescue. The author finally said that this would save time for the agents to travel to the target place.

Chen, Y., Zhang, Y., Xin, J., Wang, G., Mu, L., Yi, Y., ... & Liu, D. (2019, June)Et al, the author proposed an image-based forest fire detection approach using unmanned aerial vehicles (UAVs). Here firstly, the smoke detection system employs the local binary pattern (LBP) feature extraction and the support vector machine (SVM) classifier. This method is performed on two CNN models that detect smoke and flame. The "circular equivalent rotation-invariant LBP mode (tight RILBP)" is employed in the detection of smoke based on the result of feature extraction. In LBP+SVM for smoke detection, the local receptive domain is CNN's first key feature. Two CNN models are used to detect smoke and flames. The author finally said that when compared to the CNN-9 model the CNN-17 model reduces the algorithm's complexity and enhances detection accuracy.

Maksymiv, O., Rak, T., & Peleshko, D. (2017, February) et al, the author proposed a Convolutional Neural Network (CNN) which is the combination of Adaboost and Local Binary Pattern (LBP) models are used for getting Region of Interest (ROI). These methods are used to speed up the processing time. Furthermore, there are a handful of approaches for detecting movement in a video clip. In the context of fire detection, the fundamental shortcoming of these systems is that flames take on varied colors and speeds. The author tested the database on 500 photos, which contain categories like smoke, fire, and others. The proposed experiments reveal that the approach can obtain a detection rate of above 95%.

Geetha, S., Abhishek, C. S., & Akshayanat, C. S. (2021) et al, the author proposed Convolutional Neural Network (CNN) model for machine vision-based fire detection. There have been a variety of approaches described for automatically detecting and predicting flame and smoke in movies and photos using Deep Learning and Convolutional Neural Networks (CNNs). Here the traditional, image processing method is used for classifying the Support Vector Machine (SVM) is used and also the CNN models are demonstrated with the Alex Net, ResNet, GoogLeNet, and VGG-Net.

We reviewed above papers as part of this literature review. Each paper consists of the work of fire detection using a few models, almost all the papers are based on deep learning and Convolutional Neural networks in deep learning. Each paper consists of different models like VGG16, ResNet, MobileNet, YoloV3, R-CNN, AlexNet, etc. In the studied papers they have used the transfer learning technique and pre-trained models so that existing knowledge can be used and build the required model.

Methodology

The proposed model takes the image as an input from the user and determines whether fire exists in the image or not. There have been two different sorts of models proposed. They are:

i) Data Pre-processing
ii) Image Classification

Data pre-processing

Data pre-processing is the process of preparing raw data for usage with machine learning models. It is the most important and first step in creating a machine learning model. We don't always come across clean and prepared data when working on a machine learning project. Furthermore, before any process, data must be cleansed and prepared. As a result, we use data pre-processing services.
Steps in Data Pre-processing in Machine Learning:

Acquire the dataset

The acquisition of the dataset is the initial stage in data pre-processing in machine learning. Before you can create and evolve Machine Learning models, you must first gather the required dataset. This dataset will be constructed from data collected from a variety of sources, which will then be integrated into a suitable format to make a dataset. The format of a dataset depends on the application.

Import all the crucial libraries

Python is the most extensively used and recommended library among Data Scientists worldwide. The Python libraries can be used to conduct specialized data pre-processing tasks. The second stage in machine learning data pre-processing is to import all of the necessary libraries. The following Python libraries are often used in Machine Learning for data pre-processing:
NumPy - The most popular Python package for scientific calculations is NumPy. As a result, it's used to add any form of mathematical operation to the code. Large multidimensional arrays and matrices can also be used in NumPy programmers.
Pandas - Pandas are superb Python data manipulation and analysis tools that are open-source. It's frequently utilized for data collection, import, and upkeep. It includes Python data structures and data analysis tools that are fast and simple to use.
Matplotlib - Matplotlib is a Python 2D charting toolkit that may be used to create a variety of charts. It can produce publication-quality numbers in a variety of hard copies and interactive forms.

Splitting the Dataset into the Training set and Test set

We divide our dataset into a training set and a test set during machine learning data preprocessing. This is an important step in data pre-processing since it improves the performance of our machine learning model. Assume we've trained our machine learning model with one dataset and then tested it with a different dataset. Our model will then struggle to comprehend the links between the models. If we correctly train the model and it has a high training accuracy, but then give it a new dataset, its performance will suffer. As a result, we make every effort to develop a machine learning model that works well with both training and test datasets. These datasets can be defined as follows:

Data Augmentation

To artificially enhance the size of an actual dataset, data augmentation techniques generate different versions of it. Computer vision and natural language processing (NLP) models use data augmentation methodologies to deal with data scarcity and insufficient data diversity. Data augmentation tactics can help machine learning models. According to an experiment, a deep learning model with picture augmentation performs better in image classification task training and accuracy, as well as validation loss and accuracy, than a deep learning model without image augmentation.

