Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766359

A CUSTOM YOLOV5-BASED REAL-TIME FIRE DETECTION SYSTEM: A DEEP LEARNING APPROACH

V. Bharathi
Professor, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College.

M. Vishwaa
Under Graduate Student, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College.

K. Elangkavi
Under Graduate Student, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College.

V. Hari Krishnan
Under Graduate Student, Department of Electronics and Communication Engineering, Sri Manakula Vinayagar Engineering College.

Abstract
Fire is a serious hazard in different places around the world. The detection of fire attains greater importance in the last decadesdue to the loss and damages caused by fires. Thus, A fire detection system is required for minimizing injuries with financial loss. Previous approaches to fire detection using deep learning have relied on a Convolutional neural network (CNN), which have limitation in terms of speed and real-time detection. This paper proposes a custom YOLOv5 fire detection system providing high accuracy and improved real-time performance. This work represents a real-time video fed into a deep learning model and this approach shows promising results in detecting the fire with high accuracy of 98.3%, precision of 98.6%, F1– score of 96% with low false positive rate. The system is deployed on a Raspberry Pi 3 Model B for efficient and low-cost implementation, providing a timely warning to prevent property damage and loss of life.

Keywords Fire detection, YOLOv5, Real-time video, Raspberry pi, Deep learning.

INTRODUCTION

Fires are always endangering ecosystems, infrastructure, and people's lives. As urbanization accelerates, it can also increase the frequency of fire and causes great damage to human life and property. In places where fireconstitutes an unjustifiable threat to property or human life, it is necessary to mitigate fire risk. Because of damage done by fire hazards is so great, early identification of fires has now become increasingly crucial. Numerous sites have lately installed fire alarmsthat detect fires using smoke, temp, and light sensitivities. Unfortunately, it is insufficient to suit the requirements of big rooms, severe conditions, outdoor use, and so on.

For a long time, traditional contact sensors which include a smoke sensor, temperature monitoring, and particle sensors have just been created for fire detection.Solutions based on touch sensors are only effective in limited spaces and are insufficient for a wide range of environments. Traditional fire detection systems using sensors have several disadvantages such as the differentredundant systems are necessary with fault-pronehardware systems, false and warnings regular maintenance. Maintaining a reasonable trade-off between accuracy with false alarms continues to bedifficult for these approaches.

Deep learning has been used successfully in a variety of domains, including picture categorization and object recognition. When compared to traditional fire detection techniques, CNN-based techniques significantly reduced false alarm rates & improved environmental resilience. The reliability of system discrimination is strongly impacted more by precision of flames photograph feature recognition. The fire detection problem may be handled by employing image processing technology and fire flame image features. Neural networks may be used to create the nonlinear mapping between the chance of a fire igniting and the pertinent flame characteristics since they are capable of complex pattern categorization and multi-dimensional function mapping.

VARIOUS METHODOLOGIES FOR FIRE DETECTION

H. Yang et al (2019) proposed a CNN-based monitoring system and just used Custom Method of " as an input signal to lower channel multiplier weights to 95.44% accuracy and improve performance. Z. Yin et al (2017) to speed up training, normalization convolution layers [DNCNN] were used in place of conventional convolutional layers. Moreover, DNCNN has an excellent precision of 95.28 percent and a small number of false warnings. D.Sheng et al (2021) integrated SLIC-DBSCAN using conventional convolutional neural networks, which offer a strong smoke feature extractor and a sophisticated background interference splitter. M.D.Nguyen et al (2021) integrate long short-term memory (LSTM) networks with conventional convolution neural networks. They utilized a SoftMax classifier to determine whether the flames represent a true fire or a non-fire moving object. G. Xu et al (2019) proposed an end-to-end trainable framework based on fast detector SSD and MSCNN for smoke detection. M. Ajith et al (2019) proposed K-Means, GMM, GMRF, and MRF have created a new method for detecting fire and smoke in IR images. Z. Liu et al (2019) discussed a framework for detecting smoke in high-definition video. The ViBe algorithm is used to extract motion regions, after which the number of regions to be classified is reduced through morphological processing and image block filtering, and finally, extracted small image blocks are fed to the SVM for classification. Y. Liu et al (2019) discussed a deep convolutional neural network joint dark channel proposed to detect smoke images. The experimental results indicate that the proposed method can achieve a high classification performance on real-world data sets. J. Zhang et al (2021) proposed another recognition model which is a combination of Squeeze Net and a modified Fire module on ATT U-Net allowed for more effective feature learning with limited data. For classification, The proposed architecture achieved competitive segmentation accuracy and recognition reliability. D. Q. Tran et al (2022) discussed the effective mitigation and addressing a forest fire, the right response is essential. Damage may be evaluated in real-time using a BNN model and weather data, and the searched backbone can be applied to real-time monitoring systems.

