AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

Fire Detection Using Machine Learning

Duggempudi Venkata Suresh
Reg. No. 24Q71F0012
duggempudisuresh70@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Mr. S. Appallaraju, Associate Professor
saraju827@gmail.com

***Abstract*—*Fire accidents cause significant damage to life, property, and the environment, making early detection essential for safety and prevention. This project focuses on developing a fire-detection system using Machine-Learning techniques to identify fire and smoke in real time. The system utilises image processing and deep-learning models, particularly Convolutional Neural Networks (CNN), to analyse visual features such as colour, texture, shape, and motion patterns associated with fire. A dataset consisting of fire and non-fire images is collected and preprocessed to train the model effectively, and the trained model is capable of accurately classifying images and detecting fire in video streams captured through surveillance cameras. When fire is detected, the system can trigger alerts such as alarms or notifications to ensure a quick response. Compared with traditional sensor-based methods, this ML-based approach offers improved accuracy, faster response time, and reduced false alarms, and enables continuous monitoring without human intervention. The prototype is implemented in Python using TensorFlow/Keras for the CNN, OpenCV for image and video processing, and a Flask or Django web interface for uploading images or video and viewing detection results. The system was validated through nineteen functional, validation, and performance test cases covering application launch, image and video upload, invalid input, model execution, fire/no-fire classification, alert generation, and real-time response, all of which passed. The proposed system can be applied in residential, industrial, and forest environments, providing a reliable and cost-effective solution for early fire detection and disaster management.*

Keywords*—*Fire Detection; Machine Learning; Convolutional Neural Network; Image Processing; Video Surveillance; Deep Learning; Real-Time Alerts; Disaster Management.

I. INTRODUCTION

The increasing occurrence of fire accidents in residential, industrial, and public environments has become a serious safety concern worldwide. Fire incidents often lead to significant loss of life, property damage, and environmental destruction. Early detection of fire is crucial in minimising these risks and enabling quick response from emergency services. Traditional fire-detection systems, such as smoke detectors and heat sensors, are widely used; however, they often suffer from limitations such as delayed response, false alarms, and inability to analyse complex environments effectively.

With the advancement of technology, Machine Learning has emerged as a powerful approach for intelligent fire-detection systems. Machine learning enables systems to learn patterns from data and make

predictions without being explicitly programmed; by analysing images, videos, temperature readings, or sensor data, ML-based fire-detection systems can identify fire incidents more accurately and quickly than conventional methods. In real-world scenarios, fire detection is challenging due to varying environmental conditions such as smoke-like objects, lighting variations, fog, and heat reflections, which often lead to false alarms in traditional systems, and early-stage fire detection is difficult because initial signs may be subtle and not easily detectable by simple threshold-based systems.

To overcome these challenges, machine-learning techniques such as Convolutional Neural Networks, Support Vector Machines, and other deep-learning models are widely used for fire detection. These models analyse visual data from cameras or sensor data and identify fire patterns such as flames, smoke, and heat anomalies with high accuracy. Computer-vision-based fire-detection systems use image-processing techniques to extract features from video frames and detect fire regions in real time, and deep-learning models further improve accuracy by automatically learning complex features from large datasets of fire and non-fire images.

II. LITERATURE SURVEY

Fire detection has been an active area of research due to the increasing number of fire accidents in residential, industrial, and forest environments. Early detection is essential to minimise loss and ensure timely response, but traditional smoke and heat sensors often suffer from delayed detection, false alarms, and inability to handle complex environmental conditions. With the advancement of Artificial Intelligence and Machine Learning, researchers have shifted toward intelligent fire-detection systems that analyse data more effectively to provide faster and more accurate results, using image processing, video analysis, and sensor data trained on large datasets of fire and non-fire scenarios.

