Wildfire Prediction Using Modified Convolutional Neural Networks and Chromatic Variants

D. Crystal Jaba Kani[1], S. Saudia[2]
[1]*Research Scholar, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli- 627012, India*
[2]*Associate Professor, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli- 627012, India*

Abstract- Environmental disaster by wildfire is a serious issue of concern. This paper proposes an analysis on the performance of various CNN based Deep Learning models, AlexNet, Lenet, VGG16 and their modified versions in the color spaces: RGB, YUV and HSV for predicting wildfire. The dataset with 'Fire' and 'NoFire' images is taken from the DeepFire dataset. The images in the RGB color space are converted into YUV and HSV spaces before using the images for training the CNN models and their modified versions. The modified version of LeNet5 produces higher accuracy than all versions. YUV color space produces high accuracy with original LeNet5 and AlexNet models, RGB color space produces high accuracy with modified LeNet5 and AlexNet models. HSV space is better for the VGG16 models.

Keywords: Color Spaces, Convolution Neural Network (CNN), Deep Fire, Deep Learning, Forest Fire

I. INTRODUCTION

Forests are large green areas with dense grass, plants, trees and crowded habitation of animals, birds, other lives. Forest area covers 31% of the world land mass[1]. Forest fire occurs when forest fuel catches fire under suitable heat, light conditions, wind intensity, carbon dioxide and water content[2]. It is a major issue to the environment since it affects human and animal health, greenland, trees, water, air and all forest life. So forest fire prediction is important as a precaution against disaster. Many papers are available in literature [3-15] for forest fire detection and prediction using the Deep Learning (DL) techniques. DL involves depth learning of data such as text, audio, video and images to train machines take detection, prediction, recognition, tracking decisions similar to the humans. Different DL algorithms in literature are Multilayer Perceptron (MLP), Long Short-Term Memory Networks (LSTM), Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN), Recurrent Neural Networks (RNN) etc[3]. CNN also called ConvNet is a Deep Neural Network (DNN) with many numbers of convolution layers and fully connected layers. The various CNN architectures, AlexNet, LeNet, ResNet, GoogleNet, MobileNet, Visual Geometry Group (VGG) and their versions extract features from images using Convolution layers and classify the features into classes in the fully connected layers[4]. The two components of forest fire are flame and heat. The different light/ colours of the flame are visible specifically in the different color spaces like RGB (Red, Green, Blue), YUV (Y-luminance, U-blue chrominance, V-red chrominance), and HSV (Hue, Saturation, Value)[5]. Identifying a suitable color space for fire prediction shall improve the prediction accuracy of the CNN models. So, color space conversion is done as a preprocessing technique before subjecting the CNN algorithms for training. Many color space based DL techniques are in literature[3-15]. Hazim Gati' Dway and Ahmed Fakhir Mutar[6] studied fire color features using set of color spaces, RGB, HSV, Lab, Yiq by calculating Scale factor that depends on the histogram of hue values. The best color space was identified as HSV for fire detection [6]. Vipin.V. proposed an image processing based Forest Fire Detection [7] using RGB and YCbCr color spaces. YCbCr color space separates the luminance from the chrominance more effectively than RGB color space. Real-time video fire/ smoke detection based on CNN was proposed by Sergio Saponara and

Abdussalam Elhanashi [8] using YOLOv2 CNN in antifire surveillance systems. It could detect fire and smoke in 1 or 2 seconds as an early alerting alarm for the occurrence of fire and smoke accidents surveillance systems. Dai Quoc Tran and Minsoo Park proposed a Forest-Fire Response System using Deep-Learning based approaches with CCTV Images and Weather Data[9] for early fire detection and damaged area estimation using Residual Neural Network (ResNet), VoVNet, FBNetV3, Robust Granular Neural Networks (RGNN) and Bayesian Neural Network (BNN). The RGNN got a mean average precision of 27.9[9]. An image processing based forest fire detection[10] proposed by Vipin Venugopal used RGB and YCbCr color spaces for fire pixel classification. It gave an accuracy of 92.5%[10]. Rafik Ghali and Moulay A. Akhloufi et.al., proposed forest fire segmentation using the deep CNN like U-Net, U2-Net, Efficient Segmentation[11] and vision-based fire detection methods[11].

Zhentian Jiao and Youmin Zhang et. al. proposed a DL based forest fire detection approach using YOLOv3[12] for UAV-based aerial images that got test result accuracy of 83%[12]. A forest fire identification system based on Weighted Fusion Algorithm[13] was proposed by Jingjing Qian and Haifeng Lin using weighted fusion of YOLOv5 and Efficient Det to identify forest fire sources in different scenarios that gives the detection performance accuracy of 85%[13]. Lightweight forest fire detection based on Deep Learning[14] proposed by Ruixian Fan, Mingtao Pei used YOLOv4 feature extraction network for forest fire detection[14]. Jianmei Zhang and Hongqing Zhu proposed a ATT Squeeze U-Net for forest fire detection and recognition with an accuracy of 93%[15]. Wang Yuanbin and Dang Langfei et al., proposed a forest fire image recognition based on CNN that gave the accuracy 80.4%[16]. Dongqing Shen proposed flame detection using DL to detect flames at an accuracy of 80%[17]. S. Gayathri and P.V. Ajay Karthi et.al proposed prediction and detection of forest fire based on a DL approach that gives the detection accuracy of 92%[18]. Thus, color spaces and DL models are investigated in the domain of forest fire detection.

Accordingly, this paper proposes an analysis on the performance of various color features RGB, HSV in predicting forest fire using the Convolution Neural Network (CNN) based DL models, AlexNet, LeNet, VGG16 and their modifications. The DeepFire dataset with 'Fire' and 'NoFire' images are taken from the reference [19]. Fire images include fire and smoke images and non-fire images include nature scenery and forest images. The images in the RGB color space are converted into YUV and HSV spaces before using the images to train the CNN models. The paper evaluates the performance of the CNN models and their modified versions using Accuracy and Area Under Receiver Operating Characteristics curve (AU-ROC) will be discussed. The conclusion will be listed in the end of the paper.