IMAGE PROCESSING & FEATURE EXTRACTION

The evaluation and manipulation of visual media from sources like images and videos are known as image processing. Image processing comprises three main processes. First, ensures the right must be converted into a binary format that that machine can comprehend. The second is a data compression and picture enhancement. The third step is an output phase that following respects or prints the revised image. Image processing methods are used in many applications, such as satellite weather mapping, machine vision, and computer-aided pattern recognition. V. Bharathi et al (2020) compared the effectiveness of Support Vector Machine (SVM), Random Forest classifiers, and Deep Convolutional Neural Network (DCNN) classifiers for the diagnosis of Parkinson's disease. C. Chaoxia et al (2020) presented a colour-guided anchoring technique that leverages flame colour properties to restrict the anchor's placement (image processing).C. -Y. Chiang et al (2020) proposed a retrained RCNN method and a transfer learning technique, which provides a new framework for automated tree detection from aerial photos. Y. Liu et al (2019) improved the CNN to extract detailed features of dark channel images using DarkC-DCN classification. V. Bharathi et al (2020) proposed the Hue saturation value (HSV) and HOG algorithms to identify people and injuries in real-time streaming video and used Support Vector Machine (SVM) for classification.

PROPOSED METHODOLOGY

In order to detect Fire, the YOLOv5 Classification technique is proposed. YOLOv5 is an object detection algorithm for deep learning that can identify objects in images and videos with high accuracy and speed. YOLOv5 uses a novel architecture that combines a CSP (Cross-Stage Partial) backbone with a spatial pyramid pooling neck and a YOLOv3like head. This architecture enables YOLOv5 to achieve state-of-the-art performance on a wide range of object detection tasks while being computationally efficient. It can detect objects in an image in a single pass. This makes the model faster and more efficient than its predecessors. It uses a modern architecture that consists of several convolutional layers and shortcut connections that improve the model's accuracy and reduce its memory footprint. The model is also highly customizable, which allows it to be trained on specific datasets for various applications.

YOLOv5 ARCHITECTURE

The anchor-free, one-stage object detection pipeline underpins the YOLOv5 design (Fig .1).

Overview of YOLOv5 Architecture:

