Grenze International Journal of Engineering and Technology, January Issue

Forest Fire Detection using Next-Gen Technologies for Ecosystem Protection

Nandini K[1], Kavyashree I Pattan[2], Arpita Paria[3], Diana George[4], Girisha G S[5] and Renuka Devi M N[6]
[1-6]Department of Computer Science and Engineering, Dayananda Sagar University Bangalore, India
Email: nandini-cse@dsu.edu.in, kavyashreepattan5536@gmail.com, arpita.paria-cse@dsu.edu.in, Dianageorge1994@gmail.com, girisha-cse@dsu.edu.in, renukadevi.m-cse@dsu.edu.in

***Abstract*— We present our research work. Forest fire detection framework that leverages multiple deep learning models—YOLOv8, Keras SNN, MobileNetV2, ResNet50, and EfficientNetB0—and some data analysis methods. To propose a robust and comprehensive framework using the real-time capability for object detection enabled by YOLOv8, in our system, two main signs of a forest fire would quickly and accurately detect fire and smoke. Along with that, the characteristic extraction and classification accuracy were augmented by CNN, ResNet50, MobileNetV2, and EfficientNetB0, among others. Keras SNN provided improved performance with respect to dynamic environments. It will notify by email the department associated with and in charge of firefighting and detection of a fire so that action is taken accordingly. The data analysis would also help in identifying risk zones prone to catching fire and working on structural requirements after analysing environmental parameters like temperature, humidity, and wind. A central dashboard gives real-time streaming along with alerts. Combining the efficiency of YOLOv8 in predicting bounding boxes with the feature-extraction strengths of CNN-based models gives one chance for quick and reliable detection of forest fires. This combination of advanced machine learning provides a scalable and adaptable solution that constitutes a tremendously effective early warning system used in forest fire prevention and mitigation efforts.**

***Index Terms*— Object detection, YOLOv8, Data Analytics.**

I. INTRODUCTION

A forest fire is an unplanned, unrestrained blaze that occurs within a combustible vegetation area. A dependable system for detecting forest fires is crucial to provide early warnings and allocate resources effectively. Early detection of a forest fire plays a pivotal role in mitigating the destructive consequences associated with forest fires. Fast recognition here and ignition allows for quick detection and enables the timely deployment of firefighting resources, reducing the flame spread and limiting the total impact of the fire. Early detection plays a crucial role in safeguarding biodiversity by preventing the loss of flora and fauna, which also protects property, ecosystems, and human lives. Furthermore, early detection contributes to the preservation of air quality, given that forest fires release harmful matter and gases. Hence, the importance of early detection lies in its ability to safeguard lives, maintain environmental equilibrium and conserve the ecosystems, making it a very important element in global comprehensive wildfire management strategies.

Grenze ID: 01.GIJET.12.1.327

A. Role of Data Analytics and CNN

As we explore the applications of CNN and Data Analytics in the forest fire detection domain, CNN, a subset of deep learning techniques, has shown great success in image recognition tasks, making it a perfect fit for analyzing visual data for higher efficiency. On the other hand, data analytics seek to identify anomalies, patterns, and early indicators of a forest fire, which enables measures to be taken before the fire escalates.

B. History of Forest Fires in India

As per studies conducted by the Forest Survey of India as of 2023, the country is currently grappling with around 337 active forest fires. Over the past two decades, from 2000- 2021, the frequency of these forest fires has increased by an alarming 52%. According to official information, from November 2022 to June 2023, Madhya Pradesh recorded the highest number of forest fires, followed by other states such as Chhattisgarh, Odisha and Maharashtra. Because of an increase in population, the ecosystem struggles to survive. Increased human intervention in the forests has heightened the frequency of wildfires. This makes a forest fire detection system a necessity.

