JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

FOREST FIRE DETECTION USING DEEP LEARNING

[1]Dr. Sreenivasa B.C., [2]M Shreya, [3]Rishita Goyal, [4]Sudhiksha R., [5]Tanya P. Bobichan

[1]Associate Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Computer Science and Engineering,
[1]Sir M Visvesvaraya Institute of Technology, Bangalore, India

Abstract: A forest fire is an unplanned fire that breaks out in a wilderness setting like a forest or prairie. Forest fires have proven to be a threat to humans and wildlife creatures. Early detection of forest fires will decrease the severity preventing huge loss of ecosystems and its effect on global conditions. The forest fire detection model that is developed can be set up to analyze and process images from security cameras, drones, and satellites. Dataset consisting of various images of forests and surroundings resembling forests is used and the images are classified into two categories: “ fire ” and “ smoke ”. To identify the existence or onset of a forest fire in an image efficiently, a deep-learning model is created and trained. In this study, we propose a forest fire detection system that makes use of YOLOv5 for detection and classification and DenseNet for feature extraction. Further the nearest authorities will soon be informed after the specifics of the incident are known.

***Keywords*:** Machine learning, deep learning, convolutional neural network, forest fire detection, object detection, YOLO, DenseNet.

I. INTRODUCTION

The uncontrollable burning of vegetation, animals, meadows, and bushes in their path is known as a forest fire. The wind significantly increases air pollution while rapidly fueling the fire. As a rule, long-lasting and highly flammable fires are caused by climate change. There are also examples of man-made wildfires, lightning strikes and extreme droughts. As a result of the tremendous loss of lives and properties caused by wildfires, this issue has spread to every nation. In addition, lung and skin illnesses in people are brought on by the carbon dioxide that forest fires emit into the atmosphere. Wildfires typically break out in India between the months of March and April when the land is covered in parched timber, dried grass, woodlands, and leaves. In most instances, the rubbing of branches can cause wildfires in hot or severely dry conditions.

Since most households rely on woods for food, fuel, and other necessities, wildfires can have an economic impact. It causes landslides and habitat loss as it ignites through grasses and small shrublands. Harmful gas pollutants and smoke from burning forests cause major health issues for people. Wildfires destroy animal habitats, spread across communities, and claim many lives because they are out of control. All flora vegetation, wildlife and soil health are destroyed by these flames. Persistent fires erupted on the borders of Himachal Pradesh with Nagaland and Manipur. Also massive wildfires erupted in Simlipal National Park in Orissa. The Indian Express said the areas of southern Chhattisgarh, central Orissa, Telangana, western Maharashtra and Andhra Pradesh are highly susceptible to wildfires.

For efficient fire control and prevention, forest fires must be identified quickly and accurately classified. We can develop a forest fire detection model that can be designed to assess and process images from security cameras, drones, and satellites in order to analyse and process images to detect forest fires. The photographs will be divided into two categories: "fire" and "smoke" using a dataset made up of diverse images of fire and smoke in forest-like settings. DenseNet will be used by the suggested system to extract features from input photographs of data from forest fires. The YOLOv5 model will then be used with the collected features to detect and classify objects, specifically detecting areas of the image that contain forest fires. A convolutional neural network (CNN) architecture called DenseNet has demonstrated higher performance in image classification tasks while enabling effective feature reuse. The object detection algorithm YOLOv5, which stands for "You Only Look Once," is capable of concurrently detecting and classifying things in real-time. To enable a quick and efficient response to forest fires, the system will generate real-time alerts and warnings for relevant authorities and stakeholders.

A reliable and effective forest fire detection system with high accuracy and real-time capabilities is anticipated as a result of the integration of DenseNet for feature extraction with YOLOv5 for detection and classification. The technology has the potential to dramatically improve efforts to manage and prevent forest fires, helping to safeguard forests, wildlife, and people's lives.

II. RELATED WORK

Aditi Kansal, *et al.,2015* [1], have proposed a system where different machine-learning techniques such as regression, SVM, neural networks, decision trees, etc. are compared. Wireless Sensor Networks. In WSNs, sensors are used to monitor specific environmental factors and transmit the resulting data to a ground station for analysis. The perfect sensor node uses little power, can collect data quickly, is reliable, inexpensive, and requires little maintenance. It also detects events, allowing for adequate and effective physical world sensing. By dividing the dataset into months, the method suggested in this research demonstrates how regression performs best for accurately identifying forest fires. This algorithm generates the result without processing the complete dataset and yields a low mean square error and large R- squared.

