iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 Issue: XI Month of publication: November 2022

DOI: https://doi.org/10.22214/ijraset.2022.47749

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

Survey on Fire Detection Using Image Processing

Shrikant Dandge[1], Saba Syed[2], Gunjan Sharma[3], Tanay Zope[4], M. M. Phadtare[5]
[1, 2, 3, 4]Department of Computer Engineering, AISSMS College of Engineering, Pune, India
[5]Assistant Professor, Department of Computer Engineering, AISSMS College of Engineering, Pune, India

***Abstract:** With rising Urbanisation the frequency of fires has increased. A rapid need exists for quick and effective fire detection. Traditional fire detection systems are utilizing physical sensors to detect fire. Sensors gather information about the chemical characteristics of airborne particles, which traditional fire detection systems then use to generate an alarm. However, it can also result in false alerts; for instance, an ordinary fire alarm system might be triggered by smoking inside a space. Using a computer system based on vision for detecting fire would facilitate rapid and precise detection of fire with the ongoing developments in image processing. A lot of observable improvements have been developed to help a successful fire detection algorithm or model. This paper compiles research on methods that, when used, can effectively detect fire. In addition, a system architecture for fire detection is developed in this study. It suggests many fire detection methods, including Celik, SDD, F-RCNN, R-FCN and YOLOv3. This paper offers a thorough comparison of the same.*

***Keywords:** Fire Detection, Image Processing, Accuracy of Models, Convolutional Neural Network, Deep Learning.*

I. INTRODUCTION

Because of anthropogenic reasons and a arid climate, the number of reported fire cases in forests has risen yearly. Among the most crucial elements of surveillance systems used to keep an eye on structures and the surroundings are systems that detect fire. It is preferable for the system to be able to report a fire at its earliest stage as part of an early warning process. A variety of detecting systems have been extensively explored and put to use in order to prevent the horrible calamity of fire. for fire detection, two major sorts of approaches can be distinguished: 1) Conventional fire alarms, and 2) fire detection supplemented by vision sensors.

Traditional fire detection systems are built around proximity-activated sensors like optical and infrared ones. These are not suitable for critical situations, and in the event of an alarm, human intervention is required and visit to the fire’s location is required. Additionally, such systems typically are unable to offer details like the size, location, and level of the fire’s burning. Researchers in this subject have looked into a variety of optical sensor-based technologies to get beyond these constraints; these systems have the advantages of requiring less human involvement, quicker response times, lower costs, and wider surveillance coverage. These can also confirm the presence of fire without the need for a person to travel to the scene and can provide comprehensive information on the fire, such as its degree, dimensions, etc. Other than the Benefits, these systems still have several drawbacks, such as complicated observational settings, unreliable illumination, and inferior frames. Researchers have made many attempts to overcome these concerns by taking into account both colour and motion attributes. Although this method is more accurate than conventional ones. there are still high cases of false alarms, the precision of the system can still be increased.

A. Need for fire Detection

The fourth disruptive risk identified by the survey participants for 2021 is FIRE. In India, there were 9,329 documented fire accidents with over 9,000 fatalities in 2020, which is a significant source of worry for businesses [1].

Rapid urbanisation and a lack of fire safety standards and guidelines are two major contributors to the increase in fire accidents. 58% of fire-related fatalities in residential structures in 2019 compared to 2% in factories, according to NCRB data. THREAT MAPPING: A total of 9,329 instances of fire accidents were reported in 2020, with residential buildings reporting roughly 58% of all fatalities. The Supreme Court ordered all the states to conduct fire safety audits of designated Covid-19 hospitals in 2021 as a result of the at least 15 known incidents of fire-related accidents at covid hospitals in that year. According to the data from the Mumbai fire brigade, a total of 324 fires were reported in high-rise buildings in Mumbai between January 2020 and October 2021, and 127 of those buildings, or 39.2%, lacked a working firefighting system. This system was functional and used to use to combat fires in the remaining 197 buildings. Rapid urbanisation and densely inhabited urban clusters increase the risk of fire accidents because fires spread quickly in densely populated areas. Therefore, Prompt and effective fire detection is critical.