Data augmentation techniques

- Adding noise: Adding noise to a blurry photograph might help it stand out. The image appears to be made up of white and black dots when it is called "salt and pepper noise."

- Cropping: An area of the photograph is chosen, cropped, and resized to its original size.
- Flipping: The image is horizontally and vertically flipped. Flipping the image protects the image's features while rearranging the pixels. For some photographs, vertical flipping is pointless.
- Rotation: The image is rotated by several degrees between 0 and 360. In the model, each rotated image will be unique.
- Scaling: The image has been resized in and outward. Scaling allows an object in a new image to be smaller or larger than it was in the original image.
- Translation: Because the image is displaced along the x-axis or y-axis, the neural network looks for it everywhere in the image.
- Brightness: The image's brightness is altered, and the new image will be darker or lighter. This method enables the model to detect images in various lighting conditions.
- (viii)Zooming: In the Zooming Augmentation technique, the image is randomly zoomed and new pixels are added.

Image Classification

The labeling of images into one of several predefined classifications is referred to as image classification. A single image can be categorized into an infinite number of categories. Manually reviewing and classifying images takes time, especially when there are a lot of them, as a result, using computer vision to automate the procedure would be quite beneficial. Convolutional neural networks, or CNNs, are commonly used in deep learning image classification. The output of the nodes in the hidden layers of CNNs is not always shared with every node in the following layer. Machines can recognize and extract information from photographs using deep learning**.**

We employ Inceptionv3, a deep learning model based on Convolutional Neural Networks, for image classification. On the ImageNet dataset, Inception v3 is an image recognition model that has been shown to achieve higher than 78.1 percent accuracy. In comparison to the Inception V1 and V2 models, the Inceptionv3 model has higher efficiency, is computationally less expensive, and has a deeper network. It's a 48-layer deep pre-trained convolutional neural network model. It's a version of the network that's already been trained on millions of photos from the ImageNet collection**.**

Inception Networks are more computationally efficient than VGGNet, both in terms of the number of parameters generated and the cost incurred. If changes to an Inception Network are made, care must be taken to ensure that the computational advantages are not lost. As a result, because of the uncertainty of the new network's performance, adapting an Inception network for multiple use cases becomes an issue. Several strategies for improving the network have been proposed in an Inception v3 model to loosen the constraints for faster model adaptation. Factorized convolutions, regularization, dimension reduction, and parallelized calculations are some of the techniques used. Factorization into Smaller Convolutions, which reduces the computational cost, is one of the significant changes made to the Inception V3 model. Spatial Factorization into

Asymmetric Convolutions replaces the 55% convolutional layer with two 33% convolutional layers. It's the same as sliding a two-layer network with the same receptive field as a 33% convolution.

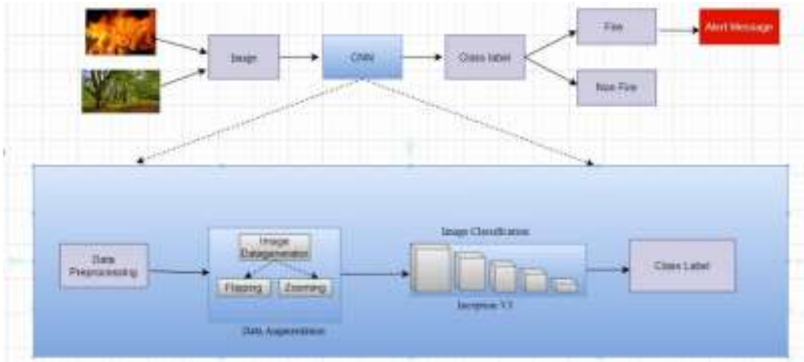

Fig 1: Flowchart of the proposed model

Inception V3

Convolutional Neural Networks are used in the Inception V3 deep learning model for picture categorization. The Inception V3 is a more advanced version of the basic model Inception V1, which was first released as GoogLeNet in 2014. It was created by a Google team, as the name implies. The Inception v3 model, which was launched in 2015, features 42 layers and a reduced error rate than its predecessors.

i. Factorization into Smaller Convolutions
ii. Asymmetric Convolutions from Spatial Factorization
iii. Utility of Auxiliary Classifiers
iv. Efficient Grid Size Reduction