Medi Rahul, *et al.,2020* [2], have proposed a convolutional neural network-based picture identification technique for early forest fire detection. Fire detection methods based on image processing systems have reached their peak and have replaced many conventional methods. In this study, a technique based on transfer learning is proposed for the early detection of forest fire. Two categories may be employed to appropriately classify the majority of photos using the suggested model: fire and no fire. ResNet50 is demonstrated to be a trustworthy model from DenseNet121, ResNet50 and VGG16. ResNet50 accurately classifies majority of these photos with a high training and testing accuracy of 92 and identifies the model's input image type.

Preeti T, *et al.,2021* [3], have proposed Random Forest regression and hyperparameter tuning with the RandomizedSearchCV algorithm which uses different subsamples of datasets that fit multiple decision trees and uses averaging to enhances control overfitting and prediction accuracy. The forest fire incidents are portrayed based on the study of the models using meteorological parameters. Various forest fire prediction methods (regression techniques) such as random forest, decision tree, artificial neural network (ANN) and support vector machine algorithm are compared in this paper. These models were implemented on the Python platform. By squaring the distances between the points and the regression curve to exclude any negative signs, mean square error calculates how near a regression curve is to a set of points. Experiments are conducted to obtain a various number of training and evaluation occurrences for wildfire prediction.

A. Sheryl Oliver *et al.*,2020 [4], have proposed an approach for recognizing forest fires built on Convolution Neural Network (CNN). Numerous classification strategies have been put forth, however the models that have been suggested suffer from drawbacks that make them ineffective and unable to deliver accurate results. When compared to supervised machine learning techniques, which involve human data- training, a revolutionary convolution neural network algorithm offers great efficiency, accuracy, and relative reduced data-training stress. The approach primarily reshapes the raw dataset to meet the requirements before training the CNN model. When the trained model is given the visuals to predict, it outputs whether or not the image contains a forest fire. With a 94.3% accuracy rate, the algorithm performs well.

Rafik Ghali *et* al.,2021 [5], have proposed an approach for segmenting wildfire pixels and detecting fire areas developed on U2-Net, EfficientSeg and U-Net (deep convolutional networks). The loss functions (Binary Cross Entropy Dice loss and Dice loss) and also the data augmentation methods (rotation and horizontal flip) were utilized to train these models. The three models exhibit outstanding F1-score results and accuracy, demonstrating their dependability to partition pixels of fire and identify the specific contours of wildfire zones. EfficientSeg, U-Net, and U2-Net performed well on the CorsicanFire dataset, with F1-scores of 0.95, 0.94, and 0.92, and accuracy of 0.96, 0.98, and 0.97, respectively. According to the F1-score, which measures the effectiveness of per-pixel segmentation, EfficientSeg is the method that performs the best.

R. Shanmugapriya et al.,2019 [6], have proposed classification and detection of forest fire in satellite images. For improving the performance of feature extraction using traditional and hand-crafted algorithms which are not suited for large datasets, the use of an efficient approach of Inception-V3, CNN based, is proposed by the system for training the satellite images and for improving the accuracy in classifying the images dataset into 'non-fire' and 'fire' images. Inception-V3 framework is implemented for extracting features of datasets containing fire, also Local Binary pattern is applied to mark the locations showing the presence of fire and apply bounding box in the fire occurred region.

João Alves et al.,2019 [7], have proposed a system for automatic detection of forest fire in early stages. This system works by processing or classifying the images of the forest environment for having the presence of smoke or flame using Deep Convolutional Neural Network Inception-V3 which extracts the descriptors. A Machine-Learning Classifier is trained with the use of descriptors obtained which are also applied to the supervised learning model LR. Computational Vision technique applications are also used by the proposed system to spot the area under ignition to give information about the size of the area affected. The system aims for detection of forest fire during both night and day and also in various different scenarios of the forest.