II. LITERATURE SURVEY

Pu Li, Wangda Zhao et. Al.[1] Recent advancements in image-based fire detection technologies have significantly contributed to early fire warning systems, thereby reducing fire-related losses. Traditional methods relying on manual or conventional automatic feature extraction often suffer from limited accuracy, delayed response, and high computational demands. To address these challenges, modern approaches utilize advanced object detection convolutional neural network (CNN) models such as Faster R-CNN, R-FCN, SSD, and YOLO v3. These models enhance the precision and speed of fire detection by leveraging deep learning techniques for more accurate image analysis. Among these, YOLO v3 demonstrates superior performance with an average precision of 83.7% and a real-time processing speed of 28 frames per second (FPS), outperforming other models in both accuracy and robustness. The literature highlights YOLO v3 as a particularly effective solution for real-time fire detection applications due to its balance of speed and reliability

Sung Wook Baik et al. shows a shift from traditional sensor-based fire detection to vision-based methods that use CNNs for better feature extraction. Early CNN models like AlexNet and VGGNet improved accuracy but required a lot of computation, which limited their use in real-time situations. GoogleNet, with its optimized inception modules, offers a more efficient option for fire detection. Many studies have looked at lightweight CNNs and fine-tuning strategies to boost accuracy while keeping computational costs low. However, there is still a gap in creating models that balance real-time performance with accuracy for embedded surveillance systems [2].

Filonenko, Alexander et al. describe effective modern convolutional neural network architectures for smoke detection in images. Models like AlexNet, VGG, ResNet, Inception-V3, Inception-V4, and Xception have performed well on large datasets such as ImageNet. These models were evaluated for this specific task. The results specify that Inception-based architectures excel when the training dataset includes a wide range of scenarios. This improved generalization [3]. In contrast, in older architectures, under similar kinds of conditions, there is a significant loss in accuracy. This highlights the standing of using newer models for tasks that contains limited data and high variability.

Horvat, Marko et al. [4] Deep learning model uses the widely used YOLOv5 Model, which shows great efficiency in computer vision tasks, mainly in real-time object detection. It is one of the most popular object detection frameworks, to its ability to handle noisy and complex datasets, well-known for speed and accuracy. It uses the Python programming language to use in research and real-world applications. For specific situations to find the best model, many studies emphasise the importance of comparing different YOLOv5.Comparative studies of standard image datasets help researchers to understand the trade-offs between computational resources, speed, and accuracy, allowing for better model selection.

Aicha Khalfaoui and her team [5]. In recent advances, enhancing the security in smart cities is very crucial using person-finding technology, a computer vision has significantly improved. It’s hard to spot people in real life because everyone stands differently, wears different stuff, and sometimes gets blocked from view. The YOLO (You Only Look Once) group of algorithms has become a go-to for finding things fast. in these Studies, head-to-head on datasets like Penn-Fudan that put YOLOv3 and YOLOv5, it shows what they're good at and where they fall short. YOLOv3 is very fast, making it a good pick when time is of the essence, but YOLOv5 it spots people better when things get tricky, so it is more accurate. This means when picking person-finding models, you have to pick between speed and accuracy for security in smart cities.

Zhao, Zhong-Qiu, et al.[6] Object detection has always been super important in computer vision because it helps us figure out what's in images and videos. Early methods used basic features and simple models, but they didn't work so well, even with lots of extra image details. That is why deep learning, especially CNNs, came along—they can learn complex features straight from data.different designs are for detection setups, training, and ways to get the most out of them, which makes them way better at being accurate and reliable. Besides normal object finders, Finding important objects we've got special ones for recognizing faces and spotting people. Each has its own tough problems. If we look at how these methods stack up, we can see what's good and bad about them. What people are working on now is making them even better and faster by tweaking their designs and using smarter practice methods.