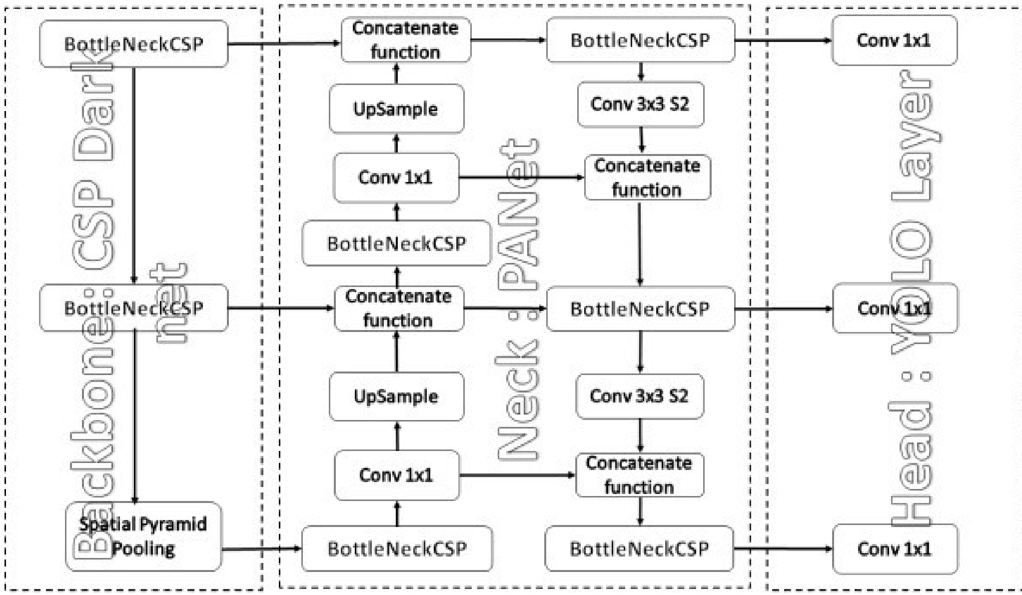

Fig 1. Network architecture for YOLO v5

BACKBONE: YOLOv5 uses a modified version of the CSPDarknet53 backbone as the feature extractor. CSPDarknet53 is a deep residual network with skip connections, It is intended to increase the model's training speed and accuracy.

NECK: The model then uses a neck architecture to fuse the features extracted by the backbone. YOLOv5 uses an SPP (Spatial Pyramid Pooling) module to aggregate contextual information across multiple scales and improve the model's recognition of items of various sizes.

HEAD: Final part of the model is detection head, which predicts the bounding boxes and class probabilities of the objects in the image. YOLOv5 uses a novel anchor-free architecture called YOLOv5 Head, which predicts the offsets of the bounding boxes directly, without the need for anchor boxes. This approach simplifies the architecture and improves the performance of the model.

DATASET

The first step in applying YOLOv5 for fire detection is to gather a dataset of images and/or videos that contain examples of fires and non-fires (Fig .2). This dataset shouldbe diverse and include different types of fires and different lighting conditions. The more diverse and comprehensive the dataset, the better the model will be at detecting fires inreal-world scenarios. The data has been obtained from the Kaggle dataset of 3250 images subdivided into 2550 non-fire images and 700 fire images. The entire dataset is broken into training (70%) and test set (30%) (Table .1). Labeling the dataset plays an important role in identifying the areas of the frame where a fire is present. This is done using an annotation tool called labeling Ing. The labeled images are then used to train the YOLOv5 model.

Fig 2. Fire Dataset Photo

Table 1. Training and testing data set of fire and Non-fire images.

Dataset	Fire image	Non-fire image
Training	490	1785
Testing	210	865

TRAINING

Once the dataset is prepared, the next step is to train a custom YOLOv5 model to detect fires in images or videos. This involves using the open-source YOLOv5 code base, which includes scripts for training and evaluating models. During training, the model learns to identify the features that distinguish fires from non-fires in images or videos. The YOLOv5 model is trained on a GPU, which fast up the training process and enhances the accuracy of model. The processes involved in training the model are shown in the (Fig .3).