M. Shreya et al.,2022 [8] have proposed different algorithms compare the data to identify the best performing model. A front-end website responds to a set of inputs from a user with the likelihood of a forest fire will occur. The presence of vegetation fuels the fire, and with each second the fire grows more intense and more water is needed to extinguish it. This WebUI of his takes specific traits from user input and generates an output showing the likelihood of wildfires occurring under certain input conditions . Both data related to forest fires are gathered from the UCI machine learning repository from Portugal and Algeria, respectively. This indicates that the model is trained using 80% of the data and 20% is left over for testing the trained model.

Z. Jiao et al.,2020 [9] have proposed a forest-fire detection algorithm which is applied to platforms based on UAV. Images captured by the UAV are transmitted to the ground station, where using the proposed YOLOv3 algorithm we can process the images in real-time with great accuracy on a powerful computer on the ground station. The tensor magnitude is transformed in forward propagation by altering the convolution filter's step size. The predicted bounding box is formed by the coordinates of the center after the coordinates of the point that serves as the cell's center are calculated. For ensuring detection speed and accuracy, the platform combines the maneuverability of a UAV with the powerful computational capacity of a ground station.

Y. Chen *et al*.,2019 [10] have proposed that the photograph is made clearer by enlarging the grey stage with a large variety of pixels and compressing the grey stage with a tiny variety of pixels, which increases the dynamic range of the pixel cost and improves the assessment and alternate of the grey tone. The community common is primarily based totally at the history of the photograph or the alternate in grey of the goal component is non-stop and slow, even as the particle noise reasons a unexpected alternate withinside the grey stage of a few pixels at the photograph. The concept is to apply the common cost of the grey stage of the factor and its friends withinside the photograph as opposed to the grey cost of the factor, that may produce a "smooth" impact at the factor wherein the brightness is abrupt.

III. METHODOLOGY

The System design mainly consists of:

3.1 Image Collection

3.2 Image Preprocessing

3.3 Image Segmentation

3.4 Feature Selection and Extraction

3.5 Training

3.6 Classification

3.1 Image Collection

The dataset we utilised for this project is accessible to everyone online. Three datasets—training, validation, and test are created from the data. The training set is used to train the model. The validation set is used to fine-tune model hyperparameters and assess model performance during training, while the test set is used to assess the final performance of the trained model on unobserved data.

We must annotate photos for object identification models in order to train the YOLO model. Our dataset needs to be properly annotated; this involves labeling the data to reflect the outcome you want your machine learning model to forecast. You are marking a dataset with the properties you want your machine learning system to learn to recognise, whether by labeling, tagging, transcribing, or processing it. There are various annotating formats available. A bounding box technique is applied here. Over the images that are present in our dataset, it draws a rectangular area.

3.2 Image Preprocessing

Pre-processing aims to improve image data by reducing undesirable distortions and enhancing some aspects that are crucial for further image processing. Three main processes are involved in image pre-processing: 1) Grayscale conversion 2) Noise removal 3) Image enhancement

3.2.1 Grayscale conversion: Only brightness information is contained in a grayscale image. In a grayscale image, each pixel value represents a certain amount or quantity of light. In grayscale images, the brightness graduation can be distinguished. Only light intensity is measured in a grayscale image. The brightness of the 8-bit image will range from 0 to 255, where 0 denotes black and 255 denotes white. A colour image is turned into a grayscale image during the grayscale conversion process. Compared to coloured photographs, grayscale images are simpler and quicker to process. Grayscale images are used to apply all image processing algorithms.

3.2.2 Noise Removal: Digital photos with undesirable noise should be identified and removed using noise removal. It might be challenging to distinguish between noise-induced features and the real or actual picture features. Random fluctuations in pixel values is noise.

3.2.3 Image Enhancement: Image processing is done with the intention of making features of interest more visible. Here, contrast enhancement is used to produce results of higher caliber or better quality.

3.3 Image Segmentation

Image segmentation is frequently used to identify objects and boundaries in images (such as lines, curves, etc.). Image segmentation, in more exact terms, is the process of giving each pixel in an image a label so that pixels with the same label have specific properties. Image segmentation is carried out automatically as part of the network architecture of the YOLO method. In other words, the network learns to anticipate the bounding box and class of each object within each grid cell and to divide the image into various grids.