III. METHODOLOGY

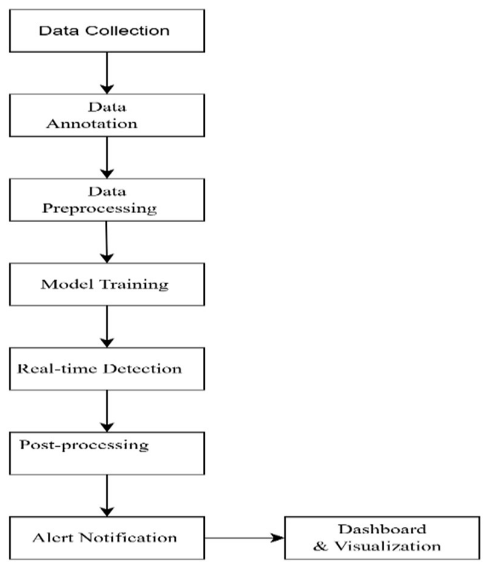

Figure 1. Data Flow for Forest Fire Detection

Early Fire detection systems play a critical role in reducing overall response efficiency by offering rapid detection capabilities, minimizing risks, and significantly reducing delays in decision-making to accelerate. By preventing the potential displacement and extinction of the biosphere, they effectively safeguard and protect diverse ecosystems. Additionally, the timely containment of the wildfires helps in saving human lives, properties and air pollution as well. Another vital scope is the conservation of carbon sequestration in trees and soil, which helps mitigate the impacts of climate change, as mentioned in Fig. 1.

Step 1: Data Collection

$$D = \{I_i \mid i = 1,2,3, \dots . n\} \quad (1)$$

D Forest Image, May or May not include fire I_i as mentioned in equations (1)

Step 2: Annotation $A = \{(I_i, B_i, C_i)\}$ *(2)*

B_i, Bounding Box, For classification $C_i \in \{\text{Fire, Non Fire}\}$ as mentioned in equations (2)

Step 3: Preprocessing motioned in equation (3)

$$I_i^{pre} = Resize\big(Normalize(I_i)\big) \quad (3)$$

Step 4: Feature Extraction

$$F_i = \sigma(W_i * F_{i-1} + b_l), l = 1,2 \dots , L \quad (4)$$

σ Activation Function Like $ReLU$ * Convolution operation
F_l High-Level Fire-related pattern as mentioned in equations (4)

Step 5: Pooling

$$F_i^{pool} = MaxPool(F_i) \quad (5)$$

as mentioned in equations (5)

Step 6: YOLOv8 Detection Model: real-time inference, the CNN should provide efficient and accurate.

$$\hat{Y} = f_{YOLOv8}(F) \quad (6)$$

YOLOv8 output like Bounding boxes, Abjectness Score and Class Probabilities (Fire/Non-Fire). as mentioned in equation in (6).

Step 7: Loss Function

which contains Box Loss as Location Accuracy, Abjectness Loss used for Fire Presence confidence and Classification loss used for detecting the Fire vs non-fire, as mentioned in equation (7).

$$\mathcal{L}_{YOLO} = \lambda_1 \mathcal{L}_{box} + \lambda_2 \mathcal{L}_{obj} + \lambda_3 \mathcal{L}_{cls} \quad (7)$$

Step 8: Backpropagation:

Gradient descent Optimizes network weights using η Learning rate (0.01) and repeats until convergence (loss minimum)

$$W_l \leftarrow W_l - \eta \frac{\partial \mathcal{L}_{YOLO}}{\partial W_l} \quad (8)$$

Step 9: Post-processing (NMS)

$$\hat{Y} = NMS(\hat{Y}, \theta) \quad (9)$$

Non-maximum suppression removes duplicate boxes. θ Intersection Over Union (IoU) threshold (0,5), High-Confidence, on-overlapping detections.

Step 10: Evaluation Matrices

$$Precision\ (P) = \frac{TP(True\ Positive)}{TP(True\ Positive) + FP(Flase\ Positive)}$$

$$Recall(R) = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precison + Rec} \quad (10)$$

A. Significance of CNN-based Approaches for a forest fire detection system

The flexibility of ecosystems is sponsored while facilitating effective recovery by avoiding timber loss and reducing soil erosion. thereby supporting effective recovery. Additionally, human safety requires putting efforts into timely evacuation and firefighting. Overall, multiple environmental challenges, creating a positive impact on both flora and fauna, need to be tackled on a CNN-based forest fire detection system for humans and the biosphere as a whole. CNNs are especially known for their efficacy in image recognition tasks, and are well-suited for analyzing satellite imagery and aerial photographs to detect wildfires. Our project strives to encompass the CNN model's capability to recognize distinct visual features associated with fires, including smoke, flames and heat patterns across various diversities [7].