II. METHODOLOGY

A. *Pre-processing of Data*

Steps in Data Pre-processing for Machine Learning

1) *Obtain the Dataset*

The websites for dataset hunting are:

a) Kaggle is quite well-organized [42].
b) Reddit, where users can ask for datasets for fire detection.
c) Google Dataset Search, etc.

If we wanted to be autonomous, we could design our own dataset, start with a few hundred lines, and add the rest as we went.[43][41]

A Python library by the name of Beautiful Soup exists [11]. It is a library for extracting information from XML and HTML files

2) *Import all of the required libraries*

The primary Python libraries utilized in this machine learning data pre-treatment & picture processing are:

a) NumPy (performing scientific calculations)
b) Pandas (toolkit for handling and analysing data)
c) Matplotlib (2D charting tool)
d) Scikit-Image (open-source image processing Python libraries)
e) OpenCV (for computer vision and image processing tasks for a variety of applications is OpenCV)

3) *Load the Dataset*

During this step, the dataset(s) obtained for the current ML project must be imported.

a) Load Data with Python Standard Library [11].
b) To load machine learning data in Python, you may alternatively use NumPy and the numpy.loadtxt() function.
c) The third approach for importing the machine learning data uses the pandas.read_csv() function from the Pandas library.

4) *Pre-processing*

a) Grayscale Conversion

Grayscale is just the process of turning coloured images into black and white.[2].

grayimage = skimage.color.rgb2gray(image)
plt.imshow(grayimage, cmap = 'gray')

b) Data Augmentation

Method for increasing a dataset. Standard data augmentation methods include flipping data horizontally and vertically, rotating data, cropping data, shearing data, etc. [11]

5) *Splitting the dataset*

Before a dataset is utilized in a machine learning model, it must be split into Testing Data->Unseen Data & Training Data->Seen Data.

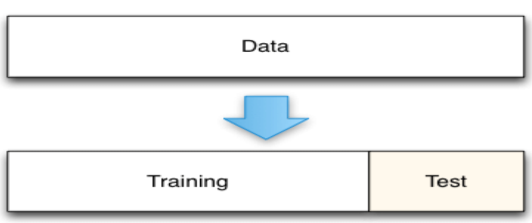


Fig 2.1 Splitting of Test and Train Dataset

The training set is used for training a machine learning model. Usually, the dataset is split into 70:30 or 80:20 ratios.[11][10].

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

6) Data Normalization

It is a method for normalizing the independent variables in a dataset within a specified range. Scaling of features limits the range of variables.

a) Standardization

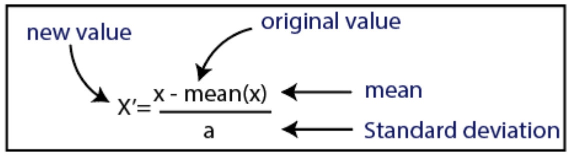


Fig 2.2 Formula for Standardization

b) Normalization

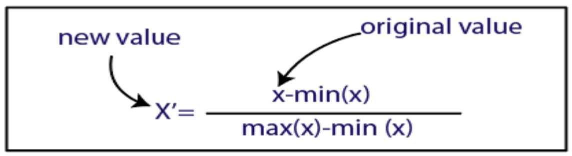


Fig 2.3 Formula for Normalization

B. Feature/Attribute Selection

The performance of your model is significantly impacted by feature selection, one of the fundamental ideas in machine learning. The data properties you use to train your machine-learning models have a considerable influence on the performance you can achieve.[23][26].