3.4 Feature Selection and Extraction:

A model's performance can be improved by minimising the amount of unnecessary or redundant features, which can result in overfitting. Feature selection is the process of automatically or manually choosing the subset of most acceptable and relevant characteristics to be utilised in model development.

The next step would be to extract features from the images using pre-trained convolutional neural networks (CNNs) such as DenseNet. Feature extraction may involve computing a set of features such as texture, shape, and color of the image.

Convolutional, RELU, pooling, and fully connected normalised layers are among the many hidden layers that make up a CNN, a kind of Deep Neural Network (DNN). In order to improve network performance and reduce memory usage, CNN shares weights at the convolutional layer. The 3D volumes of the neurons, local connection, and shared weights are the key characteristics of CNN. A convolution layer creates a feature map by combining various input picture sub - regions with a learnt kernel. The ReLu layer is then used to apply a non-linear activation function in order to enhance the convergence properties when the error is small. To reduce a 2x2 or 3x3 grid to a single scalar value, a portion of the image or feature map is selected for the pooling layer, and the pixel with the highest value among them or the average value is chosen as the representative pixel. As a result, the sample size is significantly reduced. In the output stage, convolutional layers may occasionally be combined with a conventional Fully-Connected (FC) layer.

DenseNet, which consists of densely connected convolutional networks, was created specifically to increase accuracy in high-level neural networks caused by the vanishing gradient, which occurs when information travels a long distance between input and output layers before disappearing. Each layer in DenseNet receives extra inputs from all levels that came before it and transmits its own feature-maps to all layers that came after it. If there are L levels, then there will be L connections between them. However, there will only be about L(L+1)/2 connections in a DenseNet. Since a dense net has fewer layers than the other models, we can easily train a model with more than 100 layers here. As we delve further into the network, this kind of becomes unsustainable. Let's assume that there are ten levels total. Then, we will input all of the feature maps from the previous nine layers on the tenth layer. There will be an explosion of feature maps if each of these levels produces 128 feature maps. This layer is similar to one by one convolution followed by Max pooling to reduce the size of the feature maps. To solve this issue, we create a dense block here, each of which contains a pre-specified number of layers. The output from each dense block is then given to the transition layer.

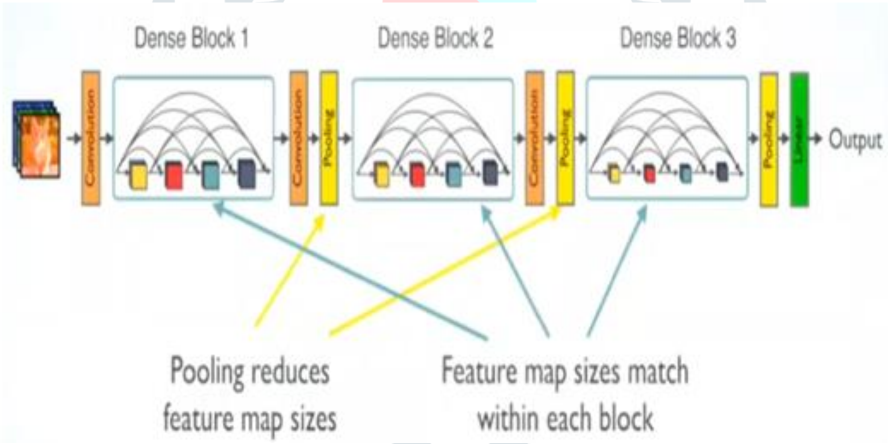

Figure.01.DenseNet Network Architecture

Source: Adapted from [21]

3.5 Training

The preprocessed and feature-extracted dataset must be used to train the deep learning model, and a separate testing dataset must be used to evaluate its performance. The model should be trained to recognise the many patterns and features connected to smoke and forest fires.

We are utilising the You Only Look Once (YOLO) algorithm in our suggested system. This algorithm employs convolutional neural networks (CNN) to detect objects in real-time and achieve high accuracy. The technique "only looks once" at the image, as the name implies, which means that only one forward propagation through a neural network is necessary to identify objects. This indicates that a single algorithm run is used to perform prediction throughout the full image. The model is trained using both the photos and the accompanying label file including bounding box coordinates and class names for the items in the images. As a result, training for regression and classification both take place here. A single neural network is applied to the entire image by the algorithm, which then divides it into regions and forecasts bounding boxes and probabilities for each region. The projected probabilities are used to weight these bounding boxes. Our approach makes use of YOLOv5, which yields effective results for object recognition and categorization.