B. Integration of YOLOV8

Integrating yolov8 in the detection system is a meticulous process that encompasses a broad scope aimed at enhancing and optimizing the accuracy of the response of model. It is configured to maximize its capability in recognizing fire-specific features such as smoke, flames and fire spread patterns .. Yolov8 suits for applications due to its ability to utilise a diverse data-set for thorough training, and fine-tuning the model to ensure adaptability across various environments. Multisource data integration, including aerial photography, is vital to enabling real-time object detection. Geographical location system enhances spatial analytics for the precise identification of the location of the wildfire, while automated email response contributes to the swift notifications of the wildfires to take preemptive measures.

The proposed system extends to ensuring scalability, optimized computational performance, and a user-friendly interface for seamless operational management. By incorporating YOLOv8, it aims to provide a reliable and flexible solution for accurate and timely fire detection in various environmental conditions.

C. Leveraging data analytics for innovative forest fire detection

YOLOv8 Model is integrated to enhance the data analytics capabilities by heightening efficiency and accuracy. The scope further extends to a predictive analysis system to forecast potential fire risks and optimize response timing. YOLOv8 model supports continuous improvement through ongoing training of the model. This approach not only ensures timely and accurate fire detection but also offers insights for proactive wildfire management. It uses the dataset Insights [8].

D. Proposed Design

The begin step is used to collect a diverse dataset that includes forest images (with and without fire) to develop a YOLOv8-based forest fire detection Model. Next, we annotate the dataset where particular data fires are present by drawing bounding boxes around the areas.. Then, we pre-process the data using methods like normalizing pixel values and resizing images. For this and future uses, we utilize Convolutional Neural Networks (CNNs). After this, we configure the model as the 'yolov8.cfg' file by adjusting its parameter, which specifies the paths to the dataset and the customizable settings. In the Second step, we pretrain on a larger dataset, keeping in mind as an option. The model should get the model's performance by evaluating on the test dataset and calculating certain metrics such as recall, F1 score, and precision. While implementing post-processing methods and techniques, some of which include non-maximum suppression, to refine the model [9].

These CNNs are deep learning models, which are mainly used in computer vision tasks and applications. They work in such a way that they automatically learn various aspects of the given input data, such as its spatial and hierarchical representations and aspects, which typically and usually include images. CNNs are made up of layers that capture local features and factors using convolution operations, activation functions are used to bring in the concept of non-linearity and important information is kept using pooling layers. The backbone structure includes connected layers that make predictions, and during training, CNNs adjust the parameters in them using optimization and back propagation algorithms, which significantly reduce the difference between the actual and predicted labels. Backing themselves up with such features, CNNs have been verified to be highly successful in performing tasks like image recognition and classification, solely due to their ability to capture and make appropriate and accurate use of the spatial hierarchies present in the input data.

CNNs involve different layers that perform operations of convolution by applying specific filters to capture spatial features. These include the use and application of Activation functions, which include non-linearity, while the pooling layers take care of down sampling data, which preserves necessary and essential information. The output later undergoes flattening and is fed into the connecting layers for global pattern learning. The final layer produces network predictions, softmax function is used for any classification tasks. During the training process of the data, the CNN adjusts its internal parameters using back-propagation. It also optimizes weights to minimize and reduce the differences that occur in between the predicted and actual output. This approach, which follows the principle of hierarchy, enables the Convolution Neural Network to automatically learn and identify complex visual patterns.