1) *Filter Method:* The characteristics are chosen, and then the model is constructed. Here, the correlation matrix is used for filtering, and VIF and Pearson correlation are typically used.
2) *Wrapper Method:* A machine learning algorithm is required by a wrapper method, which uses its performance as a criteria for evaluation.

a) *Step Forward Selection:* In each iteration, the feature that best improves our model is added, and this process continues until adding a new variable has no effect on the model's performance.

b) *Backward Elimination:* Backward elimination helps the model to perform better by starting with all the features and removing the least important one at a time.

3) *Embedded Method:* Embedded methods are iterative in that they handle each round of the model training process and carefully separate out the elements that are most helpful in training for that round.

C. Training and Model Building

The prepared data is fed into your machine learning model, which then uses the data to detect patterns and make predictions.[15]. Training can be done using various image processing algorithms [22][25].

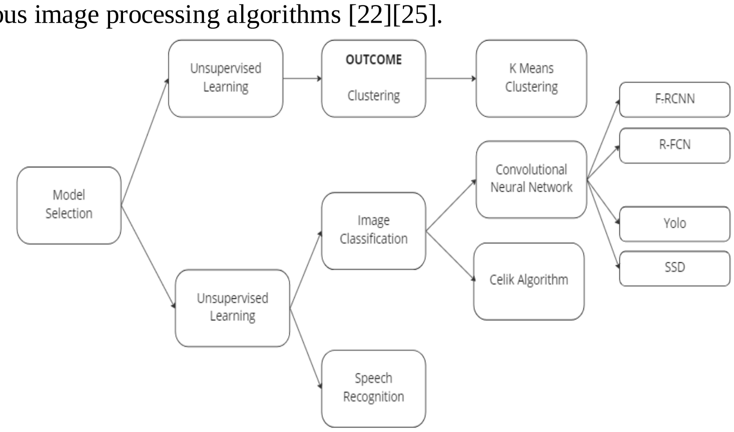


Fig. 2.4 Categorization of Algorithms

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

1) Yolo Only Look Once (YOLO)

For detection of an object, the YOLO machine learning technique uses features obtained by a deep convolutional neural network. [31].

How does YOLOv3 work?

The entire image is applied to a single neural network, which separates it into several regions. Predictions are then made using bounding boxes and probabilities depending on each region.

Fig. 2.5 Working of yolo

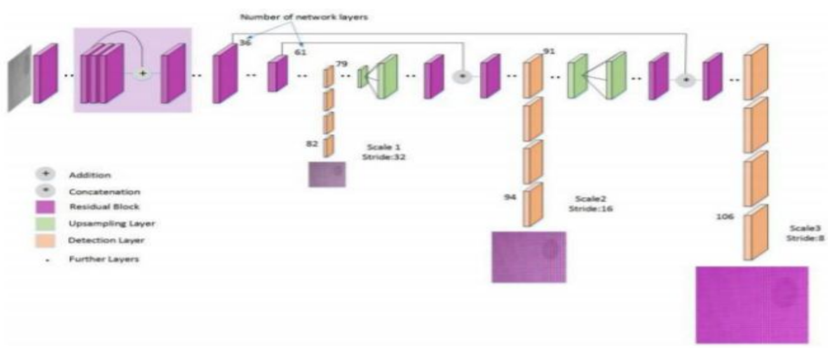


Fig. 2.6 Architecture of YOLO

The third version of YOLO uses a 53-layered convolutional neural network architecture for feature extraction, with consecutive 1 x 1 and 3 x 3 convolutional layers. The YOLO v3 convolutional architecture now has 106 layers after 53 more layers are added for detection reasons. [18][14].

2) Celik Algorithm

The algorithm may be used in combination with conventional fire detection systems to minimize false alarms.[2].