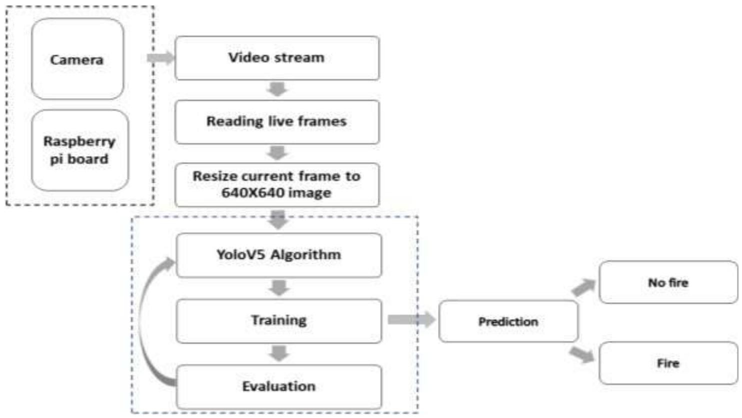

Fig 3. Flow diagram for proposed module

MODEL ARCHITECTURE

Model architecture involves specifying the number of layers in model, the size of input image, and number of output classes. The architecture can be defined using a YAML configuration file, which specifies the model's hyperparameters, such as the learning rate, batch size, and optimizer. The YAML file can be edited to adjust these hyperparameters based on the size of the dataset, the available computing resources, and other factors.

TRAINING TABLE

In Training parameter table, it summarizes the key hyperparameters and training settings used during the training of a machine learning model. The training table (Table .2) has been compiled that summarizes the training parameters and their corresponding details, based on the proposed model.

Table 2. Training parameters of the fire detection model

Training parameter	**Details**
Batch size	64
Subdivision	8
Learning rate	0.001
Numbers of masks	3

IMAGE AUGMENTATION

By applying random changes to the pictures or frames, the practice of image augmentation enhances the dataset. By doing so, the dataset's heterogeneity and durability all are enhanced. It can be done by flipping the images or frames horizontally or vertically, rotating them by a random angle, changing the brightness or contrast, and adding noise.

IMAGE ANNOTATION

Image annotation involves splitting the dataset into training and validation sets, and creating annotation files that specify the location of the fire in each image or video frame. In the YOLOv5 algorithm, bounding box annotation is carried off. Each annotation file includes bounding box coordinates of the fire and the class label, which in this case is fire.

TRAIN THE MODEL

During training, the model learns to recognize fire regions in the images and predict bounding boxes around them. The training method entails reducing the discrepancy between the dataset's ground truth frames and predicted bounding box coordinates. The Image annotation is converted into PASCAL VOL XML format which is required by YOLO v5. Compared to other methodologies, YOLO v5 required less training time (Table .3).

Table 3. Comparison of training and prediction time.

Method	TrainingTime	Predictiontime (s)
Proposed Model	1:55:00	2.1
AlexNet (Krizhevsky,A et al 2012)	9:26:40	7.9
VGG16(Simonyan, K et al (2015)	3:20:00	2.4
VGG19(Simonyan, K et al (2015)	4:43:20	3.1

DEPLOY THE MODEL

The model can be deployed on a target system, such as a camera or sensor network. It can be integrated into a software system that monitors the video feed for signs of fire. This study integrates a raspberry pi to interface a camera with the deep learning model.

RESULT

In order to evaluate our proposed model, different evaluation metrics such as accuracy rate, recall, precision, F-1 measure are used. The detailed result is shown in (Table .4).

Table 4. performance of YOLOv5 on our test dataset

Metrics	**Our dataset**
Accuracy	98.3
True positives	207
True negatives	851
Recall	93.6
Precision	98.6
F-1 measure	96

The representation of (Fig .4) demonstrates that the proposed system shows a stable Loss vs Batch Curve with minimal variance across epochs which determines the overfitting of the model to training data and it can adapt to new data. This is critical for real-world applications, because the model must be able to detection of flame reliably under a variety of diverse settings and circumstances. By comparing other CNN algorithms, YOLOv5 algorithm is faster due to its smaller model size, optimized architecture, better use of GPU, dynamic input resolution, and efficient backbone network. Its smaller model size is achieved by using a single backbone network, while its optimized architecture and better use of GPU parallelize the computation and reduce the computational complexity of the model.