Convolutional Neural Networks (CNNs): how features are pulled out is pretty straightforward. First off, you run convolution layers on the data you put in – typically images. Then, these convolution filters, or kernels, slide all over the input, grabbing features and patterns from various spots and levels. These filters are set up to learn to spot edges, textures, and harder shapes. Activation functions are also needed they add non-linearity, which helps the network learn tricky associations. Pooling layers reduce the space needed, keeping the important stuff while making things less complicated to compute. This step-by-step way of pulling out features helps the CNN pick up on things, going from easy stuff to harder stuff as it goes deeper into the network. The feature map acts kind of like a cheat sheet that the network learned [10].

E. Assumptions and Dependencies

Developing and deploying a forest fire detection model using YOLOV8 and Convolution Neural Networks includes certain factors. The model requires a very diverse and high-quality set of data in an image framework, also used to find objects. It requires the hardware and software tools. The system also relies on certain software packages, such as PyTorch and other deep learning frameworks, to get successful performance. Training environment requires computing power and such as a GPU. Human in the Loop considerations should give option for human involvement and verification. The first step in the data collection process is to obtain a varied and representative dataset of images and their annotations covering various forest fire conditions in the Indian landscape[11]. To organize and store different datasets, such as weather reports, past fire incidents, and annotated images of forests, a centralized database should be established. A thorough approach to data storage and management is important for the success of India's YOLOV8 Ultralytics-based forest fire monitoring system.

F. Yolov8

Data undergoes pre-processing, that is to gather satellite or aerial pictures of the forested areas, considering different spectral bands and resolutions to acquire pertinent data. The images are then labeled with places that have seen fire occurrences and then the training set is expanded and diversified by using augmentation techniques to the dataset, which include rotation, flipping, and scaling. To solve class imbalance while additionally making sure the model is equally adept at recognizing both fire and non-fire instances, data balancing techniques are to be used. Finally, the preprocessed data is arranged into validation and training sets so that the YOLOv8 model can be fed with it[12].

Here, we use YOLOv8 as our forest fire detection model. The YOLOv8 model was created by Ultralytics in particular. This is a cutting-edge Object detection algorithm intended for real-time applications. Because YOLOv8 can accurately and rapidly identify items in pictures and video frames, including fire occurrences, it is a vital tool for identifying forest fires in India.

Authorities and environmental organizations in India could strengthen their capacity for early and efficient detection of possible forest fire situations by utilizing YOLOv8forfiredetection, thus avoiding any severe damage.

To enable quick action and reduce damage, the system should be able to identify and notify users of forest fires within a predetermined time range (e.g., 500 milliseconds to 1 second). In order to support more monitoring locations and grow with the dataset, the solution should scale horizontally. Make sure all data is secured when it is sent between components to stop manipulation or illegal access. To limit access, use role-based access control (RBAC), which would only allow authorized personnel to have access to crucial components and data. For ease of interaction with other emergency response systems, weather monitoring systems, the solution should be well-documented.

The system builds on deep learning approaches and employs an efficient multi-model procedure—that is, YOLOv8 with a Keras SNN, MobileNetV2, ResNet50, and EfficientNetB0 with Convolutional Neural Networks (CNNs)—for optimal performance in real-time object detection. It is based on the speed and accuracy of YOLOv8 with complementary feature extraction properties from other advanced deep learning architectures for improved fire detection performance. This was accomplished through data collection and preprocessing of a diverse annotated image dataset that included fire and smoke in different environmental conditions[13]. In order to accelerate convergence and increase detection accuracy, transfer learning was applied using pre-trained weights.