II. METODOLOGY

The paper proposes a performance analysis of different CNN models namely AlexNet, LeNet and VGG16 and their modifications in different color spaces: RGB (Red, Green, Blue), YUV (Y-luminance, U-blue chrominance, V-red chrominance) and HSV (Hue, Saturation, Value) for forest fire prediction in terms of Accuracy and AU-ROC. The methodology of the work is shown in Fig 1 and briefed in sub-sections below:

A. Data Collection

This paper uses the DeepFire image dataset collected from the reference [19]. These images captured from different data sources include forest fire, forest smoke images and forest scenery images. There are 759 'Fire' images and 759 'NoFire' images in the positive 'Fire' folder and the negative 'NoFire' folder of the training dataset respectively in various sizes: 1200x700 pixels, 750 x 450 pixels, 400 x 300 pixels etc. Fig. 1 shows sample 'Fire' and 'NoFire' images in the DeepFire dataset. The images in the Fire and NoFire folders are combined into a single dataset by assigning respective labels to the images as 'Fire' and 'NoFire'. The images which are in BGR (Blue, Green, Red) format by default in opencv are converted to RGB colour space for further processing [20]. Different operations in the methodology of the work are performed using the functions from opencv, keras and tensorflow libraries of Python.The different colors of the flame in the RGB color space as seen Dark Red, Dull Red, Bright Cherry Red, Orange, Bright Yellow and White in Fig. 2 correspond to different temperature ranges of the fire as mentioned in Table 1[21]. The fire features are visible differently in different color spaces as in Fig.

2-A and Fig. 2-B. Also it is found from literature that color information is not affected by illumination when images are investigated in YUV color space [11]. So, the work proposes to analyze the fire features of 'Fire' and 'NoFire' images in different color spaces: RGB, YUV and HSV for forest fire prediction.

A. Pre-Processing

Pre-Processing techniques are applied on the image data prior using the data in the training phase of Machine Learning/ Deep Learning algorithms. It involves cleaning image data by removing noises, resizing images, extracting image features, cropping images and color analysis. In this methodology, the 'Fire' and 'NoFire' images in different sizes are resized to 256x256 pixels. The resized image dataset is subjected to Label Encoding where the labels 'Fire' and 'NoFire' are encoded as '1' and '0' to facilitate coding in Python.

The dataset is then split into training and test datasets such that 80% of data is training dataset and 20% of data is test dataset. After splitting, the training dataset has 1093 images, test dataset has 304 images and validation set has 121 images. Then, to explore the possibility of identifying fire features in different colour spaces, the images of the balanced training, validation, and test datasets in RGB colour space are individually converted to YUV colour space & HSV colour space. The training dataset in different color spaces are separately subjected to the training phase of different CNN architectures: LeNet, AlexNet, VGG16 and their modifications. The details of different colour spaces are in sub-sections below.

RGB Color Space Model. The RGB (Red, Green, Blue)[23] color model is an additive color space. That color model has three different images, Red image, Green image and Blue image. The intensity values of RGB channels ranges from 0 to 255[23]. The color image matrix is formed by adding the intensities at specific pixel positions, (i, j) of the Red, Green and Blue color matrices as shown in (1)[23]. The maximum R, G and B values for forest fire image are identified as 255, 136 and 51 respectively [23]. Also, the wild forest fire color belong to orange family as a mixture of orange and brown colors [24,25] in RGB color space as shown in Fig. 1-A, B.

$$CIM(i,j) = RM(i,j) + GM(i,j) + BM(i,j) \tag{1}$$

where CIM is color image matrix, RM is red matrix, GM is green matrix and BM is blue matrix.

YUV Color Space Model. The YUV color model is a color model which provides better human color perception than RGB[26]. The different components in the YUV color space model are Y-luminance, U-blue chrominance, and V-red chrominance. Y values range from 0 to 255, U and V values range from -0.5 to 0.5[26, 28]. The formulae for conversion from RGB to HSV as per ITU BT.601 [27] in equations from (2) to (4) are used in this work. Fig. 2 shows the 'Fire' and 'No Fire' images in HSV color space.

$$Y = 0.257R + 0.504G + 0.098B + 16 \tag{2}$$

$$U = -0.148R - 0.291G + 0.439B + 128 \tag{3}$$

$$V = 0.439R - 0.368G - 0.071B + 128 \tag{4}$$

HSV Color Space Model. HSV color space has the attributes Hue, Saturation and Value. Hue (H) is the dominant color/ tint perceived by humans and it represents the actual color information such as red, green, and blue. Saturation (S) is the amount of white light mixed with Hue and Value (V) is the Brightness [28].

The images in RGB color space are converted to HSV color space using the formulae given in equations from (5) to (7). Fig. 3 shows the 'Fire' and 'No Fire' images in HSV color space. In the equations, MaxRGB is the maximum of Red, Green and Blue values and MinRGB is the minimum of Red, Green and Blue values. The images in different color spaces are then subjected to the training phases of the Deep Learning based CNN variants: AlexNet, LeNet5 and VGG16. The architectures and working of these CNN variants are briefed in the sub-section below.

$$H = \begin{cases} G - B / 6\left(Max_{RGB} - Min_{RGB}\right), \\ 1/6\left(2 + B - R / Max_{RGB} - Min_{RGB}\right), \\ 1/6\left(4 + B - R / Max_{RGB} - Min_{RGB}\right) \end{cases} \tag{5}$$

$$S = \frac{\left(Max_{RGB} - Min_{RGB}\right)}{Max_{RGB}} \tag{6}$$

$$V = Max_{RGB} \tag{7}$$

A. Convolution Neural Network (CNN)

Convolution Neural Network (CNN) is a class of Deep Neural Network which comes under the domain of Artificial Intelligence (AI). The different stages in the general architecture of CNN training and testing phases are shown in Fig. 5. CNN has multiple convolution layers as the first part of the network after the input layer. The convolutional layer is generally used to distinguish the essential features in different input images of the training dataset [29] using different sets of convolution filters. The convolution of an input image, 'I' with dimensions, (h x w x d) and a filter/ mask 'f' of size (k x k x d) produces an output 'g' of size (h-k+1) x (w-k+1) x 1[3031]. The convolution operation between the 3-dimensional (3D) filter and a color image for a pixel position is the sum of the products of the pixel values of 'I' in a region covered by the filter and the corresponding filter values. A two-dimensional convolutional operation is shown in (9). When a 3D filter is applied on a 3D color image, the filter extends to the depth of the color image and produces a 2D image as output. Padding and Striding functions are sometimes applied along with the convolution operations to prevent the loss of information and to capture relevant information respectively. The output due to a convolution is the feature map and when multiple filters are available in a convolution layer, many feature maps are obtained.