With the well-known PyTorch library, we are utilising YOLO. YOLOv5 returns detections in PyTorch output formats and accepts PyTorch inputs. An open-source deep learning toolkit called PyTorch is based on the well-known Torch library. It is also a Python-based library that is more frequently used for computer vision and natural language processing.

Since YOLO trains on complete images, it directly optimises detection performance. During training and testing, it sees the complete image, so it implicitly encodes contextual information about classes in addition to their appearance. Multiple objects in an image will be recognised by the real-time recognition system. CSPDarknet-53 is the YOLOv5's supporting structure. Darknet is an open-source neural network framework written in C and CUDA. It is used to train new YOLO models and run the existing ones. This model is employed for object detection after training. We receive the detection coordinates and a confidence score from YOLO, which we utilise as an input for OpenCV to construct bounding boxes around the detections and count them. Every frame is counted for detections, and the procedure is repeated.

The "obj.names" file, which includes the names of the classes the model sought to identify, is first created. The "obj.data" file, which is saved in the backup folder, also contains the "obj.names" and weights path, as well as the train data directory and a number of classes. Finally, there are two classes in a cfg file. The YOLOv5 model is then given the image. This will search the image for any coordinates that are visible. It creates a grid out of the image and uses that grid to analyse the features of the target object. Here, data is used for training to an extent of 80% and for validation to an extent of 20%. The YOLOv5 weights that were trained on the dataset are now stored in a file. Now that these weights have been taught, we can detect fire or smoke.

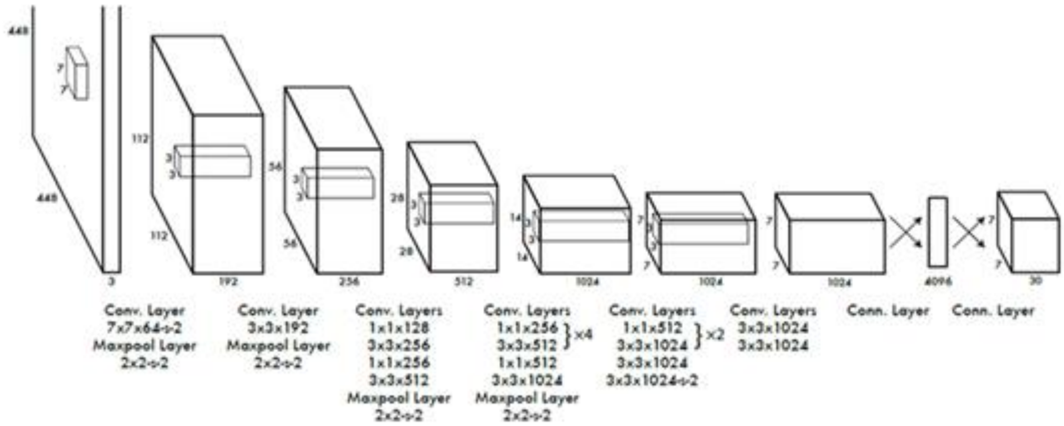

Fig 02. Basic YOLO Architecture

Source: Adapted from [22]

3.6 Classification

The YOLO network can be used to categorise items in fresh photos once it has been trained. After analysing the input image, the network makes predictions about the type and location of each object in each grid cell. The trained model in our proposed system divides the computer vision input into the classes of fire and smoke. If fire or smoke is seen, a message can be sent to the appropriate authorities utilising Telegram's facilities after being classified. Telepot is utilised in this action. You can create applications using the Telegram Bot API using Telepot. Both Python 2.7 and Python 3 support it. It also includes an async version based on asyncio for Python 3.5 and later. The bot is being developed here for the purpose of implementation for a particular Telegram account, and it can later be used in real-time by the relevant authorities.