The various models contribute uniquely to the system: YOLOV8 executes real-time object detection through bounding box prediction and class confidence with efficiency. ResNet50, MobileNetV2, and EfficientNetB0 enhance feature extraction and classification through the modelling of complex patterns present in fire and smoke images. The Keras SNN Spiking Neural Networks optimize performance in dynamic and resource-limited environments for further enhanced robustness in actual-world conditions. YOLOv8 produces bounding box predictions from a set of combinations of classification loss, confidence loss, and localization loss during the training process. The CNN models, pre-trained on large datasets, are then fine-tuned with a smaller, task-relevant dataset to maximize detection accuracy. The models operate by partitioning the images into grids, where each grid cell predicts multiple bounding boxes, localizing with coordinates (x, y, width, height) and confidence scores for fire presence. Wherever there are overlapping boxes, only the box with the highest confidence score will remain after applying Non-Maximum Suppression (NMS), and the remaining boxes will be eliminated. A Feature Pyramid Network (FPN) is also incorporated in some YOLOv8 model variations to enhance multi-scale detection to give the system more robustness against a changing scale of fire severity and environmental conditions[14].

By combining these advanced models, our approach ensures a scalable and adaptable solution for early forest fire detection, maximizing precision and response efficiency. Forest fires, a local problem with global consequences, may occur more frequently than they have in the recent past as a result of rising temperatures and global warming. In this project, several datasets of Indian forest landscapes and fire situations are used to train the model. Deliverables include an annotated training and evaluation dataset, a customized

IV. RESULTS

YOLOv8 model optimized for Indian forest settings, and an extensive report outlining the model's performance metrics, constraints, and possible deployment options. This has been adjusted to be exceptionally good at spotting possible fire incidents. To make sure the model is reliable and effective, its performance is carefully assessed using a different annotated dataset. Important metrics, including precision, recall, and F1 score, are measured.

Figure.2 Analyzing video frames

As evident from Fig. 2, the system endeavors to analyze the video frames continuously, checking each frame for any presence of fire. Among those algorithms for object detection, one very popular one is YOLOv8, which is quite efficient for real-time fire detection in frames.
The detected fires have been notified to the appropriate forest fire department via email by the system without any delay, as shown in Fig. 3. This automated alert system ensures an immediate response when danger approaches, allowing the authority to take timely action and cut the potential for huge losses.

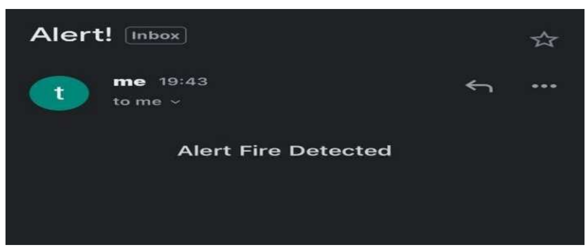

Figure 3: automated alert system

With the help of the automated email warning along with YOLOv8 object detection, the system has further improved forest fire monitoring and management, allowing a greater efficiency for fire prevention and control.

(a) (b)

Figure 4 (a) and 4 (b) fire detection frame

As seen in Fig. 4.1 and 4.2, the YOLOv8 network predicts multiple bounding boxes at once and assigns class probabilities to detected objects, thereby increasing system efficiency in real-time fire detection. These findings show how well combining object detection with deep learning models can be implemented as a scalable and precise method for forest fire prevention [15]. Also, the system was using multiple deep learning models- MobileNetV2, VGG16, ResNet50, and EfficientNetB0 were put to the test to generate results as mentioned in Table I.

TABLE I: COMPARATIVE ANALYSIS

Model	Accuracy	Precision	Recall	F1-score
MobileNetV2	91.2%	89.5%	88.0%	88.7%
VGG16	87.6%	85.2%	82.1%	83.6%
ResNet50	93.1%	91.4%	90.2%	90.8%
EfficientNetB0	94.5%	92.8%	91.6%	92.2%
YOLOV8 without CNN	96.2%	93.5%	92.3%	93.7%
YOLOV8 with CNN	98.2%	95.5%	94.1%	96.7%

EfficientNetB0, with 94.5% accuracy, was the model that had done the best among various other models since it also had the best precision and recall. Following this was ResNet50 with 93.1% accuracy and then MobileNetV2 with 91.2% accuracy. VVG16, unlike other models, performed poorly, getting an accuracy of only 87.6%.