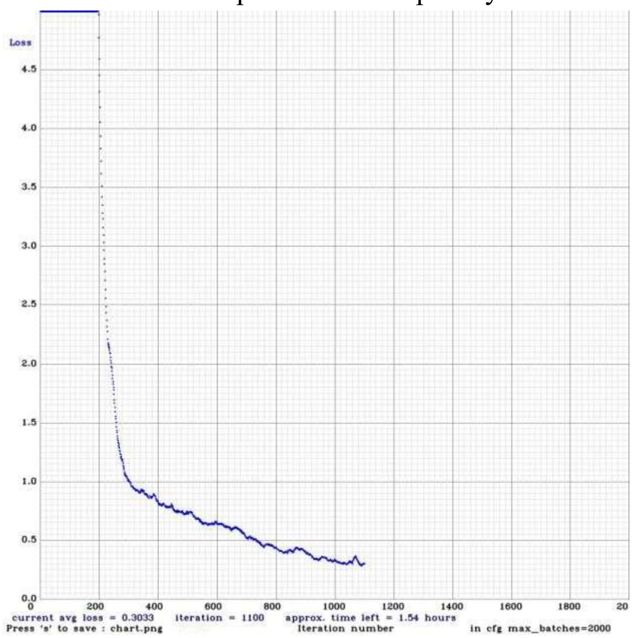

Fig 4. Loss vs Batch Curve highlighting the training lossof a deep learning model changes as the batch size is varied.

The comparison between the YOLOv5 algorithm with other mainstream algorithms is made (Table .5). Accuracy rate of custom YOLOv5 algorithm in this paper reaches 98.3% which is lighter than the other algorithms in the table.

Table 5. Performance of YOLOv5 vs other methodologie

Method	Accuracy (%)	Recall (%)	F- 1 Score(%)	Precision (%)
Our method	98.3	93.6	96	98.6
Faster R- CNN (Wu, S et al 2018)	97.4	94.5	87.2	82
MoAm- YOLO v4 (Zhang, Y et al 2022)	93.45	94.5	92	93.2

CNN and LSTM NETWORK (H. N. Vu et al 2021)	93.3	91	91.5	92
YOLO v3+OHEM(S hi F et al2020)	84.5	90.2	82	86.6

The proposed system was able to detect fires in real-time video streams (Fig .5) with an average detecting time of 0.02 seconds per frame, making it a viable solution for firedetection of practical applications. The system successfullydetected 98% of all fires that occurred during the 2 hours, with a false positive rate of 2%.