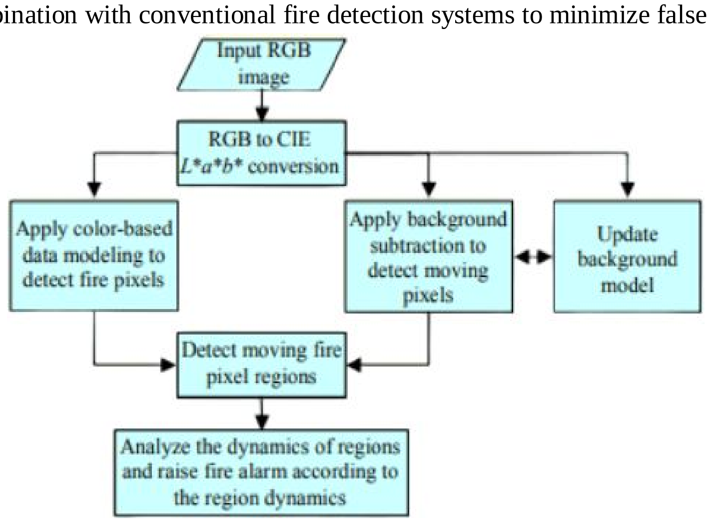


Fig.2.7 Celik Algorithm

3) R-FCN (Region - Fully Convolutional Network)

R-FCNs, also known as region-based fully convolutional networks, are a kind of region-based object detector. R-FCN is fully convolutional with almost every calculation shared on the actual situation, unlike initial object detectors (region based) like Fast/Faster R-CNN which use an expensive per-region subnetwork multiple of times.[38].

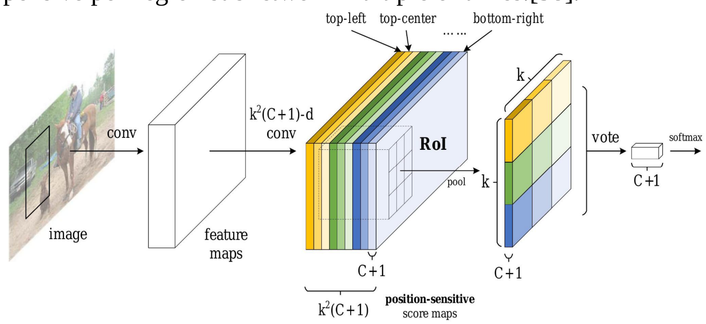


Fig. 2.8 R-FCN Architecture

4) Faster Region based Convolutional neural network (F-RCNN)

An expansion of Fast R-CNN is Faster R-CNN. Faster R-CNN performs calculations for all proposals (i.e., ROIs) at once as opposed to performing individual calculations for each proposal.

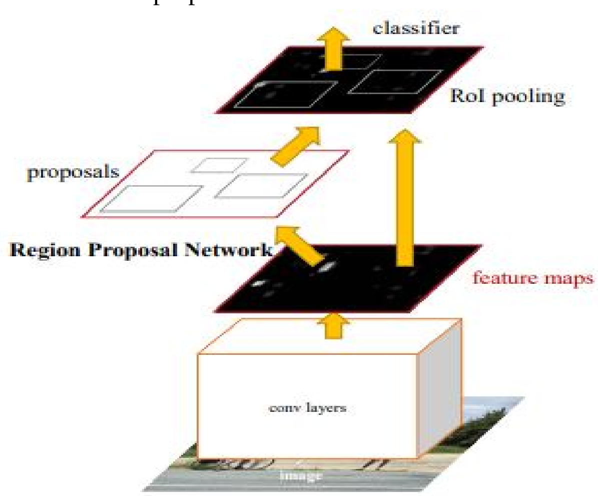


Fig. 2.9 F-RCNN Architecture

5) Single-Shot Detector (SSD)

While using a multi-box to identify numerous things present in an image, single-shot detectors like YOLO only take one shot. It is a very accurate object detection technique that is much faster.[37][20].