Factorization into smaller convolutions

Convolutions with bigger spatial filters are disproportionately computationally costly. For example, computing a 5 5 convolution with n filters across a grid with m filters costs 25/9 = 2.78 times as much as a 3 3 convolution with the same number of filters. However, a 55 filter can catch relationships between signals between activations of units in previous layers, thus reducing the physical size of the filters comes at a significant sacrifice in terms of expressiveness. We could, however, see if a 5*5 convolution is possible. replaced by a multi-layer network with fewer parameters with the input and output depths being the same When we zoom in on the 5*5 convolution computation graph, we can observe that each output seems to be a little fully connected network moving over 5*5 tiles versus its input. Because we're building a vision network, it seems natural to take advantage of translation invariance and replace the fully connected component with a two-layer convolutional architecture: the first layer is a 3*3 convolution, and the second layer is a fully connected layer on top of the first layer's 3*3 grid of output.

Spatial Factorization into Asymmetric Convolutions

The filters greater than 3*3 a may always be converted into a sequence of 3*3 convolutional layers, the previous results show that they may not be effective in general. Still, we may consider if they should be factorized into smaller units, such as 22 convolutions. However, it turns out that utilizing asymmetric convolutions, such as n 1, one may perform much better than 2*2. For example, sliding a two-layer network with the same receptive field as in a 3*3 convolution is equal to employing a 3*1 convolution followed by a 1*3 convolution. Even still, with the same number of output filters, the two-layer technique is 33% less expensive. By contrast, factoring a 3*3 convolution into two 2*2 convolutions saves just 11 percent of the computing time.

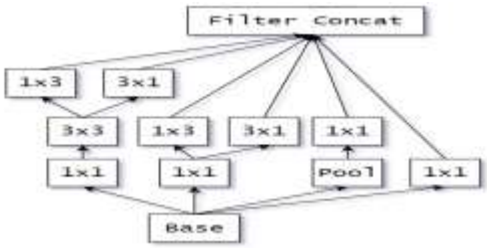

Fig 2: Filter concat

Utility of Auxiliary classifiers

Auxiliary classifiers have been proposed as a method of speeding up the convergence of very deep networks. The initial objective was to push valuable gradients to lower layers so that they could be employed immediately, and to improve training convergence by addressing the vanishing gradient problem in excessively deep training. They make learning and convergence more constant.

Interestingly, early in the training, we discovered that auxiliary classifiers did not help convergence: the training Before both models reach high accuracy levels, the evolution of the network with and without side head looks to be nearly identical. Near the conclusion of training, the network with auxiliary branches begins to outperform the network without auxiliary branches in terms of accuracy and achieves a slightly higher plateau.

Efficient Grid Size Reduction

Reducing Grid Size Effectively To minimize the grid size of feature maps, convolutional networks have usually used a pooling strategy. Before implementing maximum or average pooling, the activation dimension of the network filters is increased to prevent are a presentational bottleneck. For example, if we want to get to a d 2*d 2 grid with 2k filters, we can start with a d*d grid with k filters. There are two different ways to reduce grid size. The method on the left breaches Section 2's principle 1 of avoiding adding a representational bottleneck. Computationally, the variant on the right is three times more costly.

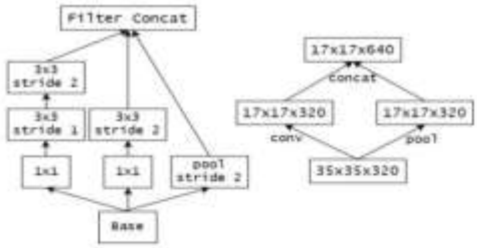

Fig 3: Stride grid

TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Conv	3×3/1	149×149×32
Conv padded	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Conv	3×3/1	73×73×64
Conv	3×3/2	71×71×80
Conv	3×3/2	299×299×3
Conv	3×3/1	35×35×192
3 × Inception	Module 1	35×35×288
5 × Inception	Module 2	17×17×768
2 × Inception	Module 3	8×8×1280
Pool	8 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
SoftMax	Classifier	1 × 1 × 1000

Table 1: Patch/Stride Size

After all of the adjustments, the finished Inception V3 model looks like this. The Inception V3 model has 42 layers in total, which is somewhat more than the previous Inception V1 and V2 models. However, this model's efficiency is truly remarkable. We'll get to it in a minute, but first, let's take a look at the components that make up the Inception V3 model. The outline of the concept V3 model is shown in the table below. Each module's output size is the following module's input size in this case. Inception Networks are more computationally efficient than VGGNet, both in terms of the number of parameters produced and the cost incurred.