Initially, fire-detection systems were based on hardware sensors such as smoke detectors, heat detectors, and flame detectors. Smoke detectors identify the presence of smoke particles, heat detectors respond to abnormal temperature increases, and flame detectors detect infrared or ultraviolet radiation; these were simple and effective in controlled environments but had several limitations. Another traditional approach involved manual monitoring through CCTV, which is prone to human error, fatigue, and delayed response. Modern approaches focus on intelligent classification using machine-learning algorithms—convolutional and recurrent networks, and image-processing techniques such as Otsu thresholding—trained on labelled fire and non-fire data, achieving stronger detection in varied real-world conditions. Foundational deep-learning work such as Krizhevsky's ImageNet CNN and He et al.'s deep residual networks underpins modern image-based fire detection.

TABLE I. EVOLUTION OF FIRE-DETECTION APPROACHES

S.No	Approach	Technique	Limitation / Note
1	Smoke / heat / flame sensors	Threshold hardware	Delayed; false alarms
2	Manual CCTV monitoring	Human observers	Fatigue, human error
3	Image processing	Colour/texture, Otsu (1979)	Limited robustness

S.No	Approach	Technique	Limitation / Note
4	Classical ML	SVM, Random Forest	Manual feature engineering
5	Deep learning (CNN)	Krizhevsky 2012; ResNet (He 2016)	Strong vision baseline
6	Video-surveillance fire detection	IEEE TIP, 2020	Real-time CCTV analysis

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Existing fire-detection systems rely mainly on smoke sensors and heat detectors, which often fail to detect fire at an early stage or generate false alarms due to environmental interference. Traditional systems are not intelligent enough to distinguish between actual fire and fire-like conditions such as steam, dust, or lighting effects, leading to unnecessary alerts and reducing trust in the system. These systems also lack real-time image or video analysis, so subtle early signs of fire often go undetected until smoke or heat levels reach detector thresholds.

Limitations of the existing system:

- Smoke/heat sensors miss early-stage fire and produce false alarms.
- Cannot distinguish fire from steam, dust, or lighting effects.
- No real-time image or video analysis capability.
- Manual CCTV monitoring suffers from human fatigue and error.
- Limited adaptability to complex environments.

B. Proposed System

The proposed system is an ML-based fire-detection platform that analyses images and video streams in real time and classifies them as Fire or No Fire. A Convolutional Neural Network is trained on a labelled dataset of fire and non-fire images and integrated into a web-based application; users can upload images or video, and the system processes input, runs prediction, displays the result with a clear label, and generates an alert when fire is detected. The model captures visual features such as flame colour, smoke texture, brightness, and motion characteristics that conventional sensors cannot evaluate.

Advantages of the proposed system:

- Real-time visual analysis of images and video streams.
- Reduced false alarms through learned fire-pattern features.
- Continuous monitoring without human intervention.
- Alert generation on fire detection (alarms / notifications).
- Web-based interface for easy upload and result viewing.
- Suitable for residential, industrial, and forest environments.