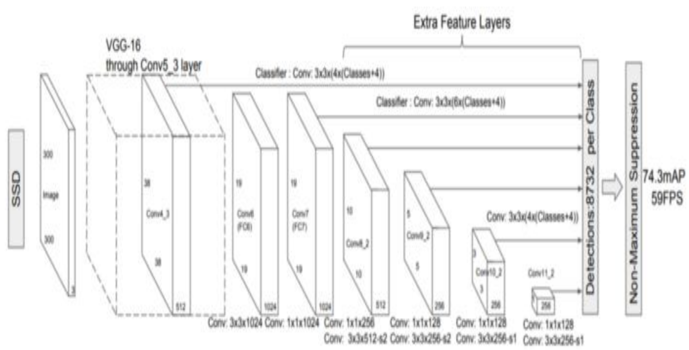


Fig. 2.10 SSD Architecture

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

D. Prediction of Model

Use the logistic regression method for predicting binary (0/1) outcomes; the decision tree classifiers, random forests classification algorithm, or KNN to predict multi-class (1 or 2 or 3...) outcomes; and the linear regression technique to predict continuous (0/1) outcomes. It is advised to use artificial neural network (ANN) approaches for the analysis of other unstructured data types (image/voice), such as speech recognition using Recurrent Neural Networks (RNN) and Natural Language Processing (NLP) and image identification using Convolutional Neural Networks (CNN).

E. Evaluation of Model

The model's performance is evaluated using data that has never been seen before [23].
Testing Data -> Unseen Data
Training Data -> Seen Data

1) Confusion Matrix

It is only a matrix-format representation of the mentioned parameters [24].

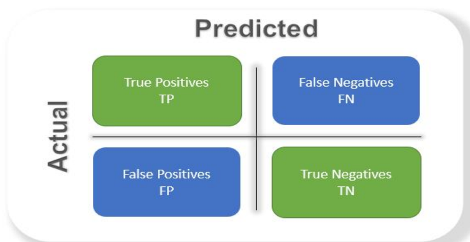


Fig. 2.11 Confusion Matrix

True positives (TP): Results that are both predicted and actual positive.
False positives (FP): are predictions of positive results that turn out to be negative.
True negatives (TN): Results that are both predicted and actual negative.
False negatives (FN): Predicted negative results that turn out to be positive.

2) Precision

Precision Percentage of cases that are positive compared to all predicted instances that are positive. "How much the model is accurate when it says it is right" is the goal [40][33].

$$\frac{TP}{TP + FP}$$

Fig. 2.12 Precision Formula

3) Recall

The percentage of incidents that are positive out of all instances that are actually positive. "How many more correct ones, the model missed when it presented the right ones." [26][24]

$$\frac{TP}{TP + FN}$$

Fig. 2.13 Recall

4) F1 score

It is the precision and recall harmonic mean. Greater The better because both contribute to this[23][17]

$$\frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall}$$

Fig. 2.14 F1 Score

5) ROC (Receiver Operating Characteristics)

A graph representing the performance of a classification overall classification thresholds is described as a ROC curve (receiver operating characteristic curve).[26].

$$\text{TPR (sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR (1-specificity)} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

Fig. 2.15 ROC (Receiver Operating Characteristics)

III. DISCUSSIONS

This paper summarizes the most popular algorithms applicable to fire detection. Below is the comparison table between the different fire detection algorithms with respect to the given parameter [49].

TABLE I
COMPARISON BETWEEN DIFFERENT ALGORITHMS

Sr. no	Algorithm	Missed Detection Rate(%)	False Alarm Rate(%)	Accuracy(%)
1	F-RCNN	29.41	0	83.87
2	R-FCN	0.018	0.69	99.43
3	SSD	0.036	0.97	99.20
4	Celik	0	1.29	98.93
5	YOLO v3	0	0.46	99.62