Result and Discussion

In this work, we have taken a fire image dataset that contains 1820 images. Dataset is trained under inception v3 model and CNN model as mentioned below in figure 4. Before training the model, data augmentation is performed on the dataset for better performance. Then, the dataset is trained under the inception V3 model and CNN model. Finally, while testing we have taken two different scenarios. The first one is based on a fire image dataset and the second one is based on a real-time fire detection system using an open CV as shown in figure 5.
In a real-time scenario, there is a capture option to capture the real scenario and predicted whether the fire is present or not along with the alarm sound in case of

fire. The below table showed the accuracy of models and finally, inception V3 performed well compared to basic CNN in terms of speed and accuracy.

Scenario	CNN Model	InceptionV3 Model
Fire image	89% (Accuracy)	99% (Accuracy)
Real scenario(Captured)	89% (Accuracy)	99% (Accuracy)

Table 2: Result of Accuracy

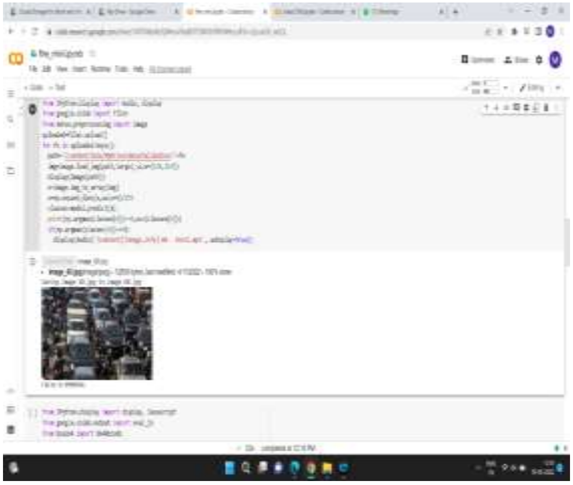

Fig 4: Dataset for fire detection

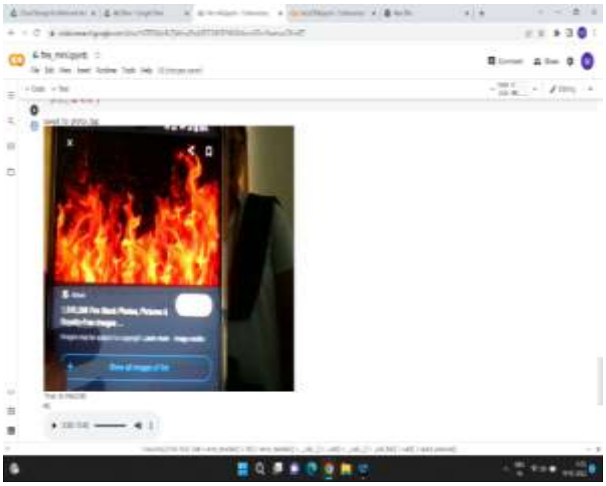

Fig 5: Image scan from Open CV

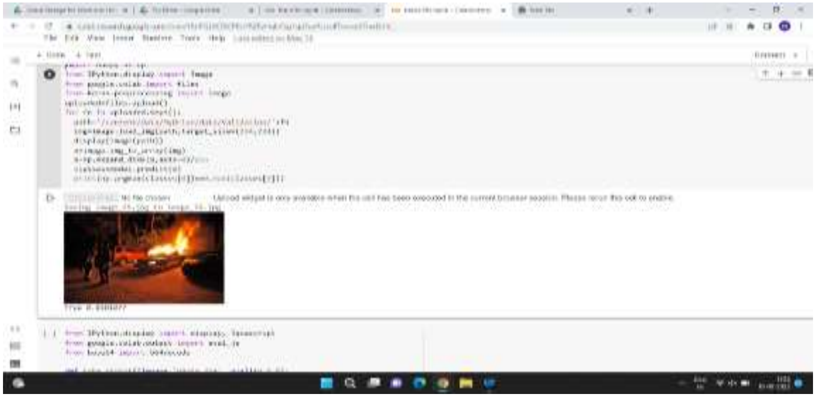
Fig 6: Fire detection using CNN

Conclusion

Fire is an abnormal event that can cause severe harm to people, animals, and property in a short period. Early fire detection systems are significant nowadays because they can provide an early warning system, potentially saving lives and decreasing property damage. In this work, we looked at the inception V3 and Convolutional Neural Network model which automatically detects the fire in the images. Their performance is compared using the accuracy metric. Among the chosen models, the inception V3 is computationally efficient and is more accurate. Inception V3 has given better performance (accuracy) in every scenario. Inception V3 has given an accuracy of 99%, and as CNN is a basic model, it has given an accuracy of 89%. The suggested model not only accepts inputs from the local system but can also catch images from the camera and accurately forecast whether or not a fire is present. Through this system, the fire can be identified through surveillance cameras and an alert will be sent to the forest officers so that fire can be controlled.