$$g(i, j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} f(u, v) I(i-u, j-v) \qquad (9)$$

The output of the convolution layer is subjected to an activation function. CNN uses ReLU, Soft max, Sigmoid or Tanh functions as activation functions depending upon the Use Case. The ReLU (Rectified Linear Unit) function is used in most CNN. The output of ReLU is the input itself when the input values are positive and zero when the input is negative [32] as shown in (10). The Pooling layer reduces the size of the feature map and attempts to fasten the training process. In the meanwhile, it reserves the important features of the input image. Max Pool and Average Pool are the two generally used Pooling techniques. Max Pool and Average Pool respectively produce the maximum and average of the intensities in the feature map covered by a pooling window as output[33]. Batch Normalization reduces training time by normalizing the input of each layer in the network. Dropout reduces overfitting by removing the contribution of some neurons to the next neuron layer [34].

$$R(g(i, j)) = \begin{cases} g(i, j) \ \text{if} \ g(i, j) \geq 0 \\ 0 \ \text{if} \ g(i, j) < 0 \end{cases} \qquad (10)$$

The fully connected layer is a feed-forward network which learns the likelihood of category/ class. The output from the Pooling layer is flattened into a 1D vector and fed to the fully connected network [32].The weights of the network between its input, hidden and output layers are trained using back propagation algorithm. The activation function used in the hidden layer of the fully connected part of the CNN for binary classification is the ReLU function which is defined in (10) for the input 'x' obtained from the flattening layer. The outputs are then applied to the Softmax function to calculate the probability of a class in class vector. In the proposed work, the two classes are 'Fire' and 'No Fire'. Softmax function values range from 0 to 1[36] and is defined as in (11) for the 'T' input values 'f' obtained from the Fully connected layer[32]. 'T' also corresponds to the number of classes. The different parameters of the CNN network, the weights of the convolution filters and the weights of the fully connected layer are optimized during the training phase. The optimized weights are used in the testing phase. In this work, the images in different color spaces are subjected to the training phases of the Deep Learning based CNN variants: AlexNet, LeNet5, VGG16 and their modifications. The architectures and working of these CNN variants are briefed below:

$$S(f) = \frac{e^{f_l}}{\sum_{l=1}^{T} e^{f_l}} \qquad (11)$$

LeNet5 Model. LeNet or LeNet-5 is a CNN architecture introduced by Unclench et al., in 1998[37]. The overall architecture of the network of the original version of LeNet5 has 2 Convolution layers, 2 Pooling layers, 2 Fully connected layers with a dense layer- [CONV1 POOL2 CONV3 POOL4 FC5 FC6]. In the convolution layer, CONV1, 64 convolution filters each of size 3x3 and stride 1 are applied on input images of size 256 x 256 x 3 to produce 64 feature maps of size 256 x 256. The convolution operations extract the important features of 'Fire' and 'NoFire' images from RGB, YUV and HSV color spaces. The Pooling layers, POOL2, POOL4 uses Max Pool technique with a pool size, 2 x 2 and stride 2. POOL2 produces 64 outputs of size 128 x 128. The second 2D Convolution layer, CONV3 uses 16 filters of size 3 x 3 to obtain 16 feature maps of size 64 x 64. POOL4 produces 16 outputs of size 64 x 64 using a pool size of 2 x 2 and stride 2. The output of POOL4 are flattened and the 65536 inputs ((64 x 64) x 16) are subjected to the fully connected layer, FC5. There are 684 neurons in FC5, the total trainable parameters being 44827308 ((65536 x 684) + 684). The output from 684 neurons in FC5 are fed to the fully connected layer, FC6 with 2 neurons. The 684 outputs from the FC5 are fed to the output fully connected layer, FC6 provided with soft max activation to determine the probabilities of the input images to belong to 'Fire' or 'No Fire' classes.The original architecture of LeNet5 is modified in terms of filter numbers for improved performance while finding the 'Fire' and 'NoFire' images. The summary of hyper parameters in the LeNet5 architecture used in the work, both original and modified versions is provided in Table 2.

AlexNet Model. The architecture of AlexNet is put forth by Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton [38] which won ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. The architecture of AlexNet used in this work has 12 layers: 5 convolutional layer, 3 pooling layers, 3 fully connected layers and a fully connected dense layer- [CONV1 POOL2 CONV3 POOL4 CONV5 CONV6 CONV7 POOL8 FC9 FC10 FC11 FC12]. Input images in different color spaces, RGB, YUV, HSV of sizes 256 x 256 x 3 are applied individually to first convolution layer, CONV1 where 96 convolution filters of size 11 x 11 and stride size 4 are applied to produce 96 feature maps of sizes 64 x 64. In POOL2, POOL4 and POOL8, the filters are of size 3 x 3 and stride 2 to produce outputs of sizes 32 x 32, 16 x 16 and 8x8 respectively using MaxPool technique. CONV3 produces 256 feature maps of size 32 x 32 with filters of size 5 x 5. The convolution layers, CONV5 CONV6 CONV7 uses 3x3 filters to produce 384, 384 and 256 feature maps respectively of size 16 x 16. The output from POOL8 are flattened to produce 16384 vector values which are fed to the fully connected layer, FC9. The fully connected layers, FC9, FC10 and FC11 have 1093, 1093 and 121 neurons respectively. The total trainable parameters of the fully connected layers are 17908805 (16384 x 1093) + 1093), 1195742 ((1093 x 1093) + 1093) and 132374 ((1093 x 121) + 121) respectively. The 121 outputs from the FC11 are fed to the output dense layer with the fully connected layer, FC12 provided with softmax activation to determine the probabilities of the input images to belong to 'Fire' or 'NoFire' classes. The ReLU activation function is used as activation function in all the convolution layers and other fully connected layers. The original architecture of AlexNet is modified in terms of filter numbers for improved performance while finding the 'Fire' and 'No Fire' images. The summary of hyperparameters used in the original and modified versions of AlexNet architecture used in the work is provided in Table 3. Batch Normalization is used after all convolution and fully connected layers. Dropout of 4% is used in all the fully connected layers and the dense layer.