IV. CONCLUSION

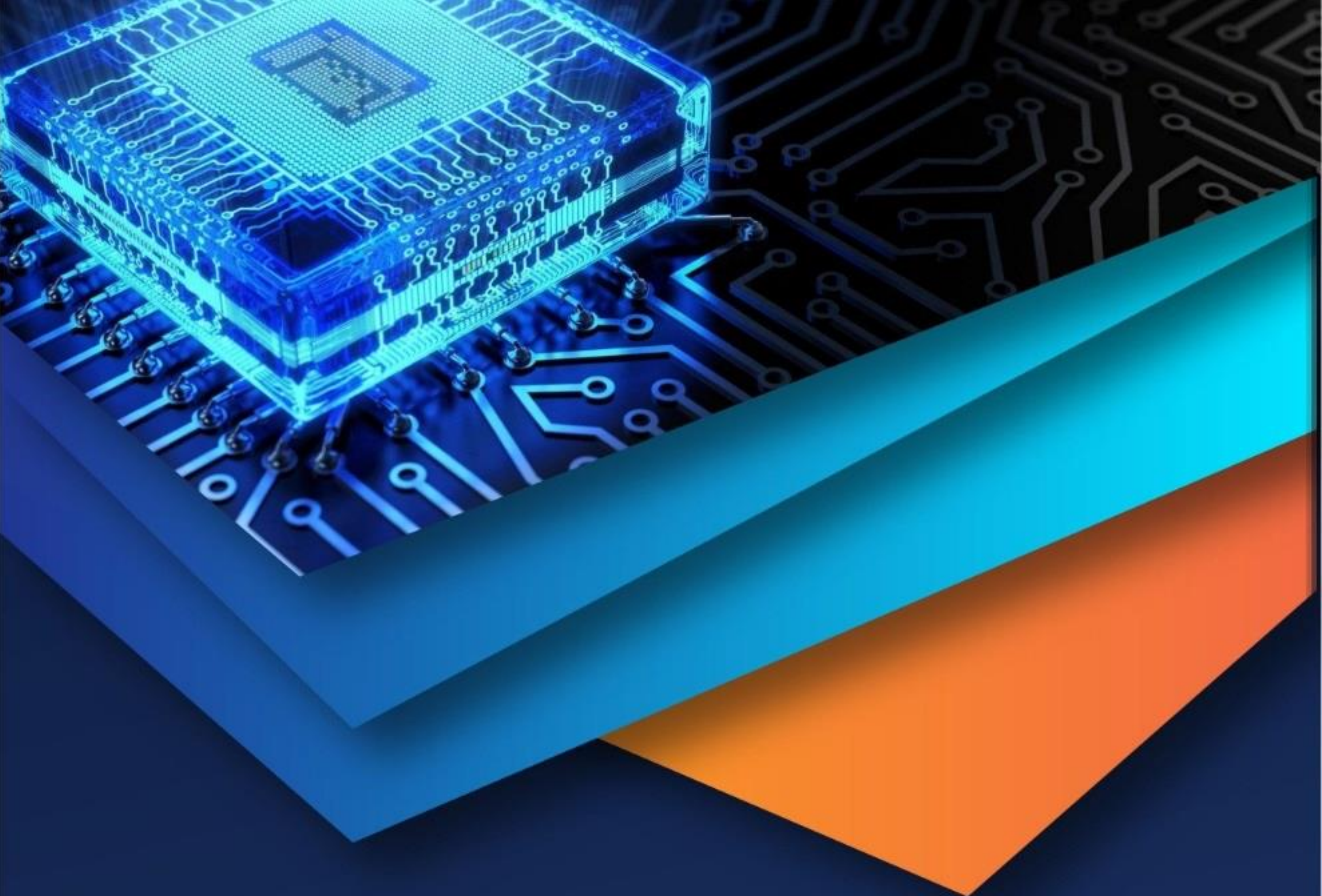
This research presents a comprehensive literature review on algorithms for fire monitoring and detection. The primary goal of these algorithms is the real-time detection and estimation of fire evolution. The suggested methods reliably identify fire in many scenarios and extract complex image fire properties automatically. YOLOv3 offers a significant advantage for fire detection and monitoring, according to a comparison of YOLOv3, SSD, R-FCN, Celik and Faster RCNN in terms of Accuracy, False Alarm Rate and Missed Detection Rate.

