Group 73 Final Report:
Wildfire Detection Classification Plan

Andy Huynh, Berk Yilmaz, Tanisha Tasnin
huynha3@mcmaster.ca, yilmag1@mcmaster.ca, tasnint@mcmaster.ca

1 Introduction

Wildfires are becoming increasingly prevalent and destructive due climate change, particularly in northern and temperate forest regions such as Canada [1]. This poses a growing threat to both human safety and the environment with upto 2740 deaths in Canada between 2013 to 2018 alone [3]. The economic consequences are equally substantial: a good example being the 2020 Australian mega-fires which caused approximately US $20 billion in damages [8]. Wildfires also have profound and well-documented impacts on air quality and biodiversity worldwide [7]. Early detection of wildfires is crucial for effective response and mitigation efforts. However, detecting wildfires at an early stage can be challenging. Subtle indicators such as light smoke or small flames can be obscured by dense vegetation, clouds, or varying terrain, often making traditional monitoring methods unreliable [15]. In an attempt to address this issue, our project explores the use of Convolutional Neural Networks (CNNs) for automatic wildfire detection from images. Prior work by Tsalera et al. and Spiller et al. provided valuable insights for the research going into our project, highlighting both foundational and advanced CNN-based wildfire detection techniques [14], [9]. Our project formulates the task as a binary classification problem: given an RGB image of an area, the model identifies it as a “fire” or “no fire” scenario using ‘The wildfire dataset’ by El Madafri on kaggle [4]. To benchmark performance, we begin with a conventional CNN serving as our baseline. We then created 3 separate and individual models that used modern architectures: ResNet18, MobileNetV2, and EfficientNet-B0. These model choices were motivated by the works from He et al., Tan et al., and Howard et al. [5], [12], [6].

2 Dataset

Our research utilizes the Wildfire Dataset, specifically Version 2 as published in the article *"The Wildfire Dataset: Enhancing Deep Learning-Based Forest Fire Detection with a Diverse Evolving Open-Source Dataset Focused on Data Representativeness and a Novel Multi-Task Learning Approach"* by El-Madafri et al. [4]. This dataset serves as a comprehensive benchmark for evaluating deep learning approaches to forest fire detection, addressing the critical need for reliable wildfire monitoring systems.

2.1 Data Sources and Composition

The Wildfire Dataset comprises 2,700 high-resolution RGB images collected from diverse sources including government databases (such as NASA and USGS), social media platforms like Flickr, and public domain image repositories such as Unsplash. These sources were selected to ensure geographic diversity across different forest ecosystems worldwide.

The dataset follows a two-class binary classification structure:

- **Fire**: Images showing evident fire-related phenomena
- **No Fire**: Forested areas without any signs of fire or smoke

The dataset dataset is provided in training (70%), validation (15%), and test sets (15%) with consistent class distribution across all subsets:

Table 1: Dataset partition distribution

Set	Total Images	Fire Images	No-Fire Images
Train	1,888	642	1,246
Validation	402	139	263
Test	410	109	301

This distribution maintained a roughly consistent class ratio of approximately 1 fire image for every 1.5-2 no-fire images across all partitions, preserving the original dataset's imbalance while allowing proper evaluation.

2.2 Dataset Characteristics

The Wildfire Dataset is characterized by substantial resolution variability, which presents both challenges and opportunities for deep learning models:

- Average Resolution: 4057×3155 pixels
- Minimum Resolution: 153×206 pixels
- Maximum Resolution: 19699×8974 pixels
- Standard Deviation (Width): 1867.47 pixels
- Standard Deviation (Height): 1388.60 pixels

This significant variability in image dimensions required careful preprocessing to ensure computational feasibility while maintaining the integrity of visual information.

2.3 Preprocessing Steps

Our preprocessing pipeline was designed to address the dataset's unique characteristics and enhance model performance:

2.3.1 Initial Processing

The images were organized into a directory structure with clear separation between training, validation, and test sets. Images are directly placed under "fire" or "nofire" subdirectories (as per Version 2 simplification), eliminating nested subdirectories to enhance accessibility and facilitate analysis.

2.3.2 Resolution Normalization

Given the substantial resolution range in the dataset, we conducted extensive preprocessing experiments to identify optimal image size. The final configuration selected during experimentation was 224×224 pixels after testing multiple resolutions (128×128 and 224×224 primarily). This decision was made empirically, as it provided the best results with the model we designed.