VGG16 Model. VGG16 model is one of the CNN algorithms developed by the Visual Geometry Group of the Oxford [39] with increased depth and reduced filter sizes. The architecture is homogenous that the convolution filter sizes are all 3x3, with stride size 1 and pool size 1. The 5 Max Pool layers are all of size 2x2 and stride 2. It has 16 layers with learnable parameters which includes the 13 convolution layers and 3 dense layers. A small filter captures small region of the image and so the network is deeper in VGG1632. The architecture is [CONV1 CONV2 POOL3 CONV4 CONV5 POOL6 CONV7 CONV8 CONV9 POOL10 CONV11 CONV12 CONV13 POOL14 CONV15 CONV16 CONV17 POOL18 FC19 FC20 FC21]. The first convolution layer, CONV1 takes input shapes of size 256x256 in their three color space channels. CONV1, CONV2 layers has 64 filters, CONV4, CONV5 has 128 filters, CONV7, CONV8, CONV9 has 256 filters and the layers CONV11 to CONV17 has 512 filters. The fully connected layers, FC19, FC20 has 4096 output units and last dense layer, FC21 has 2 units with softmax activation to determine the probabilities of the input images to belong to 'Fire' or 'NoFire' classes.

The original architecture of VGG16 is modified in terms of filter numbers for improved performance while finding the 'Fire' and 'No Fire' images. The summary of hyperparameters used in original and modified versions of VGG16 is provided in Table 4.

III. RESULT AND DISCUSSION

In this paper, the performance evaluation metrics, Accuracy and Area Under Receiver Operating Characteristics (AU-ROC) curves are used to evaluate the classification of Forest fire images in various color space, RGB, YUV and HSV by the CNN models, LeNet5, AlexNet and VGG16 and their modified versions. The experiments are carried out in colab hardware accelerator by accessing a 2 core NVIDIA Graphics Processing Unit (GPU) with 1.59GHz speed using the python libraries, Keras, Tensorflow and OpenCV.

A. Accuracy

Accuracy is a metric for a classification problem which corresponds to the number of true predictions made by the classifier. It is defined as the ratio of sum of correct prediction are True Positive (TP) and True Negative (TN) values to sum of all values TP, TN, False Positive (FP) False Negative (FN) produced by the classifier [40].

$$\text{Accuracy}=\frac{[TP+TN]}{[TP+TN+FP+FN]} \tag{12}$$

B Area Under Receiver Operating Characteristic Curve (AU-ROC)

ROC curve is a plot drawn between True Positive Rate (TPR) and True Negative Rate (TNR) to show the classification performance of an ML model. TPR is defined as TN values to sum of TP and FN values. TPR and TNR are defined in equations below. A larger area under a curve corresponds to a better model when compared to other models under comparison [40].

$$\text{TPR}=\frac{[TN]}{[TP+FN]} \tag{13}$$

$$\text{TNR}=\frac{[FP]}{[FP+TN]} \tag{14}$$

C. Results

The forest fire prediction experiments are carried out in various color spaces using the trained AlexNet, LeNet5 and VGG16 models, both original and modified. The various color space models experimented on are RGB, YUV and HSV. The accuracy values produced by the models in different color spaces are recorded in Table 5.

From the table, RGB, YUV and HSV color space models produce high accuracy in the modified LeNet5 model than all versions of AlexNet and VGG16 models. Also, of all the color spaces, YUV and RGB color space is producing increased accuracy for original and modified versions of LeNet5 and AlexNet respectively. Also, HSV space is better for the original and modified versions of the VGG16. The accuracies as produced by other Deep learning models in literature in different color spaces are tabulated in Table 6. Of all the models in Table 6, the modified versions of LeNet5, AlexNet and VGG16 model are producing improved accuracy.

Area Under Receiver Operating Characteristic Curve (AU-ROC) analyses various color space models, RGB, YUV and HSV for classifying forest fire images using original and modified verions of LeNet5, AlexNet and VGG19. The larger areas of AU-ROC curve shown in the Fig. 9, 10 and 11 correspond to LeNet5 (original), LeNet5 (modified), AlexNet (original), AlexNet (modified), VGG16 (original), and VGG16 (modified) models respectively for the color spaces RGB, YUV and HSV.

IV. ACKNOWLEDGEMENTS

The authors would like to thank Department of Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli-627012.

REFERENCES

[1] Javier Sanchez, Maria Dolores Curt, Nicolas Robert and Jesus Fernandez, The Role of Bioenergy in the Bioeconomy-Chapter *Two Biomass Resources*; 2018. doi.org/10.1016/B978-0-12-813056-8.00002-9.

[2] Forest Survey of India, Forest Fire Activities, Uttarakhand, India, 2013. https://fsi.nic.in/forest-fire-activities.

[3] Binny Bansal, Guru Gowrappan, Shashank Kumar, Harshil Mathur, Shelly Singh and Ajay Varia, Deep Learning Algorithms in Machine Learning. Redwood City, California.

[4] Dr.Ignacio Molina, Color Computer Vision: Fundamental Application and Challenges. https://www.eso.org/sci/facilities/develop/detectors/optdet/docs/papersColour_computer_vision_ESO_2.pdf

[5] Hazim Gati Dway and Ahmed Fakhir Mutar, Study Fire Detection Based On Color Spaces, *Al-Mustansiriyah Journal of Science*. 29(4) ;2521-3520; 2018. DOI: http://doi.org/10.

[6] Vipin, V, Image processing based forest fire detection, *InternationalJournal of Emerging Technology and Advanced Engineering*. 2018; 2.2: 87-95.

[7] Sergio Saponara and Abdussalam Elhanashi, Real-timevideofire/smoke detection based on CNN in antifire surveillance systems. *Journal of Real-Time Image Processing*. https://doi.org/10.1007/s11554-020-010442020.