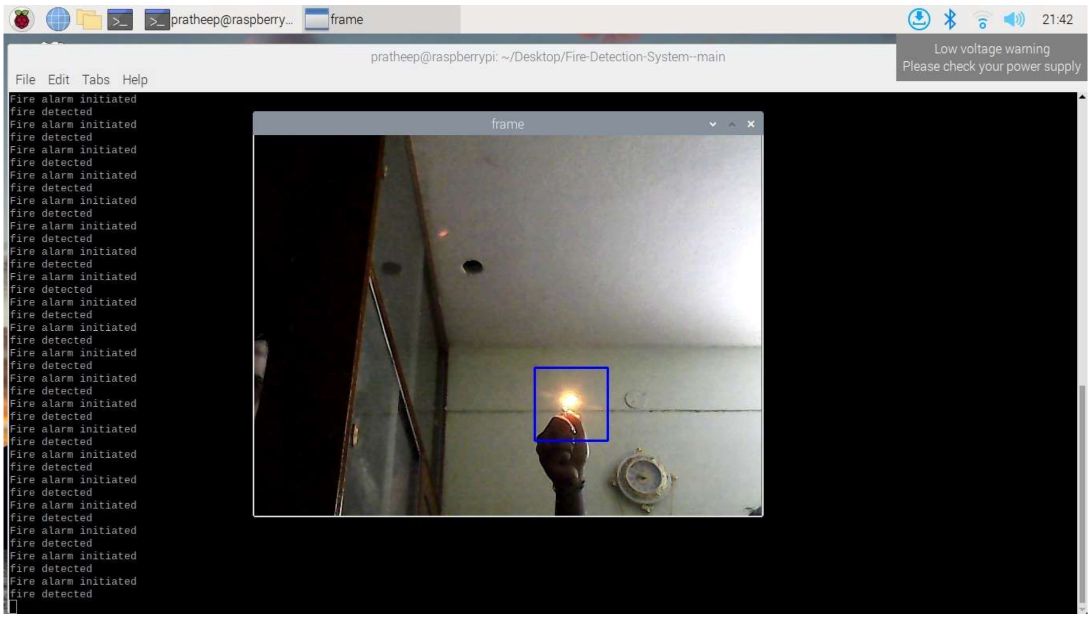

Fig .5 Visual result of real-time Fire detection of small size

CONCLUSION

A custom YOLOv5-based real-time fire detection system that uses deep learning approach to detection fires in videostreams is proposed. The model was evaluated on a different validation set to test its accuracy and speed after being trained on the dataset of fire pictures. Experimental results showed that the fire detection model by custom YOLOv5 algorithm achieved a detection time of 0.02 second per frames with improved real-time performance.

The proposed method has high accuracy rate of 98.3% which is more than the other methods as shown in (Fig .6) and a high F1- score of 98 which shows the low false positive rate. It achieves fairly moderate recall thereby making it suitable for detecting small fire objects. The system has the potential to provide an early warning of fireoutbreaks and help prevent damage and loss of life. Futureresearch could focus on improving the model's accuracy and speed, expanding the dataset used for training, and integrating the system with other technologies such asdrones or robotics.

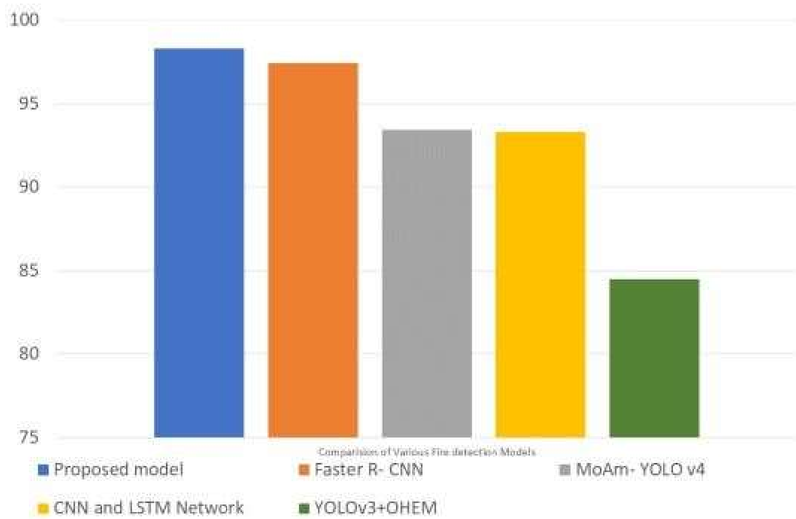

Fig .6 Bar chart comparison of accuracy of the various firedetection model

REFERENCES

Z., Yin, B., Wan, F., Yuan, X., Xia, J., Shi.:A Deep Normalization and Convolutional Neural Network for Image Smoke Detection. IEEE Access, vol. 5, pp. 18429-18438 (2017), doi: 10.1109/ACCESS.2017.2747399.

H.,Yang, H., Jang, T., Kim, B., Lee.: Non-Temporal Lightweight Fire Detection Network for Intelligent Surveillance Systems. IEEE Access, vol. 7, pp. 169257-169266 (2019), doi: 10.1109/ACCESS.2019.2953558.

G., Xu, Q., Zhang, D., Liu, G., Lin, J., Wang, Y., Zhang.: Adversarial Adaptation From Synthesis to Reality in Fast Detector for Smoke Detection. IEEE Access, vol. 7, pp. 29471-29483 (2019), doi: 10.1109/ACCESS.2019.2902606.

M., Ajith, M., Martínez-Ramón.: Unsupervised Segmentation of Fire and Smoke From Infra-Red Videos. IEEE Access, vol. 7, pp. 182381-182394, 2019, doi: 10.1109/ACCESS.2019.2960209.