2.3.3 Data Augmentation

For the training phase, we experimented with the following augmentations:

- Rotation range: ± 15 degrees
- Horizontal shift range: up to 10% of width
- Vertical shift range: up to 10% of height
- Zoom range: up to 10%
- Horizontal flip (enabled)
- Brightness adjustment (range from 80%-120%)

This augmentation strategy aimed to increase model robustness while preventing overfitting, particularly important given the dataset's resolution variability. However, in our final iteration we did not end up using most of these. [TODO: which did we actually use?]

2.3.4 Data Normalization

All images were normalized with rescaling by a factor of $1/255.0$ to ensure consistent input range for the neural network. This preprocessing step standardizes pixel values across the entire image set and improves model convergence.

The dataset is licensed under CC BY 4.0, which allows use for our purposes with appropriate attribution to the original authors and publication.

2.4 Dataset Evolution and Quality Assurance

This version of the Wildfire Dataset represents a carefully curated collection designed for machine learning applications in forest fire detection. The researchers specifically focused on data representativeness by including:

- Images from diverse geographic locations across different continents
- A wide range of environmental conditions (e.g., time of day, weather conditions)
- Different types of forest ecosystems and vegetation
- Various wildfire stages, from early ignition to fully developed flames

3 Features and Inputs

The input to all models consists of raw RGB image pixels, treated as the feature vector for each example. Because CNNs require inputs of uniform dimensionality, all images were resized to a fixed resolution prior to training. Multiple resolutions were tested: 128×128, 224×224, 299×299, and 1000×1000. These examined how increasing the number of pixels (and thus the feature dimensionality) affects model performance. The specific

numbers for pixel sizes were chosen based on commonly discussed and used pixel sizes for CNN's found in the various cited online sources [10], [11]. However, resolutions above 224×224 caused out-of-memory (OOM) failures on our RAM due to the rapid growth in tensor sizes and batch memory requirements. We tried running the higher resolution experiments on Google Cloud as well but due to limitations on our billing account, we were unable to allocate sufficient GPU resources to handle the larger resolutions as well. As a result, larger resolutions were excluded from final experiments.

During preprocessing, images were normalized to the [0, 1] range to stabilize training and improve convergence. A manual inspection of the dataset revealed a small number of incorrectly labeled images, which introduced noise into the feature space; this observation further motivated the use of robust architectures and augmentations that help models generalize under imperfect labels. To enhance the diversity of the training samples and make the models invariant to orientation, position, and scale, we applied several image augmentations: random rotations, width and height shifts, zooming, and horizontal flips. The idea to vary the pixel size and augmentations was inspired from an online project tutorial which developed a Malaria Detection model using TensorFlow's malaria dataset [13], [2]. The scope and scale of the Malaria Detection project were similar to ours, and we adapted their augmentation strategies to our wildfire dataset. These transformations effectively vary the spatial arrangement of pixel features while preserving semantic content, allowing the models to learn more resilient representations. Overall, no handcrafted feature engineering was performed; instead, the CNN architectures learn hierarchical feature representations directly from pixel data. By varying both image resolution and augmentation strategies, we were able to study how feature dimensionality and data variability affect wildfire detection performance.

4 Implementation

4.1 Model Architecture Evolution

The initial implementation used a basic CNN architecture with four convolutional blocks, followed by dense layers. However, this approach suffered from overfitting issues (as evidenced in our progress report results). To address these limitations, we experimented with three modern architectures:

- **ResNet18**: A deep residual network that mitigates vanishing gradient problems through skip connections
- **MobileNetV2**: An efficient architecture designed for mobile devices with depthwise separable convolutions
- **EfficientNet-B0**: A scaled version of the EfficientNet family optimized for accuracy and efficiency trade-offs

These models were selected based on their proven performance in similar computer vision tasks, particularly those requiring efficient resource usage. We implemented these architectures using TensorFlow's Keras API with pre-trained ImageNet weights to leverage transfer learning.

The final implementation used MobileNetV2 as our primary model. This decision was reached empirically; it was the best performer among our experiments.

We also replaced the traditional Flatten layer with GlobalAveragePooling2D (GAP), which significantly reduced parameter count while maintaining accuracy.