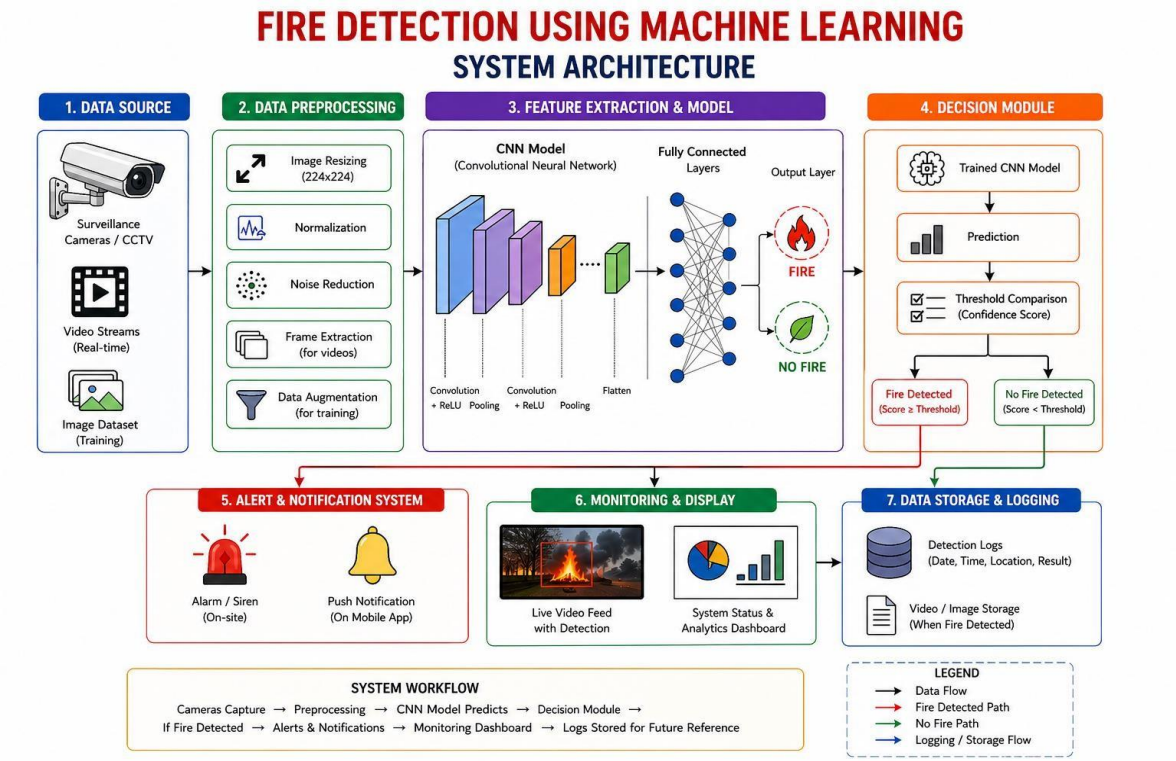
IV. SYSTEM DESIGN AND METHODOLOGY

A. Requirements

Functionally, the system must accept image or video input, preprocess and analyse it, run the trained CNN to classify each input as Fire or No Fire, display the result with a confidence score, and generate an alert when fire is detected. It must validate input formats and handle corrupted or empty files gracefully. Non-functional requirements include real-time responsiveness, accuracy under varied lighting and environmental conditions, scalability to multiple input streams, and a user-friendly web interface.

B. System Architecture

The architecture comprises an input layer that accepts images or video frames, a preprocessing layer that resizes, normalises, and augments data, a model layer hosting the trained CNN that produces the fire / no-fire classification, an alert layer that triggers notifications when fire is detected, and a presentation layer (Flask or Django web application) where users upload data and view results. Optional cloud deployment on AWS or Azure supports scalability and real-time monitoring.



C. Workflow

A user opens the web application and uploads an image or video; the file is validated and preprocessed; frames or images are passed through the trained CNN; the model outputs a label and confidence score; the result is displayed; and if fire is detected, an alert is generated. Surveillance video streams can be processed continuously to enable real-time monitoring.

V. SYSTEM IMPLEMENTATION

A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
Deep-Learning Framework	TensorFlow / Keras
Computer Vision	OpenCV
Web Framework	Flask / Django
Models	Convolutional Neural Network; transfer learning (optional)
Data	Labelled fire / non-fire image dataset
Deployment	Local server or cloud (AWS / Azure)
Alerts	Alarms / notifications on fire detection

B. Implementation Details

The implementation uses Python with TensorFlow/Keras to build and train the CNN, OpenCV for image and video processing, and Flask or Django for the web interface. The labelled dataset is divided into training and testing sets; the CNN is trained to classify images as Fire or No Fire, capturing patterns such as flame colour, smoke texture, brightness, and motion. The trained model is integrated into the web application, which accepts uploaded images or video; each frame is preprocessed and passed through the model, the classification result and confidence are displayed, and an alert is generated when fire is detected. For deployment, the system can be hosted locally or on cloud platforms such as AWS or Azure to enable real-time monitoring and scalability.

C. Inputs, Outputs, and Alerts

Inputs are static images or video frames from cameras or surveillance feeds. Outputs are a Fire or No Fire label together with a confidence score, displayed clearly in the user interface, and when fire is detected the system can trigger alarms or notifications. The pipeline is designed to be robust against variations in lighting, fog, dust, and other environmental factors that traditionally cause false alarms in sensor-based systems.