[8] Dai Quoc Tran, Minsoo Park, Yuntae Jeon Jinyeong Bak2 and Seunghee Park, Forest-Fire Response System Using Deep-Learning-Based Approaches with CCTV Images and Weather Data. 2022 June 24; https://doi.org/10.1109/ACCESS.2022.3184707

[9] Vipin Venugopal, Image Processing Based Forest Fire Detection. *International Journal of Advanced Research in Engineering & Technology*. 2012 Feb; 2(2).

[10] Giuseppe Marbach, Markus Loepfe and Thomas Brupbacher, "An image processing technique for fire detection in video images. 2006 Jun, 41(4) ;285-289.

[11] Zhentian Jiao, Youmin Zhang, Jing Xin, Lingxia Mu, Yingmin Yi and HanLiu et al., A deep learning based forest fire detection approach using UAV and YOLOv3. *2019 1st International conference on industrial artificial intelligence (IAI). IEEE*, 2019.

[12] Qian, Jingjing, and Haifeng Lin, A forest fire identification system based on weighted fusion algorithm. *Forests* .15.8 (2022): 1301.

[13] Fan, Ruixian, and Mingtao Pei, Lightweight forest fire detection based on deep learning. *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP). IEEE*. 2021.

[14] Zhang, Jianmei, ATT squeeze U-Net: A lightweight network for forest fire detection and recognition. *IEEE Access* 9 (2021): 10858-10870.

[15] Chayma Bahhar, Amel Ksibi, Manel Ayadi, Mona M.Jamjoom, Zahid Ullah and Ben Othman Soufience. *Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN*. 2022 Jan 2;12(1).

[16] Wang Yuanbin, Dang Langfei and Ren Jieying, Forest fire image recognition based on convolutional neural network. 2019 Sep 19.

[17] Dongqing Shen, Flame Detection Using Deep Learning. 2017 June 23.S. Gayathri, P.V. Ajay Karthi, Sourav Sunil, Prediction and Detection of Forest Fires based on Deep Learning Approach. *Journal of Pharmaceutical Negative Results.* 2022; 3(13). https://doi.org/10.47750/pnr.2022.13.S03.071.

[18] BGR Color Palette with Track Bars, Python OpenCV BGR Color Palette with Track Bars. 2023.

[19] Khan, B. Hassan, S. Khan, R. Ahmed and A. Adnan, DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection. *Mobile Information System*. 2022 Apr;2022 (pp. 5358359).

[20] Kevin Beck, and Lana Bandoim, B.S, How to Get Colored Lighter Flames. *Leaf Group Media*. 2020 Feb 24.

[21] Marc Ebner, Ernst Moritz Arndt, Color Constancy. *University at Greifswald*. https://stubber.math- inf.uni-greifswald.de/~ebner/resources/ uniG/encyclopedia-CC.pdf.

[22] Dexon, What are RGB and YUV color Spaces? 2022 Apr 05.

[23] Wildfire color, Wildfire color code, Venezia Drive, Sparks, NV, 89434. https://dexonsystems.com/blog/rgb-yuv-color-spaces.

[24] Color picker, #ff8833 hex color - Wildfire - Orange - Warm color Information, Venezia Drive, Sparks, NV, 89434.

[25] https://www.crispedge.com/faq/what-is-the-color-code-for-wildfire/.

[26] Michal Podpora and Grzegorz Pawel Korbas, YUV vs RGB - Choosing a Color Space for Human-Machine Interaction. 3; https://doi.org/10.15439/2014F206, ACSIS.

[27] Richard Szeliski, Computer vision: algorithms and applications. *Springer Nature*. 2022; 2nd edition. https://doi.org/10.1007/978-3-030-34372-9.

[28] Emmy Prema, C.S.S. Vinsley, and S.Suresh, Multi feature analysis of smoke in YUV color

space for early forest fire detection. *Fire Technology*. 52 (2016): 1319-1342.

[29] Guangfen Wei, Gang Li, Jie Zhao and Aixiang He, Development of a LeNet-5 Gas Identification CNN Structure for Electronic Noses. Sensors. 2019;19(217). https://doi.org/10.3390/s19010217.

[30] Dishashree26 Gupta, Fundamentals of Deep Learning – Activation Functions and When to Use Them? 2017 Oct; https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/.

[31] Great Learning Team, How Convolutional Neural Network Model Architectures and Applications Are Enhancing Our Daily Lives. 2020 May 11.

[32] ProjectPro, Convolutional Neural Networks Explained. The A-Z Ultimate Guide to Understanding Deep Convolutional Neural Networks. 2023 Oct 20.

[33] Zhi-Hao Zhang, Zan Yang, Yang Sun, Yang- Fan Wu, and Yi- Dan Xing, LENET-5 Convolution Neural Network With Mish Activation Function And Fixed Memory Step Gradient Descent Method. *IEEE Xplore*. 2020 May 07.

[34] Christian Garbin, Xingquan Zhu, and Oge Marques, Dropout vs. Batch normalization: an empirical study of their impact to deep learning. *Springer Nature*. 2020 Jan 20.

[35] Ernest Jeczmionek and Piotr A. Kowalski, Flattening Layer Pruning Convolutional Neural Networks. 2021 Jun 27.

[36] Kalyani Ingole and Narayani Patil, Preformance Analysis of Various Activation Function on a Shallow Neural Network. *JETIR*. 2020 Jun 7; 2349-5162.

[37] Lecun, Y, Bottou. L., Bengio. Y, Haffner. P, Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998. https://doi.org/10.1109/5.726791. S2CID 14542261.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks. 60; 6; https://doi.org/10.1145/3065386, Jun, 2017.

[39] Andrea Vedaldi, D.Phil, Representation of spatial transformations in deep neural networks. *Visual Geometry Group Department of Engineering Science University of Oxford.* 2 018.https://www.robots.ox.ac.uk/~vgg/publications/2018/Lenc18a/lenc18a.pdf.

[40] D. Crystal Jaba Kani and S. Saudia, Analysis on the Performance of Machine Learning Models for Forest Fire Prediction. Centre for Information Technology and Engineering. 2023; 978-1-6654-7467-2. https://doi.org/10.1109/ICSSIT55814.2023.10060870.

Tables & Figures

Table 1. Flame Color and Temperature

Flame Color	Temperature Range
Dark Red (First Visible glow)	500° to 600° C (900° to 1100° F)
Dull Red	600° to 800° C (1,1000° to 1650° F)
Bright Cherry Red	800° to 1000° C (1650° to 1800° F)
Orange	1000° to 1200° C (1800° to 2100° F)
Bright Yellow	500° to 600° C (900° to 1100° F)
White	1400° to 1600° C (2,500° to 2900° F)

Table 2. Hyper Parameters of LeNet5

Layer	Filters/ Neurons-Original	Filters/ Neurons-Modified	Size	Padding	Stride	Activation
CONV1	64	16	3 x 3	0	-	relu
POOL2	-	-	2 x 2	-	2,2	-
CONV3	16	8	3 x 3	0	-	relu
POOL4	-	-	2 x 2	-	2,2	-
FC5	682	304	-	-	-	relu
FC6	2	2	-	-	-	softmax

Table 3. HyperParameters of AlexNet

Layers	Filters/ Neurons-Original	Filters/ Neurons-Modified	Kernel Size	Pooling	Stride	Activation
CONV 1	96	64	11x11	same	4 x 4	relu
POOL 2	-	-	2 x 2	same	2 x 2	-
CONV 3	256	48	5 x 5	same		relu
POOL 4	-	-	2 x 2	same	2 x 2	-
CONV 5	384	32	3 x 3	same	1 x 1	relu
CONV 6	384	16	3 x 3	same	1 x 1	relu
CONV 7	256	8	3 x 3	same	1 x 1	relu
POOL 8	-	-	2 x 2	same	2 x 2	-
FC 9	1093	1093	-	-	-	relu
FC 10	1093	304	-	-	-	relu
FC11	121	121	-	-	-	relu
FC12	2	2	-	-	-	Softmax

Table 4. HyperParameters of VGG16

Layer	Filters/ Neurons-Original	Filters/ Neurons-Modified	Kernel Size	Padding	Stride	Activation
CONV 1	64	64	3 x 3	Same	-	relu
CONV 2	64	64	3 x 3	same	-	relu
POOL 3	-	-	2 x 2	-	2 x 2	-
ONV 4	128	48	3 x 3	same	-	relu
CONV 5	128	48	3 x 3	same	-	relu
POOL 6	-	-	2 x 2	-	2 x 2	-
CONV 7	256	32	3 x 3	same	-	relu
CONV 8	256	32	3 x 3	same	-	relu
CONV 9	256	32	3 x 3	same	-	relu
POOL10	-	-	2 x 2	-	2 x 2	-
CONV 11	512	16	3 x 3	same	-	relu
CONV 12	512	16	3 x 3	same	-	relu
CONV 13	512	16	3 x 3	same	-	relu
POOL 14	-	-	2 x 2	-	2 x 2	-
CONV 15	512	8	3 x 3	same	-	relu
CONV 16	512	8	3 x 3	same	-	relu
CONV 17	512	8	3 x 3	same	-	relu
POOL 18	-	-	2 x 2	-	2 x 2	-
FC 19	4096	1093	-	-	-	relu
FC 20	4096	304	-	-	-	relu
FC 21	2	2	-	-	-	softmax

Table 5. Accuracy of Lenet5, AlexNet and VGG16 Models and Their Variants in Various Color Spaces

Models	RGB	YUV	HSV
LeNet5 [Original]	0.94	0.96	0.90
LeNet5 [modified]	0.95	0.94	0.95
AlexNet [Original]	0.89	0.92	0.88
AlexNet (modified]	0.95	0.91	0.93
VGG16 [Original]	0.45	0.51	0.52
VGG16 [modified]	0.93	0.92	0.94

Table 6. Accuracy of CNN Models in Literature

Reference Paper	Model used for Fire Prediction	Color model used	Accuracy
Zhentian Jiao, Youmin Zhang, Jing Xin [13]	YOLO v3	RGB	83%
JingjingQian and Haifeng Lin[14]	YOLOv3 and EfficientDet	RGB	83%
Wang Yuanbin , Dang Langfei and Ren Jieying [18]	AlexNet with Adaptive Pooling	YCbCr	93.8%
Dongqing Shen[20]	YOLO	HSI	81%
S. Gayathri, P.V. Ajay Karthi[21]	CNN and LSTM	YCbCr	92%.