4.2 Preprocessing Optimization

The preprocessing experiments included:

- Standard augmentation (rotation ±15°, horizontal/vertical shifts up to 10%, zoom range of 10
- Horizontal flip enabled for all training images
- Brightness adjustment between 80-120
- Pixel normalization by 1/255.0

We ultimately selected 224×224 pixels as the optimal size as the best trade-off between detail preservation and computational feasibility. We normalize pixels between 1 and 255. None of the other preprocessing methods helped improve performance.

Compared to our last check-in, our accuracy improved from around 76% to TODO: SEE ANDY'S WORK%.

4.3 Training Strategy and Optimization

We implemented several key training strategies to address the challenges encountered during our progress report:

- **Adaptive Learning Rate**: Using TensorFlow's ReduceLROnPlateau callback with factor=0.5, patience=3 epochs

- **Early Stopping**: Implemented with patience=5 epochs to prevent overfitting (as shown in the training curves)

- **Batch Size Adjustment**: Dynamically adjusted batch size based on resolution (224×224 $\rightarrow$ batch size of 16)

- **Loss Function Selection**: Binary crossentropy with sigmoid activation for binary classification

The training process was optimized to balance computational efficiency and model performance. We observed that the MobileNetV2 implementation (Figure 1) achieved a validation accuracy of approximately 89% after just 6 epochs, demonstrating rapid convergence compared to our initial CNN approach. TODO: MAYBE CHANGE THIS?

4.4 Error Analysis Implementation

Our error analysis revealed the following:

- The confusion matrix showed significant false negatives (54 cases), indicating the model struggles with early-stage fires @TODO: FACT CHECK

- Precision for fire detection was 0.912, but recall was only 0.785, highlighting a need to improve sensitivity @TODO: FACT CHECK

- Analysis of misclassified images revealed patterns: * Small-resolution images where fine details were lost * Scenes with low contrast between smoke and background elements * Images containing confounders like clouds or water reflections * Some data points were mislabelled from the dataset

The final model achieved a test accuracy of 82.2% @TODO: FACT CHECK with precision and recall values that balanced well for this critical application (Table 1). This represents significant improvement over our initial baseline while maintaining computational efficiency suitable for real-world deployment scenarios.

5 Evaluation

The evaluation strategy employed a standard 70/15/15 train-validation-test split on a dataset containing approximately 410 images with a 61:39 fire-to-non-fire class distribution based on the ResNet18 confusion matrix totals. The training set (70%) was used for model learning and weight optimization across the full diversity of fire and non-fire scenarios, the validation set (15%) for hyperparameter tuning and early stopping decisions during training, and the test set (15%) for final unbiased performance assessment. Cross-validation was not implemented due to computational constraints of training deep neural networks and the desire to ensure all three architectures (MobileNetV2, ResNet18, and EfficientNet-B0) were evaluated on identical test samples for fair comparison. While the single split approach enabled consistent comparisons and the test set size was sufficient for reliable estimates, future work should incorporate k-fold cross-validation to better quantify performance variance and ensure findings generalize across different data partitions. The evaluation utilized a comprehensive suite of metrics that evolved significantly from the progress report stage, where accuracy and training loss were likely the primary focus. Early analysis revealed these metrics were inadequate—accuracy alone masked critical issues like ResNet18's 68% false positive rate and EfficientNet-B0's catastrophic validation collapse from 63% to 40% accuracy after epoch 6. The refined evaluation framework incorporated precision to quantify false alarm rates (MobileNetV2: 90%, ResNet18: 32%, EfficientNet-B0: low), recall to measure fire detection capability (MobileNetV2: 85%, ResNet18: 80%, EfficientNet-B0: poor), F1-score for balanced assessment, ROC-AUC for threshold-independent performance evaluation (ResNet18: 0.629), confusion matrices for error pattern analysis, precision-recall curves (ResNet18 AP=0.737), threshold sensitivity analysis, training/validation loss convergence tracking, and inference time measurements (MobileNetV2: 180-320ms, ResNet18: 245-2880ms). These expanded metrics proved essential for understanding real-world deployment viability in safety-critical wildfire detection applications. The metric adequacy assessment confirmed that the progress report metrics were insufficient for this application domain. Accuracy failed to distinguish between different error types—critically important in a context

where missing an actual fire has catastrophic consequences (loss of life, property destruction) while false positives merely waste emergency response resources. The comprehensive evaluation revealed that while ResNet18's 80% recall made it attractive from a pure safety perspective, its 32% precision rendered it impractical without significant threshold adjustments. EfficientNet-B0's validation loss explosion and prediction inconsistencies indicated severe overfitting requiring complete retraining. MobileNetV2 emerged as the optimal solution with 88% validation accuracy, balanced precision-recall performance, minimal overfitting, high-confidence predictions (0.82-0.999 range), and deployment-ready stability across all evaluation metrics.

6 Progress

The original plan was to evaluate the current wildfire detection model and identify areas for improvement. We initially focused on three key aspects: efficiency, accuracy, and mobility. Accuracy was the most straightforward area to target, since increasing correct predictions—especially reducing false negatives—is critical for wildfire detection. Efficiency mattered because a faster model reduces resource usage and shortens the time needed to confirm whether a wildfire has begun. Mobility was also important, as we wanted to explore whether the model could eventually be deployed on more portable and cost-effective hardware rather than relying on expensive systems.

To follow through on this plan, we implemented and compared three different models—ResNet18, EfficientNet, and MobileNet—using the same dataset. Each model was chosen because it excels in one of the improvement categories. Based on feedback from my previous progress report, We also incorporated additional evaluation metrics, such as ROC/AUC, to better capture the strengths and weaknesses of each approach.

However, the plan shifted slightly during the process. After running the models, it became clear that more detailed error analysis was needed, especially to address the high rate of false negatives. Because of this, the direction of the project moved from general improvement across multiple categories to a more targeted focus on understanding and reducing misclassifications.

7 Error Analysis

Systematic examination of model errors through confusion matrices, training curves, and qualitative prediction samples revealed distinct failure patterns for each architecture. The ResNet18 confusion matrix showed 108 false positives (68% of non-fire images misclassified as fire) versus only 50 false negatives, indicating the model learned to err on the side of caution by over-predicting fires. Visual inspection of misclassifications revealed that ResNet18 consistently failed on ambiguous atmospheric conditions—sunsets with orange/red coloring (predicted as fire with 0.482 confidence), fog banks resembling smoke, and cloud formations with similar visual characteristics to fire plumes. The model's low confidence scores (0.3-0.6 range) on correct predictions further suggested fundamental uncertainty in its learned features. In contrast, MobileNetV2 demonstrated robust performance across diverse conditions with high-confidence predictions on both fire (0.817-0.999) and non-fire (0.309-0.358 for correct rejections) scenarios, though it occasionally struggled with heavily obscured or distant fires. EfficientNet-B0's error patterns were less systematic and more chaotic, with the validation accuracy oscillating wildly between 40-63% and misclassifying obvious cases, indicating the model failed to learn generalizable features and instead memorized training data. The models exhibited clear performance differences aligned with their architectural characteristics and training stability. MobileNetV2 excelled at distinguishing subtle visual differences between fire-related smoke (gray, billowing, rising patterns) and benign atmospheric effects (uniform fog, wispy clouds), likely due to its stable training convergence where validation and training losses tracked closely together. ResNet18 showed strength in detecting obvious fires with visible flames or dense smoke columns (80% recall) but consistently triggered false alarms on any reddish or orange-tinted imagery, suggesting it over-relied on color features rather than texture and spatial patterns. The threshold analysis revealed this wasn't simply a calibration issue—even at optimal thresholds (0.6-0.7), ResNet18 could only achieve approximately 70% recall with 50-60% precision, indicating fundamental limitations in its learned feature representations. EfficientNet-B0's performance degradation after epoch 6, where validation loss spiked from 0.66 to 0.92 while training loss continued decreasing,

demonstrated classic overfitting where the model memorized training examples rather than learning transferable patterns for fire detection. Pattern analysis across error types revealed several systematic issues requiring targeted interventions. First, all models struggled with "fire-like" non-fire scenarios: sunsets with warm lighting (orange/red sky), dust clouds with similar texture to smoke, and fog in mountainous terrain resembling distant fire smoke. ResNet18's 108 false positives concentrated heavily in these categories, while even MobileNetV2 occasionally misclassified heavily backlit smoke-like formations. Second, the 61:39 class imbalance toward fire instances contributed to ResNet18's bias toward over-prediction, as the model optimized for overall accuracy rather than balanced class performance. Third, inference time variability (ResNet18: 245-2880ms) suggested computational inefficiencies that would impact real-time deployment. To address these issues, future work should: (1) augment the training dataset with challenging non-fire examples specifically targeting failure modes (sunset images, fog, dust, clouds with fire-like colors/textures), (2) implement class-balanced loss functions or focal loss to penalize confident misclassifications and address the dataset imbalance, (3) employ stronger data augmentation including color jittering, atmospheric effects simulation, and lighting condition variations to reduce color bias, (4) implement early stopping around epoch 6-8 for ResNet18 based on validation performance plateaus, (5) explore ensemble approaches combining ResNet18's high recall (80%) with MobileNetV2's high precision (90%) to achieve optimal safety-efficiency balance, and (6) incorporate temporal context from image sequences rather than single frames to leverage fire progression patterns that distinguish actual fires from static atmospheric conditions.

Team Contributions

Tanisha contributed by writing the Introduction and Features (1 & 3) sections of the report. She implemented the preprocessing and augmentation code and the baseline model. She also helped migrate our model training onto the Google Cloud platform. She trained several of the models and also finalized the most optimal combination of pixel size and augmentations used in the experiments.

References

[1] National Aeronautics and Space Administration. 2025. Wildfires and climate change. Web page. Available at https://science.nasa.gov/earth/explore/wildfires-and-climate-change/. Accessed: 2025-11-09.

[2] Bnsreenu. 2023. Malaria binary classification using tensorflow lite. https://github.com/bnsreenu/python_for_microscopists/tree/master/237_tflite_using_malaria_binary_classification. GitHub repository, accessed on 2025-11-10.

[3] Health Canada. 2024. Human health effects of wildfire smoke — report summary. Web document. Available at https://www.canada.ca/en/services/health/healthy-living/environment/air-quality/wildfire-smoke/human-health-effects-report-summary.html. Accessed: 2025-11-09.

[4] I. El-Madafri, M. Peña, and N. Olmedo-Torre. 2023. The wildfire dataset – enhancing deep learning-based forest fire detection with a diverse evolving open-source dataset. https://www.kaggle.com/datasets/elmadafri/the-wildfire-dataset?resource=download. Kaggle dataset, accessed on 2025-11-10.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[6] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[7] Rose Morrison. 2022. The impact of wildfires on biodiversity and the environment.

[8] Laxita Soontha and Mohammad Younus Bhat. 2026. Global firestorm: Igniting insights on environmental and socio-economic impacts for future research. *Environmental Development*, 57. Accessed: 2025-11-10.

[9] Dario Spiller, Andrea Carbone, Stefania Amici, Kathiravan Thangavel, Roberto Sabatini, and Giovanni Laneve. 2023. Wildfire detection using convolutional neural networks and prisma hyperspectral imagery: A spatial-spectral analysis. *Remote Sensing*, 15(19):4855.

[10] Stack Overflow user Ivan Shelonik. 2018. What should be the dimension of image in convolutional neural network? https://stackoverflow.com/questions/48954724/what-should-be-the-dimension-of-image-in-convolutional- Accessed: 2025-12-03.

[11] Stack Overflow user user10024395. 2017. Is there any particular reason why people pick 224×224 image size for imagenet experiments? https://stackoverflow.com/questions/43434418/is-there-any-particular-reason-why-people-pick-224x224-image-size-for-imagenet-e. Accessed: 2025-12-03.

[12] Mingxing Tan and Quoc V Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

[13] TensorFlow Datasets. 2023. Malaria dataset. https://www.tensorflow.org/datasets/catalog/malaria. Contains 27,558 thin blood-smear cell images with parasitized/uninfected labels.

[14] Eleni Tsalera, Andreas Papadakis, Ioannis Voyiatzis, and Maria Samarakou. 2023. Cnn-based, contextualized, real-time fire detection in computational resource-constrained environments. *Energy Reports*, 9:247–257.

[15] Berk Özel, Muhammad S. Alam, and Muhammad U. Khan. 2024. Review of modern forest fire detection techniques: Innovations in image processing and deep learning. *Information*, 15(9):538.