VI. SYSTEM TESTING AND RESULTS

Testing was carried out through unit testing, integration testing, and acceptance testing, with functional, validation, and performance test cases. Nineteen test cases were defined, covering application launch, image and video upload, invalid file handling, model execution, fire and non-fire detection, result display, alert generation, corrupted-file handling, large-video handling, empty input, data consistency, classification validation, response time, multiple-input handling, CPU/memory usage, and real-time detection. All test cases passed and behaved as expected.

TABLE III. REPRESENTATIVE TEST CASES

ID	Description	Input	Expected Output	Status
TC-01	Verify application launch	Open system URL	Dashboard displayed	**Pass**
TC-02	Upload valid image	Fire/non-fire image	Image accepted	**Pass**
TC-04	Invalid file upload	Unsupported format	Error message	**Pass**
TC-07	Detect fire content	Fire image	Fire Detected	**Pass**
TC-08	Detect non-fire content	Normal image	No Fire Detected	**Pass**
TC-10	Generate alert	Fire detected input	Alert triggered	**Pass**
TC-19	Real-time detection	Live input stream	Immediate output	**Pass**

A. Observed Results

The implemented system classifies images and video streams as Fire or No Fire using the trained CNN, displays the result with a confidence score, and triggers alerts when fire is detected. Compared with sensor-based systems, the ML-based approach reduces false alarms by learning robust visual signatures of fire, supports continuous monitoring without human intervention, and can be deployed for residential, industrial, and forest applications. The source reports these outcomes qualitatively; no specific numeric accuracy values are claimed here, and performance depends on dataset size, balance, and environmental conditions.

Representative screenshots from the prototype implementation:

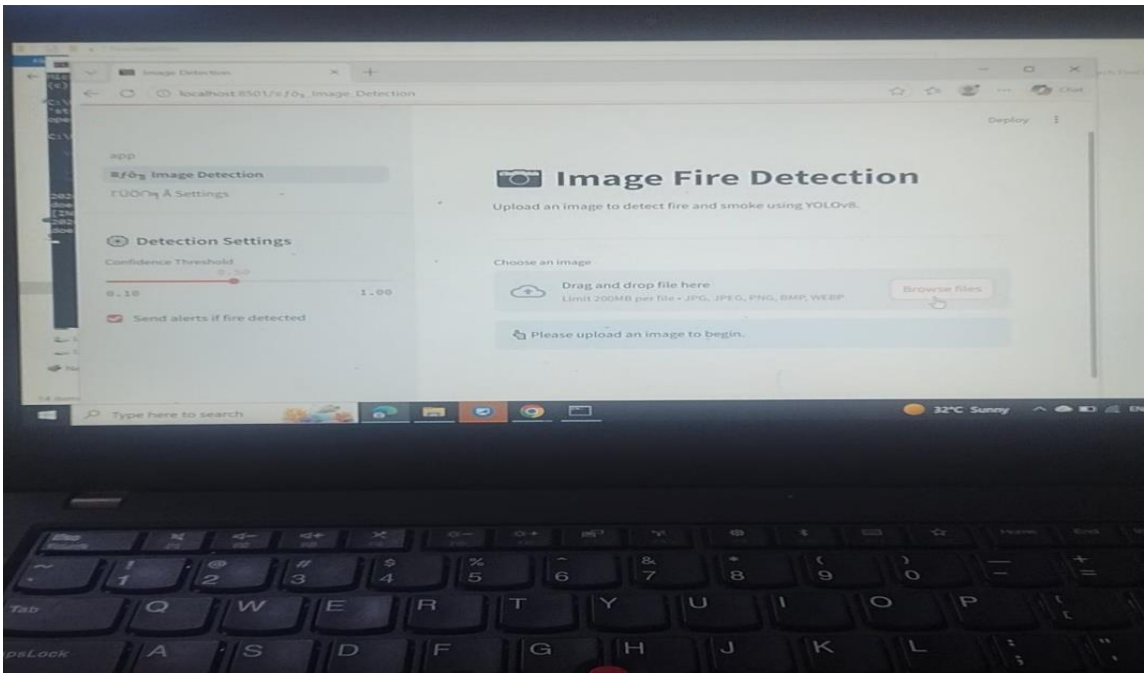


Fig. 1. Web dashboard and file upload.