Accuracy of CNN based algorithms were found out to be definitely better. The most precise algorithm based on YOLOv3, with a significantly high 83.7%, detects fire the fastest (28 Frames Per Second), and is the most resilient. This is because of its lower false alarm and missed detection rate, higher accuracy, Better speed and performance in comparison to other algorithm.YOLO v3 showed a low missed detection rate as low as 0%. It even had the second lowest false alarm rate (0.46) after Celik and the highest accuracy of 99.62%.

REFERENCES

[1] Pinkerton : India Risk Survey Report (2021).
[2] T. Celik and Hasan Demirel et. : Fire detection in video sequences using a generic color model (2009).
[3] Hidenori Maruta; Akihiro Nakamura; Fujio Kurokawa : A new approach for smoke detection with texture analysis and support vector machine (2010).
[4] Zili Zhang, Qingyong Jin,Lina wang, Zhiguo Liu : Video-based Fire Smoke Detection Using Temporal Spatial Saliency Features (2021).
[5] Surapong Surit; Watchara Chatwiriya : Forest Fire Smoke Detection in Video Based on Digital Image Processing Approach with Static and Dynamic Characteristic Analysis (2011).
[6] Osman Gunay and Habiboglu : Covariance matrix-based fire and flame detection method in video (2011).
[7] Dimitropoulos : Fire Detection from Images using Faster R-CNN and Multidimensional Texture Analysis (2019).
[8] Zhanqing Li, University of Maryland : Evaluation of Algorithms for Fire Detection and Mapping Across North America from Satellite (2003).
[9] Saad Albawi, Tareq Abed Mohammed : Understanding of a Convolutional Neural Network (2017).