Figures

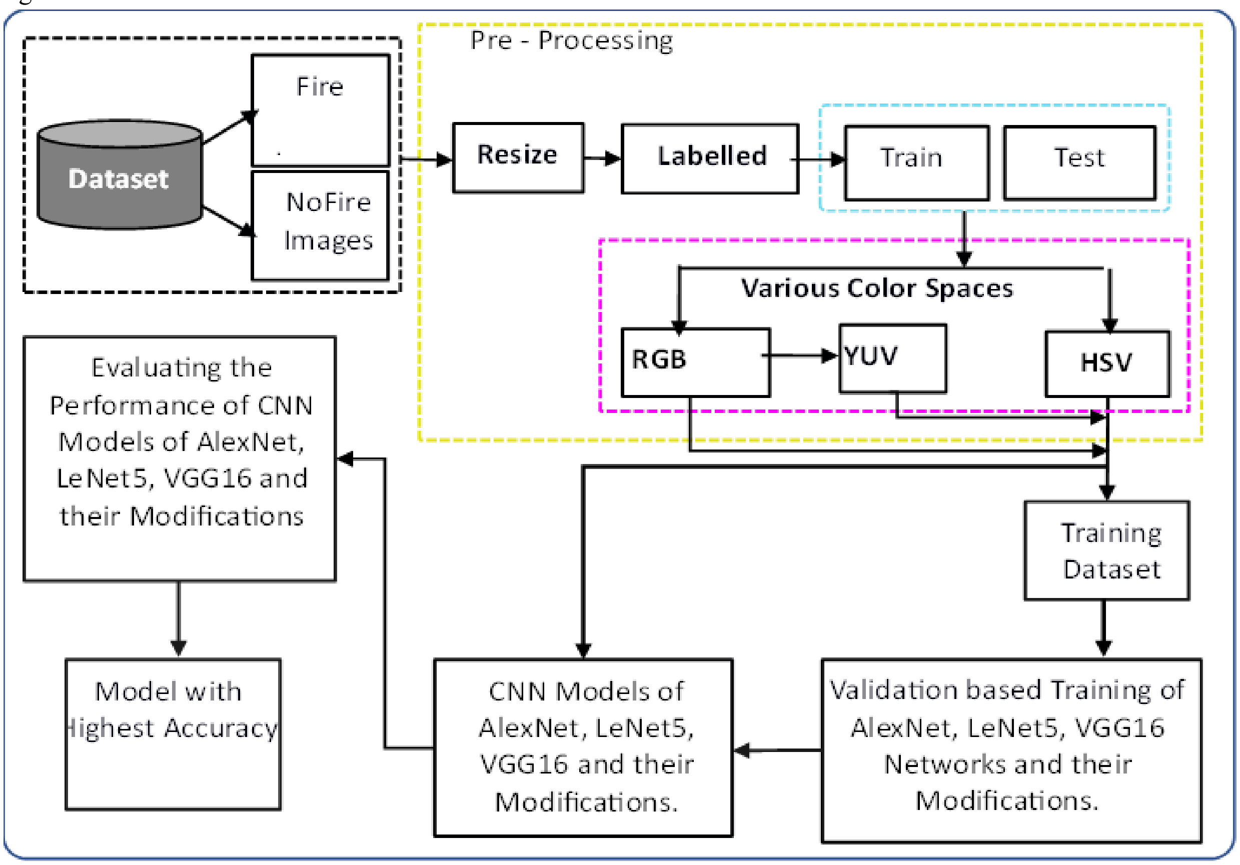
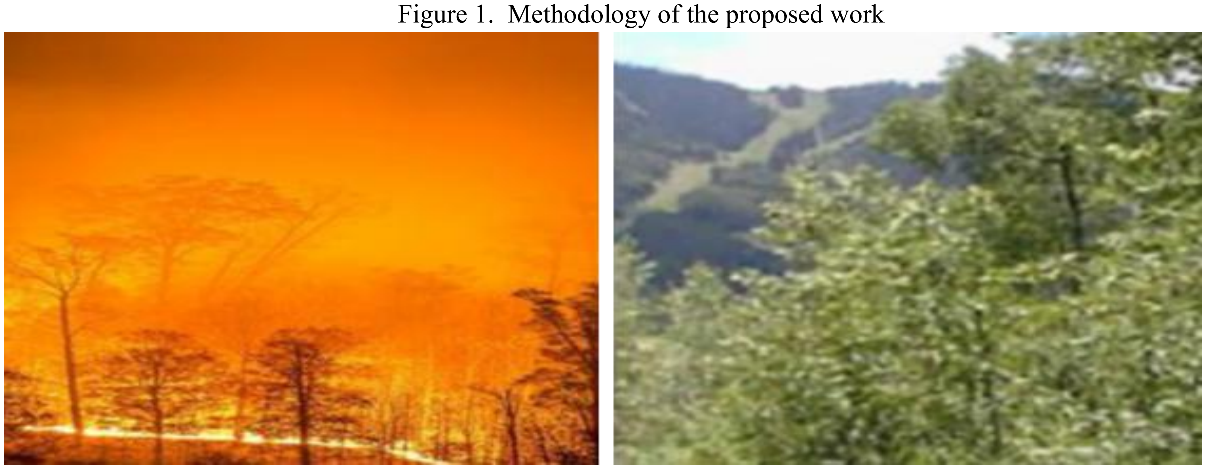
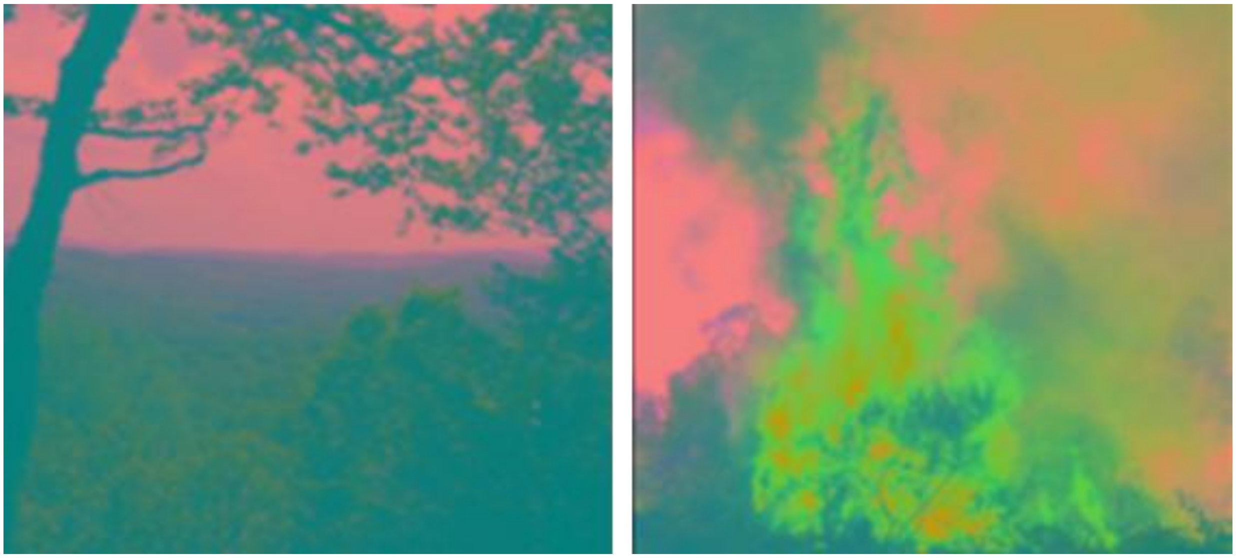