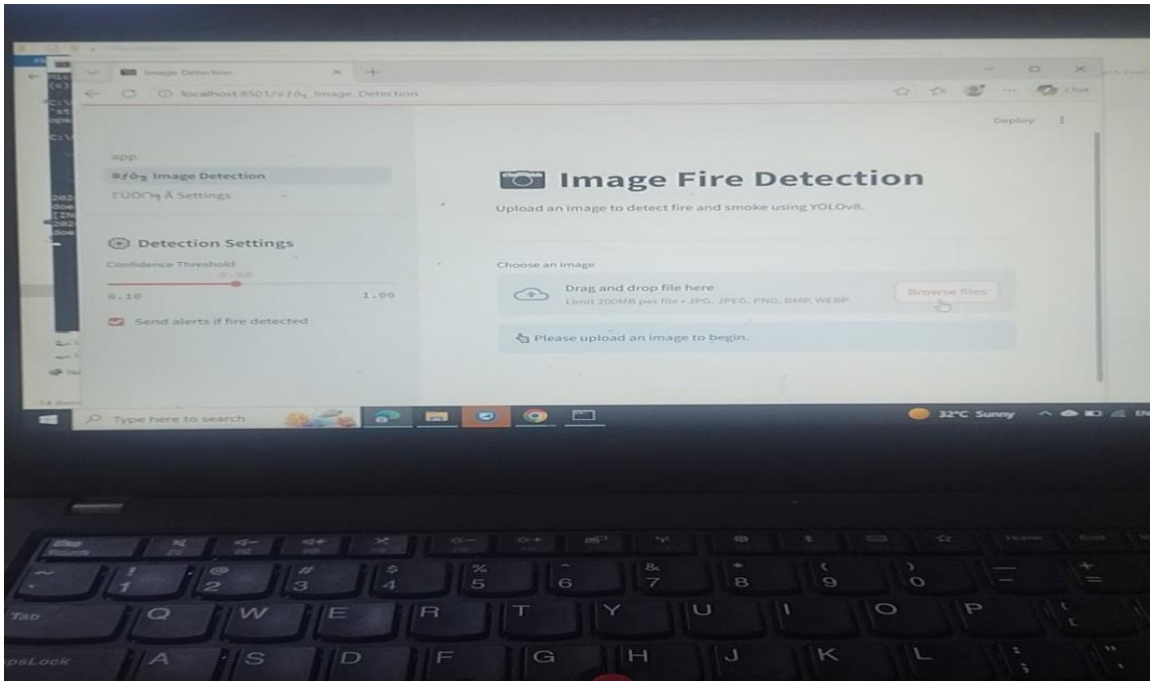


Fig. 3. CNN classification: Fire / No Fire.