[10] Ganesh Samarth, Neelanjan Bhowmik : Experimental Exploration of Compact Convolutional Neural Network Architectures for Non-temporal Real-time Fire Detection (2019).
[11] Roshan Koirala : Fire Image Detection Based on Convolutional Neural Network (2018).
[12] Zhentian Jiao, Youmin Zhang : A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection (2020).
[13] Dewi Putrie Lestari, Rifki Kosasih, Tri Handhika, Murni, Ilmiyati Sari, Achmad Fahrurozi : Fire Hotspots Detection System on CCTV Videos Using You Only Look Once (YOLO) Method and Tiny YOLO Model for High Buildings Evacuation (2019).
[14] Shuai Zhao , Boyun Liu , Zheng Chi, Shengnan Li , Taiwei Li : Characteristics-Based Fire Detection System Under the Effect of Electric Fields With Improved Yolo-v4 and ViBe(2022).
[15] Wei Fang, Lin Wang, Peiming Ren : Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments (2019).
[16] Li-Wei Kang, I-Shan Wang, Ke-Lin Chou, Shih-Yu Chen, Chuan-Yu Chang : Image-based Real-Time Fire Detection Using Deep Learning with Data Augmentation for Vision-based Surveillance Applications (2019).
[17] Zhao Rentao, Wang Mengyi, Zhai Zilong, Li Ping, Zeng Qingyu : Indoor Smoking Behavior Detection Based on YOLOv3-tiny (2019).
[18] Rustam Khamdamov, Khoshim Rakhmanov, Timur Sakiyev : Hardware-software complex detects smoke and fire from video images(2019).
[19] Dongqing Shen, Xin Chen, Minh Nguyen, Wei Qi Yan : Flame Detection Using Deep Learning (2018).
[20] Shixiao Wu, Libing Zhang : Using Popular Object Detection Methods for Real Time Forest Fire Detection(2018).
[21] Khan Muhammad, Jamil Ahmad, Zhihan Lv : Efficient Deep CNN-Based Fire Detection and Loca -lization in Video Surveillance Applications (2018).
[22] Kuo-Ming Hung, Li-Ming Chen, Jin-An Wu : Wildfire Detection in Video Images using Deep Learning and HMM for Early Fire Notification System(2019).
[23] R. Shanmuga priya, K. Vani : Deep Learning Based Forest Fire Classification and Detection in Satellite Images(2019).
[24] Al.maamoon Rasool Abdali, Rana Fareed Ghani : Robust Real-Time Fire Detector Using CNN And LSTM(2019).
[25] Raam Pujangga Sadewa , Budhi Irawan, Casi Setianingsih : Fire Detection Using Image Processing Techniques with Convolutional Neural Networks(2019).
[26] Sergio Saponara, Abdussalam Elhanashi : Exploiting R-CNN for video smoke/fire sensing in antifire surveillance indoor and outdoor systems for smart cities(2020).
[27] Jakub Gotthans, Tomas Gotthans, Roman Marsalek : Deep Convolutional Neural Network for Fire Detection(2020).
[28] Xiong Hao, Cong Linhu, He Weixin : Video Flame Detection Using Convolutional Neural Networks(2019).
[29] Metin Aktas, Ali Bayramcavus, Toygar Akgun : Multiple Instance Learning for CNN Based Fire Detection and Localization(2019).
[30] Sreejata Dutta, Soham Ghosh : Forest Fire Detection Using Combined Architecture of Sep -arable Convolution and Image Processing (2021).
[31] Joseph Redmon, Santosh Divvala, Ross Girshick ,Ali Farhadi : You Only Look Once: Unified, Real-Time Object Detection(2015).
[32] Ivan V. Zoev , Alexey P. Beresnev, d Nikolay G. Markov : Convolutional neural networks of the YOLO class in computer vision systems for mobile robotic complexes(2019).
[33] Upesh Nepal and Hossein Eslamiat : Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs (2022).
[34] Peiyuan Jiang, Daji Ergu*, Fangyao Liu, Ying Cai, Bo Ma : A Review of Yolo Algorithm Developments (2022).
[35] Sahla Muhammed Ali : Comparative Analysis of YOLOv3, YOLOv4 and YOLOv5 for Sign Language Detection(2021).
[36] Shuo Wang : Research Towards Yolo-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications(2021).
[37] S Srivastava : Comparative analysis of deep learning image detection algorithms(2021).
[38] Pu Lia, Wangda ZhaoImage: Fire detection algorithms based on convolutional neural networks (2020).
[39] Dai Quoc Tran , Minsoo Park, Yuntae Jeon, Jinyeong Bak, Seunghee Park: Forest-FireResponse System Using Deep-Learning-Based App -roaches With CCTV Images and Weather Data (2022).
[40] Akmalbek Abdusalomov ,Nodirbek Baratov ,Alpamis Kutlimuratov, Taeg Keun Whangbo : An Improvement of the Fire Detection & Classification Method Using YOLOv3 for Surveillance Systems (2021).
[41] https://drive.google.com/file/d/11KBgDW2yOxhJnUMiyBkBzXDPXhVmvCt
[42] https://www.kaggle.com/phylake1337/fire-datase
[43] https://github.com/cair/Fire-Detection-Image-Dataset/find/master
[44] https://stock.adobe.com/
[45] https://people.csail.mit.edu/torralba/code/spatialenvelope/
[46] https://www.visuallocalization.net/datasets/
[47] Bradski, G. : The OpenCV Library. Dr. Dobb's Journal of Software Tools(2000).
[48] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines (2001).
[49] Pu li, Wangda Zhao: Image fire Detection Algorithms Based on Convolutional Neural Network (2020). URL:https://doi.org/10.1016/j.csite.2020.100625



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY