PROJECT ABSTRACT

Vision-Based Fire Detection using CNN

A transfer-learned CNN flame classifier with a time-window alert rule and Grad-CAM explanations for indoor camera feeds — an educational prototype, never a certified fire-alarm system.

01. Title

Vision-Based Fire Detection using CNN

02. Introduction / Background

Indoor fires are found late because nobody is watching the right camera at the right moment: warehouses, server rooms and small factories have cameras, but the footage is only reviewed after the damage is done. Thermal and smoke sensors cover this properly, yet a student project cannot install certified hardware — so the honest version of the problem is building a software layer that watches camera frames for flames and raises an alert candidate for a human to verify. This project builds that layer as a CNN image classifier: a pretrained backbone fine-tuned on public flame imagery including BoWFire scores each sampled frame, a time-window rule suppresses one-off false positives, and Grad-CAM overlays make every alert visually checkable. The design target is useful alert quality with an explicit false-alarm analysis — because the failure modes of this system (sunsets through windows, bright lamps, reflective surfaces) are exactly what the report must document.

03. Problem Statement

Camera feeds in indoor facilities are reviewed only after incidents, so visible flames can go unnoticed for the minutes that matter. The need is a vision layer that scores camera frames for flames, triggers alerts only on sustained detections to suppress noise, explains its alerts visually for human review, and is evaluated with an explicit false-alarm analysis — while being honest that it is an educational prototype, never certified fire-alarm equipment and never a substitute for real detectors.

04. Objectives

The objectives are to fine-tune a pretrained CNN on public flame imagery including BoWFire; to build a frame-sampling and preprocessing pipeline for indoor camera feeds; to implement a time-window alert rule over sustained high-confidence frames; to add Grad-CAM overlays explaining each alert; to run a false-alarm analysis over bright lights, sunsets and reflections; and to evaluate with precision, recall and F1 on the validation split via the included notebook.

- Fine-tune a pretrained CNN on public flame imagery including BoWFire.
- Build a frame-sampling and preprocessing pipeline for indoor feeds.
- Implement a time-window alert rule over sustained detections.
- Add Grad-CAM overlays explaining each alert for human review.
- Run a false-alarm analysis over bright lights, sunsets and reflections.
- Evaluate with precision, recall and F1 on the validation split during the build.

05. Proposed System / Working

Frames are sampled from the indoor camera feed at a fixed interval and normalized with OpenCV. A pretrained CNN backbone, fine-tuned on public flame imagery, classifies each frame and outputs a flame probability. Frames staying above the threshold across a time window become a confirmed alert, suppressing single-frame noise; Grad-CAM overlays visualize the attended regions so a reviewer can judge whether the model saw a real flame or was fooled by a bright surface. Confirmed alerts are written to a timestamped log with annotated snapshots and can trigger an optional email or buzzer hook. The demo app serves frame scoring and the alert log. The evaluation notebook computes precision, recall and F1 on the validation split and runs the false-alarm analysis over hard negatives.

- OpenCV frame sampling and normalization from indoor camera feeds.
- Transfer-learned CNN flame classification per frame.
- Per-frame flame probability with a configurable threshold.
- Time-window alert rule over sustained detections.
- Grad-CAM overlays for reviewer judgment.
- Timestamped alert log with annotated snapshots and optional notification hook.
- Notebook evaluation: precision, recall, F1, false-alarm analysis.

06. Technologies / Components Used

The project uses Python 3.10 and PyTorch for training and inference, with torchvision supplying the pretrained backbone for transfer learning. BoWFire (Chino et al., 2015) supplies public flame imagery, supplemented by buyer-added hard negatives; OpenCV handles frame capture, preprocessing and Grad-CAM overlays; NumPy and scikit-learn compute metrics and the false-alarm analysis; a Jupyter notebook performs the evaluation; and a Flask app serves the demo.

- Python 3.10, PyTorch (CNN training and inference).
- torchvision (pretrained backbone, transfer learning).
- BoWFire public flame image dataset (Chino et al.).
- OpenCV (frame capture, preprocessing, Grad-CAM overlays).

- Jupyter notebook (evaluation with precision, recall, F1).
- NumPy, scikit-learn (metrics and false-alarm analysis).
- Flask demo app with camera-feed upload interface.

07. Key Features

Key features include a transfer-learned CNN flame classifier, fine-tuning on BoWFire flame imagery, per-frame flame probability scoring, a time-window alert trigger on sustained detections, a dedicated false-alarm analysis for lights and sunsets, an evaluation notebook computing precision, recall and F1, a demo app for camera feeds, and Grad-CAM overlays explaining each alert — always framed as an educational prototype, never certified equipment.

- Transfer-learned CNN flame classifier.
- Fine-tuning on BoWFire public flame imagery.
- Per-frame flame probability scoring.
- Time-window alert trigger on sustained detections.
- False-alarm analysis for bright lights, sunsets and reflections.
- Evaluation notebook: precision, recall, F1.
- Demo app for camera feeds and frame uploads.
- Grad-CAM overlays explaining each alert.

08. Applications / Use Cases

Vision-based flame alerting applies to warehouse and server-room monitoring aids, small-factory camera supervision prototypes, and demonstration dashboards. It also demonstrates transfer learning, alert-threshold design, and model interpretability as viva-ready deep-learning concepts — always framed as an educational prototype, never as certified fire-alarm equipment.

- Warehouse and server-room camera monitoring aids.
- Small-factory supervision prototypes.
- Demonstration and dashboard builds.
- Teaching transfer learning and Grad-CAM.
- Demonstrating false-alarm analysis in a viva.

09. Results & Scope

The expected outcome is a working flame alerting pipeline whose quality is measured, not asserted: the included notebook computes precision, recall and F1 on the validation split during the buyer's build and runs a false-alarm analysis over bright lights, sunsets and reflections. Scope covers indoor camera frames from the evaluated imagery sources; documented limits include night conditions, camera types outside the training data, smoke without visible flame, and the camera's field of view. This is an educational prototype, NOT a certified fire-alarm system — it must never substitute real smoke, heat or flame detectors, and it must never be deployed as safety infrastructure where people depend on it. Future scope includes video-sequence modeling, multi-camera alert fusion, and thermal-camera inputs.

- Measured outcome: precision, recall and F1 computed by the notebook during the buyer's build.
- False-alarm analysis documenting lights/sunset/reflection failure modes.
- Scope: indoor camera frames from evaluated sources.
- Documented limits: night conditions, out-of-coverage cameras, smoke without flame, limited field of view.
- Honest framing: educational prototype, never certified fire-alarm equipment, never a detector substitute.
- Future scope: video-sequence modeling, multi-camera fusion, thermal inputs.

10. Conclusion

A transfer-learned CNN with a time-window alert rule and Grad-CAM explanations makes vision-based flame alerting a tractable student-buildable system, and this project delivers it as a complete auditable experiment: frame pipeline, fine-tuning, alert logic, false-alarm analysis, and a notebook that measures precision, recall and F1 on the validation split. Built to order, it ships with source code, the evaluation notebook, report, presentation, and viva preparation material.

11. References

The project cites the BoWFire dataset paper, a foundational CNN fire-detection paper, and the ResNet architecture.

- D. Y. T. Chino et al., 'BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis,' Proc. SIBGRAPI, 2015.
- S. Frizzi et al., 'Convolutional Neural Network for Video Fire and Smoke Detection,' Proc. IECON, 2016.
- K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition,' Proc. CVPR, 2016.