References

1. Li, P., & Zhao, W. (2020). Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering*, *19*, 100625.
2. 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.
3. Jeon, M., Choi, H. S., Lee, J., & Kang, M. (2021). Multi-scale prediction for fire detection using convolutional neural network. *Fire Technology*, *57*(5), 2533-2551.
4. Li, Y., Zhang, W., Liu, Y., & Jin, Y. (2022). A visualized fire detection method based on a convolutional neural network beyond anchor. Applied Intelligence, 1-16.
5. Jadon, A., Varshney, A., & Ansari, M. S. (2020). The low-complexity high-performance deep learning model for real-time low-cost embedded fire detection systems. *Procedia Computer Science*, *171*, 418-426.

6. Zhang, Q., Xu, J., Xu, L., & Guo, H. (2016, January). Deep convolutional neural networks for forest fire detection. In Proceedings of the 2016 international forum on management, education and information technology application. Atlantis Press.
7. Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019, July). A deep learning-based forest fire detection approach using UAV and YOLOv3. In *2019 1st International conference on industrial artificial intelligence (IAI)* (pp. 1-5). IEEE.
8. Dua, M., Kumar, M., Charan, G. S., & Ravi, P. S. (2020, February). An improved approach for fire detection using deep learning models. In *2020 International Conference on Industry 4.0 Technology (I4Tech)* (pp. 171-175). IEEE.
9. Saeed, F., Paul, A., Karthigaikumar, P., & Nayyar, A. (2020). Convolutional neural network-based early fire detection. *Multimedia Tools and Applications*, *79*(13), 9083-9099.
10. Gotthans, J., Gotthans, T., & Marsalek, R. (2020, April). Deep convolutional neural network for fire detection. In *2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA)* (pp. 1-6). IEEE.
11. Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Moreau, E., & Fnaiech, F. (2016, October). Convolutional neural network for video fire and smoke detection. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society* (pp. 877-882). IEEE.
12. Iqbal, M., Setianingsih, C., & Irawan, B. (2020, August). Deep Learning Algorithm for Fire Detection. In 2020 10th Electrical Power, Electronics, Communications, Controls, and Informatics Seminar (EECCIS) (pp. 237-242). IEEE.
13. Zhang, Q., Ge, L., Zhang, R., Metternicht, G. I., Liu, C., & Du, Z. (2021). Towards a Deep-Learning- Based Framework of Sentinel-2 Imagery for Automated Active Fire Detection. *Remote Sensing*, *13*(23), 4790.
14. Kim, B., & Lee, J. (2019). Video-based fire detection using deep learning models. *Applied Sciences*, *9*(14), 2862.
15. Xu, R., Lin, H., Lu, K., Cao, L., & Liu, Y. (2021). A forest fire detection system based on ensemble learning. *Forests*, *12*(2), 217.
16. Zhong, Z., Wang, M., Shi, Y., & Gao, W. (2018). A convolutional neural network-based flame detection method in a video sequence. *Signal, Image and Video Processing*, *12*(8), 1619-1627.
17. Rahmatov, N., Paul, A., Saeed, F., & Seo, H. (2021). Realtime fire detection using CNN and search space navigation. *Journal of Real-Time Image Processing*, *18*(4), 1331-1340.
18. Chen, Y., Zhang, Y., Xin, J., Wang, G., Mu, L., Yi, Y., ... & Liu, D. (2019, June). UAV image-based forest fire detection approach using convolutional neural network. In *2019 14th IEEE conference on industrial electronics and applications (ICIEA)* (pp. 2118-2123). IEEE.
19. Maksymiv, O., Rak, T., & Peleshko, D. (2017, February). Real-time fire detection method combining AdaBoost, LBP, and convolutional neural network in a video sequence. At the *2017 14th international conference the experience of designing and application of CAD Systems in microelectronics (CADSM)* (pp. 351-353). IEEE.
20. Geetha, S., Abhishek, C. S., & Akshayanat, C. S. (2021). Machine Vision-Based Fire Detection Techniques: A Survey. *Fire Technology*, *57*(2), 591-623.