Figure 1. Methodology of the proposed work



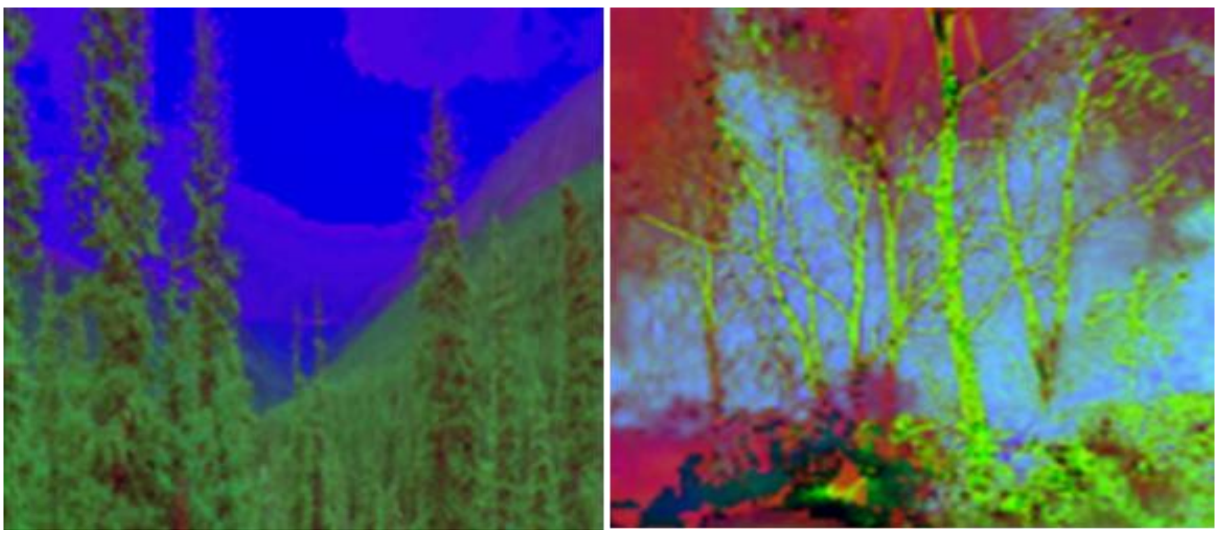
A. B.

Figure 2. A) Fire, B) NoFire images from DeepFire Dataset in RGB Color space



A. B.

Figure 3. A) NoFire, B) Fire images from DeepFire Dataset in YUV Color space



A. B.

Figure 4. A) NoFire, B) Fire images from DeepFire Dataset in HSV Color space

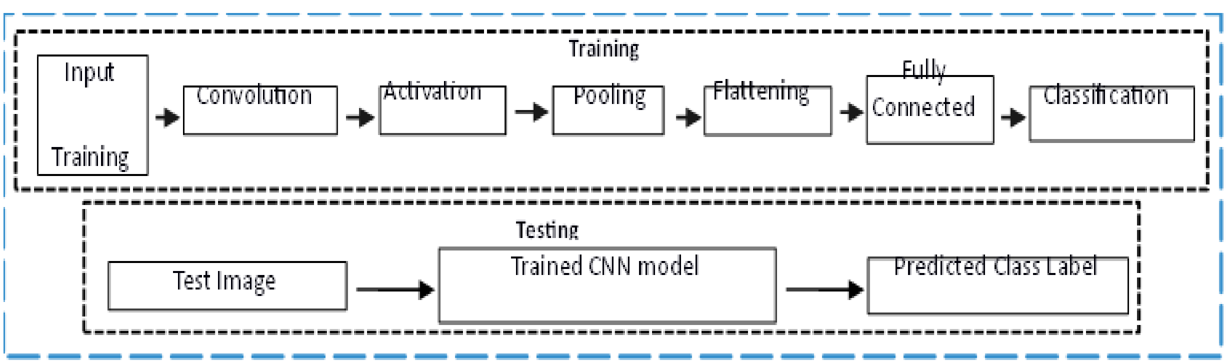


Figure 5. A General CNN Architecture

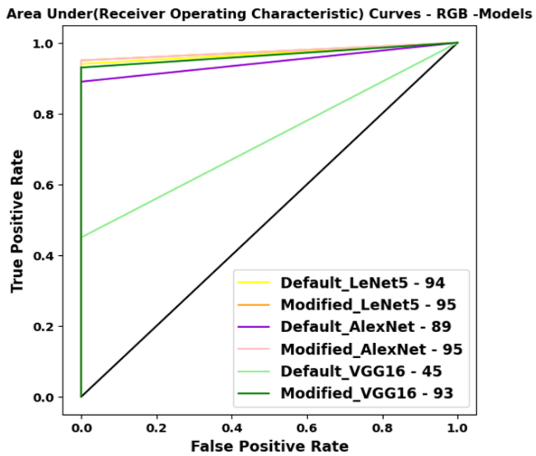


Figure 9. AU-ROC of CNN algorithms in RGB color

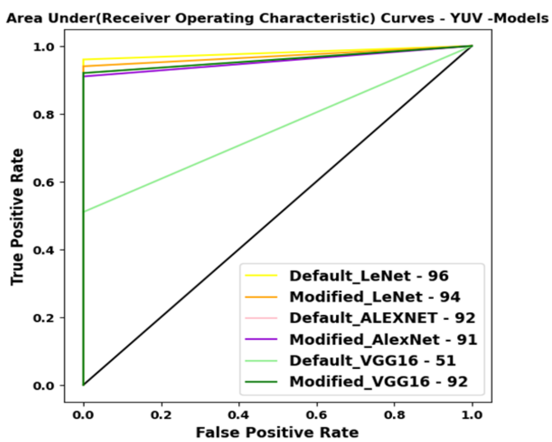


Figure10. AU-ROC of CNN algorithms in YUV color

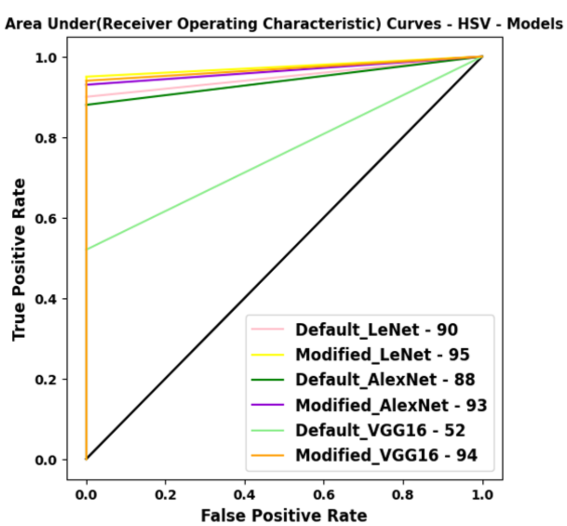


Figure11. AU-ROC of CNN algorithms in HSV color space