Z., Liu, X., Yang, Y., Liu, Z., Qian.: Smoke-Detection Framework for High-Definition Video Using Fused Spatial- and Frequency-Domain Features. IEEE Access, vol. 7, pp. 89687-89701 (2019), doi: 10.1109/ACCESS.2019.2926571.

C. -Y., Chiang, C., Barnes, P., Angelov, R. Jiang.: Deep Learning-Based Automated Forest Health Diagnosis From Aerial Images. IEEE Access, vol. 8, pp. 144064- 144076 (2020), doi: 10.1109/ACCESS.2020.3012417.

Y., Liu, W., Qin, K., Liu, F., Zhang, Z. Xiao.: A Dual Convolution Network Using Dark Channel Prior for Image Smoke Classification. IEEE Access, vol. 7, pp. 6069760706 (2019), doi: 10.1109/ACCESS.2019.2915599.

C., Chaoxia, W., Shang, F., Zhang, : Information Guided Flame Detection Based on Faster R-CNN. IEEE Access, vol. 8, pp. 58923-58932 (2020), doi:

10.1109/ACCESS.2020.2982994.

H. N., Vu, D. C., Pham, B., Choi, S. Ro.: Multistage RealTime Fire Detection Using Convolutional Neural Networks and Long Short-Term Memory Networks. IEEE Access, vol. 9, pp. 146667-146679 (2021), doi: 10.1109/ACCESS.2021.3122346.

D., Sheng, J., Deng, J., Xiang.: Automatic Smoke Detection Based on SLIC-DBSCAN Enhanced Convolutional Neural Network. IEEE Access, vol. 9, pp. 63933-63942 (2021), doi: 10.1109/ACCESS.2021.3075731.

J., Zhang, H., Zhu, P., Wang, X., Ling.: ATT Squeeze UNet: A Lightweight Network for Forest Fire Detection and Recognition. IEEE Access, vol. 9, pp. 10858-10870 (2021), doi: 10.1109/ACCESS.2021.3050628.

D. Q,. Tran, M., Park, Y., Jeon, J., Bak, S. Park.: ForestFire Response System Using Deep-Learning-Based Approaches With CCTV Images and Weather Data. IEEE Access, vol. 10, pp. 66061-66071 (2022), doi: 10.1109/ACCESS.2022.3184707.

V., Bharathi, Oviya kirubanatham.: Detection of Wound in Human using HOG and HSV Algorithm. Journal of advanced research in dynamical and control systems, Vol. 12, No.06, pp.1897-1906, (2020)

V., Bharathi, A., Devika, J., Jaya Nishanthini, A., Periya Prasanna, R., Sai Charanya.: Parkinson's Disease Classification Using Deep Convolutional Neural Network Based on Acoustic Signals, International Journal of Future Generation Communication and Networking, Vol. 13, No.3, pp. 1780-1788, (2020)

Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25. Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates Inc.: Red Hook, NY, USA, (2012); pp. 1097–1105.

Simonyan, K.; Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, (7–9 May 2015).

Valikhujaev Y., Abdusalomov A., Cho Y.I.: Automatic fire and smoke detection method for surveillance systems based on dilated CNNs. Atmosphere. (2020);11:1241. doi: 10.3390/atmos11111241.

Wu, S.; Zhang, L.: Using popular object detection methods for real time forest fire detection. Proceedings of the 11th International Symposium on Computational Intelligence and Design (SCID), Hangzhou, China (8– 9 December 2018); pp. 280–284.

Shi F., Qian H., Chen W., Huang M., Wan Z.: A Fire Monitoring and Alarm System Based on YOLOv3 with OHEM. Proceedings of the 2020 39th Chinese Control Conference (CCC); Shenyang, China (27–29 July 2020); pp. 7322–7327.

Zhang, Y. , Lin, Q. , Qin, C. Ge, H.: Forest Fire Smoke Detection Method Based on MoAm-

YOLOv4 Algorithm. Journal of Computer and Communications, 10, 1-14 (2022). doi: 10.4236/jcc.2022.1011001.