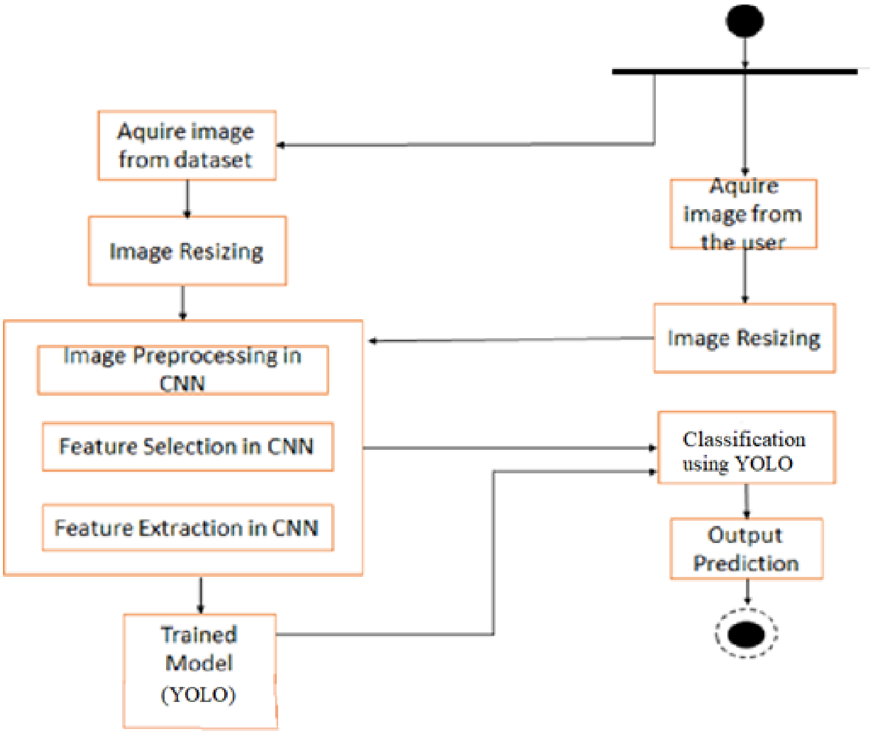

Fig 03. Activity diagram for forest fire detection system

Source: Primary

IV. RESULTS

The proposed system, Forest Fire Detection, using DenseNet and YOLO detects fire or smoke and sends the corresponding message so that necessary actions can be taken to control early forest fires and thereby reducing the damage it causes. The proposed model gives 94% accuracy in detecting the fire or smoke.

V. CONCLUSION

The proposed system has a lot of potential, opportunities, and benefits including playing a crucial role in the protection of a healthy ecosystem that could put wildlife in danger, compromise air quality, and jeopardize the safety of all communities. Extensive testing of interactions of models on the datasets are performed to get the best models for final system integration. The proposed system mainly aims to create an effective and efficient algorithm for detection and prevention of wildfire by promptly alerting the responsible authorities.

VI. REFERENCES

1. A. Kansal, Y. Singh, N. Kumar and V. Mohindru, "Detection of forest fires using machine learning technique: A perspective," 2015 Third International Conference on Image Information Processing (ICIIP), 2015, pp. 241-245, doi: 10.1109/ICIIP.2015.7414773.
2. M. Rahul, K. Shiva Saketh, A. Sanjeet and N. Srinivas Naik, "Early Detection of Forest Fire using Deep Learning," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 1136-1140, doi: 10.1109/TENCON50793.2020.9293722.
3. T. Preeti, S. Kanakaraddi, A. Beelagi, S. Malagi and A. Sudi, "Forest Fire Prediction Using Machine Learning Techniques," 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498448.
4. A. S. Oliver, U. Ashwanthika and R. Aswitha, "Detection of Forest Fire using Convolutional Neural Networks," 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020, pp. 1-6, doi: 10.1109/ICSSS49621.2020.9202200.
5. R. Ghali, M. A. Akhloufi, M. Jmal, W. S. Mseddi and R. Attia, "Forest Fires Segmentation using Deep Convolutional Neural Networks," 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021, pp. 2109- 2114, doi: 10.1109/SMC52423.2021.9658905.
6. R. S. priya and K. Vani, "Deep Learning Based Forest Fire Classification and Detection in Satellite Images," 2019 11th International Conference on Advanced Computing (ICoAC), 2019, pp. 61-65, doi: 10.1109/ICoAC48765.2019.246817.