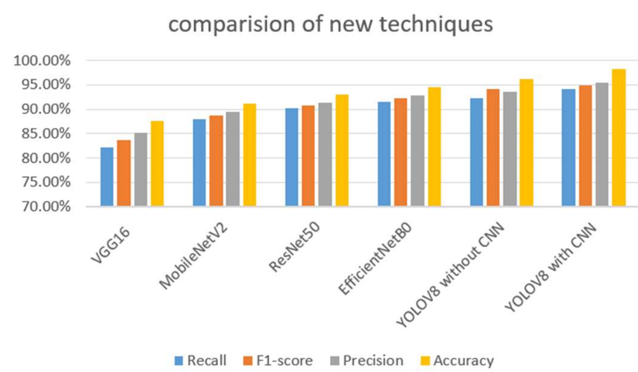

Fig 5

V. CONCLUSION

In conclusion, we developed and proposed a forest fire detection framework to unite effectively YOLOv8's real-time object detection and recognition with the powerful feature extraction and classification capabilities and strengths of models ResNet50, MobileNetV2, EfficientNetB0, and Keras SNN to result in a robust and scalable early warning solution. By combining fire and smoke detection, the system not only ensures fire outbreaks, it aids the risk assessment and proactive prevention strategies. Through the integration of fire and smoke identification, environmental data analysis, and real-time alert mechanisms, this comprehensive and adaptive approach enhances reliability, minimizes false alarms, and provides firefighting authorities with timely deliver of notifications, ultimately strengthening forest fire management to more efficiently forest fire mitigation and protect of ecosystems, human lives, and property.

REFERENCES

[1] "Li, Pu, and Wangda Zhao. "Image fire detection algorithms based on convolutional neural networks." Case Studies in Thermal Engineering 19 (2020): 100625.

[2] "Convolutional Neural Networks Based Fire Detection in Surveillance Videos", Khan Muhammad; Jamil Ahmad; Irfan Mehmood; Seungmin Rho; Sung Wook Baik., IEEE Xplore paper publications.,

[3] Filonenko, Alexander, Laksono Kurnianggoro, and Kang-Hyun Jo. "Comparative study of modern convolutional neural networks for smoke detection on image data." 2017 10th international conference on human system interactions (HSI). IEEE, 2017.

[4] "A comparative study of YOLOv5 models performance for image localization and classification"Marko Horvat , Ljudevit Jelečević, Gordan Gledec University of Zagreb., Published on:2022

[5] Khalfaoui, Aicha, Abdelmajid Badri, and Ilham EL Mourabit. "Comparative study of YOLOv3 and YOLOv5's performances for real-time person detection." 2022 2nd international conference on innovative research in applied science, engineering and technology (IRASET). IEEE, 2022.

[6] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30.11 (2019): 3212-3232.
[7] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision. Cham: Springer International Publishing, 2014.
[8] Amjoud, Ayoub Benali, and Mustapha Amrouch. "Object detection using deep learning, CNNs and vision transformers: A review." IEEE Access 11 (2023): 35479-35516.
[9] Wu, Xiongwei, Doyen Sahoo, and Steven CH Hoi. "Recent advances in deep learning for object detection." Neurocomputing 396 (2020): 39-64.
[10] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: challenges, architectural successors, datasets and applications." multimedia Tools and Applications 82.6 (2023): 9243-9275.
[11] Agarwal, Shivang, Jean Ogier Du Terrail, and Frédéric Jurie. "Recent advances in object detection in the age of deep convolutional neural networks." arXiv preprint arXiv:1809.03193 (2018).
[12] Albelwi, Saleh, and Ausif Mahmood. "A framework for designing the architectures of deep convolutional neural networks." Entropy 19.6 (2017): 242.
[13] Bengio, Yoshua, Aaron C. Courville, and Pascal Vincent. "Unsupervised feature learning and deep learning: A review and new perspectives." CoRR, abs/1206.5538 1.2665 (2012): 2012.
[14] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).
[15] Tivelius, Malcolm. "Real-time Small Object Detection using Deep Neural Networks." (2021)