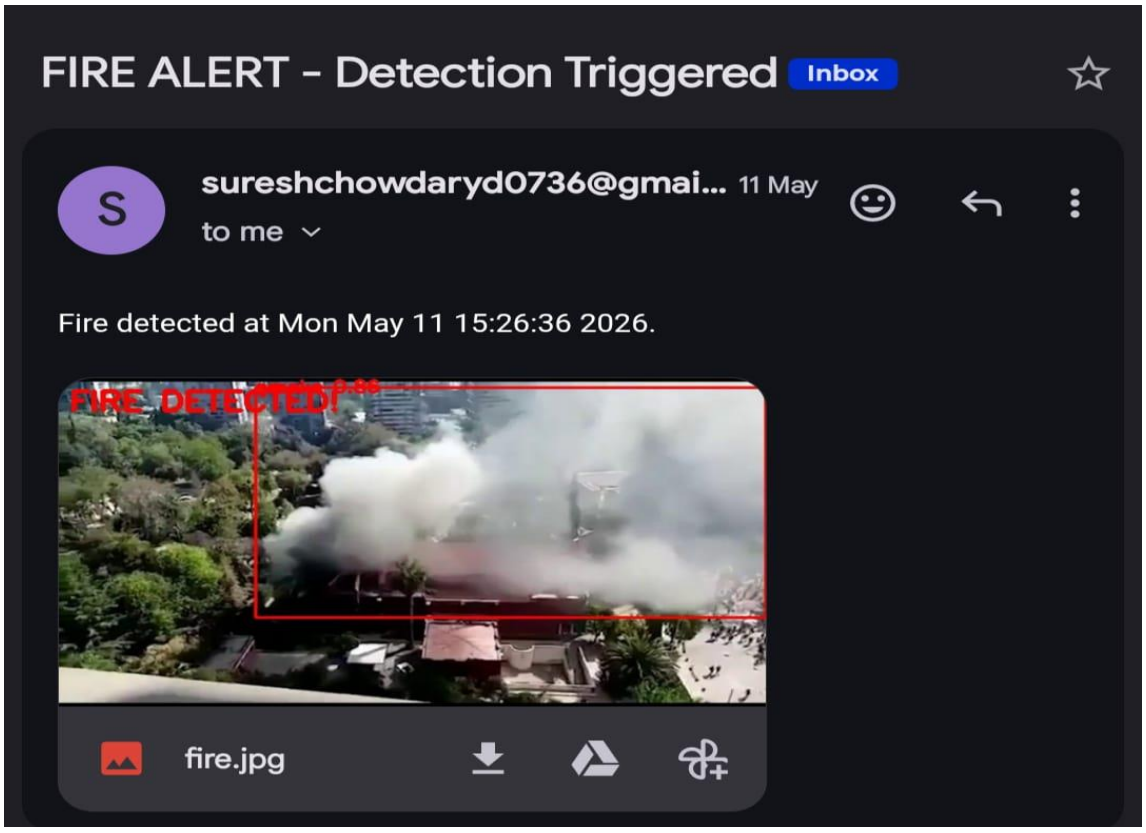


Fig. 4. Alert generation on fire detection.

VII. CONCLUSION AND FUTURE SCOPE

The project successfully demonstrates the application of Machine-Learning and Deep-Learning techniques to detect fire and smoke in images and video streams. Early fire detection in real-world environments—buildings, forests, and industrial areas—is critical to prevent loss of life and property, and this system provides an efficient and automated solution by replacing traditional manual monitoring with intelligent computer-based detection. The CNN learns patterns such as flame colour, smoke texture, brightness, and motion characteristics, accurately classifies input data, and detects fire in real time, classifying inputs as Fire Detected or No Fire Detected along with confidence scores, reducing human intervention and ensuring quick response during emergencies. The web-based interface allows users to upload images or video files, view results, and receive alerts, making the system user-friendly and suitable for real-time monitoring; compared with traditional sensor-based systems the proposed approach is more flexible, scalable, and cost-effective.

Although the proposed system provides effective fire detection, several opportunities exist for further improvement. Advanced deep-learning models such as deeper CNN architectures, recurrent models, and Vision Transformers can improve accuracy on complex patterns and video sequences. The system can be extended to real-time fire detection using live CCTV feeds, providing continuous monitoring and immediate detection. Combining vision-based machine learning with Internet of Things (IoT) sensors can produce multi-layer detection that improves reliability and reduces false alarms, and cloud-based deployment on platforms such as AWS or Azure can support large-scale, distributed monitoring for smart cities, industries, and public safety systems.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[2] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Sebastopol, CA, USA: O'Reilly Media, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems (NIPS), 2012.

[6] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[7] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, 1979.

[8] Authors, "Fire Detection in Video Surveillance," IEEE Transactions on Image Processing, 2020.

[9] OpenCV Documentation, "Open Source Computer Vision Library." [Online]. Available: https://docs.opencv.org/

[10] TensorFlow Team, "TensorFlow Machine Learning Framework Documentation." [Online]. Available: https://www.tensorflow.org/