7. Alves, J., Soares, C., Torres, J.M., Sobral, P., Moreira, R.S. (2019). Automatic Forest Fire Detection Based on a Machine Learning and Image Analysis Pipeline. In: Rocha, Á., Adeli, H., Reis, L., Costanzo, S. (eds) New Knowledge in Information Systems and Technologies. WorldCIST'19 2019. Advances in Intelligent Systems and Computing, vol 931. Springer, Cham. doi.org/10.1007/978-3-030-16184- 2_24
8. Shreya, M., Rai, R., Shukla, S. (2023). Forest Fire Prediction Using Machine Learning and Deep Learning Techniques. In: Smys, S., Lafata, P., Palanisamy, R., Kamel, K.A. (eds) Computer Networks and Inventive Communication Technologies. Lecture Notes on Data Engineering and Communications Technologies, vol 141. Springer, Singapore. doi.org/10.1007/978-981-19-3035-5_51
9. Z. Jiao et al., “A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection,” 2020 Chinese Control and Decision Conference (CCDC), 2020, pp. 4963-4967, doi: 10.1109/CCDC49329.2020.9163816.
10. Y. Chen et al., “UAV Image-based Forest Fire Detection Approach Using Convolutional Neural Network,” 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2019, pp. 2118-2123, doi:10.1109/ICIEA.2019.883395
11. Gabriel D. Caffaratti, Martín G. Marchetta, Leonardo D. Euillades, Pablo A. Euillades, Raymundo Q. Forradellas, Improving Forest detection with machine learning in remote sensing data, Remote Sensing Applications: Society and Environment, Volume 24, 2021, 100654, ISSN 2352-9385 doi: dx.doi.org/10.1016/j.rsase.2021.100654
12. J. Zhang, H. Zhu, P. Wang and X. Ling, “ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition,” in IEEE Access, vol. 9, pp. 10858-10870,2021, doi: 10.1109/ACCESS.2021.3050628.
13. B. Arteaga, M. Diaz and M. Jojoa, “Deep Learning Applied to Forest Fire Detection,” 2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2020, pp. 1-6, doi: 10.1109/ISSPIT51521.2020.9408859.
14. V. Khryashchev and R. Larionov, “Wildfire Segmentation on Satellite Images using Deep Learning,” 2020 Moscow Workshop on Electronic and Networking Technologies (MWENT), 2020, pp. 1-5, doi:10.1109/MWENT47943.2020.9067475.
15. M. D. Molovtsev and I. S. Sineva, “Classification Algorithms Analysis in the Forest Fire Detection Problem,” 2019 International Conference “Quality Management, Transport and Information Security, Information Technologies” (IT&QM&IS), 2019, pp. 548-553, doi:10.1109/ITQMIS.2019.8928398.
16. N. Kumar and A. Kumar, “Australian Bushfire Detection Using Machine Learning and Neural Networks,” 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020, pp. 1-7, doi: 10.1109/ICSSS49621.2020.9202238.
17. S. Wu and L. Zhang, “Using Popular Object Detection Methods for Real Time Forest Fire Detection,” 2018 11th International Symposium on Computational Intelligence and Design (ISCID), 2018, pp. 280-284, doi:10.1109/ISCID.2018.00070.
18. Y. Chen et al., “UAV Image-based Forest Fire Detection Approach Using Convolutional Neural Network,” 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2019, pp. 2118-2123, doi: 10.1109/ICIEA.2019.8833958.
19. Y. Cao, F. Yang, Q. Tang and X. Lu, “An Attention Enhanced Bidirectional LSTM for Early Forest Fire Smoke Recognition,” in IEEE Access, vol. 7, pp. 154732-154742, 2019, doi: 10.1109/ACCESS.2019.2946712.
20. . Dutta and S. Ghosh, “Forest Fire Detection Using Combined Architecture of Separable Convolution and Image Processing,” 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), 2021, pp. 36-41, doi:10.1109/CAIDA51941.2021.9425170.
21. Sik-Ho-Tsang, Multiple Dense Blocks (Nov.25,2018), https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803
22. N S Artamonov, Pavel Yakimov, Architecture of YOLO CNN (Sept 2018), DOI: 10.1088/1742-6596/1096/1/012086