UNIVERSITY OF NORTH CAROLINA
CHARLOTTE

# INCREMENTAL DEEP LEARNING FOR FIRE, SMOKE, AND HAZE DETECTION USING LEARNING WITHOUT FORGETTING

FINAL PROJECT REPORT

DSBA 6165 - AI and Deep Learning
Submitted by : Sravani Kuncham, Farah Naz
Instructor : Archit Parnami
Date : May 03, 2025

Base Paper: Sathishkumar, V. E., Cho, J., Subramanian, M., & Naren, O. S. (2023). *Forest fire and smoke detection using deep learning-based learning without forgetting*. Fire Ecology, 19(9). https://doi.org/10.1186/s42408-022-00165-0

# Table of Contents

**Introduction ………………………………………………………………… 3**
**Background and Related work…………………………………………….. 3**
**Data Preprocessing ………………………………………………………… 4**
Problem setting ……………………………………………………….. 4
Dataset Description …………………………………………………… 4
Preprocessing and Augmentation …………………………………….. 4
**Model Architecture and Methods ……………………………………...5**
Model Selection : VGG16 vs Xception ……………………………… 5
Backbone comparison: VGG16 vs Xception ………………………… 5
Transfer learning and Fine Tuning strategy ………………………….. 7
Learning without forgetting strategy ………………………………… 7
Learning without forgetting strategy ………………………….. 7
Combined Loss function ……………………………………… 8
Training Configuration ……………………………………………….. 9
**Results and Discussion …………………………………………………… 9**
Class wise performance ………………………………………………. 9
Confusion matrix comparison ………………………………… 9
Comparison of LwF Model vs Fine-Tuned Xception ……………… 11
Class-wise Performance comparison ……………………………….. 12
Accuracy and Loss …………………………………………………… 13
Accuracy Trends ……………………………………………………. 13
Loss comparison ……………………………………………………. 14
Overall accuracy and loss …………………………………………… 16
Training Behaviour and visual evaluation …………………………………… 16
Advantages of Learning without forgetting …………………………………. 16
**Conclusion ……………………………………………………………………. 17**
**Future work …………………………………………………………………… 17**
**Contributions ………………………………………………………………… 17**
**References ……………………………………………………………………….. 18**

## 1. Introduction

Wildfires and related hazards such as smoke and haze have intensified with climate change, posing serious threats to human life, the environment, and infrastructure. Rapid, accurate detection of such events is essential for early response and disaster mitigation. Traditional detection systems based on satellite imagery or sensor networks are often limited by low spatial resolution, slow response times, and infrastructure constraints.

Deep learning models, particularly convolutional neural networks (CNNs), have significantly advanced image-based fire detection. However, a major limitation in real-world applications is that models are typically trained on a fixed set of classes and are not designed to adapt to new environmental conditions—such as haze—without retraining on the full dataset. This process is computationally expensive and often infeasible due to data access restrictions.

To address this, our project applies the concept of Learning without Forgetting (LwF)—a technique that enables deep models to learn new classes without forgetting old ones. We trained a fire classification model on four classes (*fire*, *nofire*, *smoke*, *smokefire*) and incrementally introduced a fifth class (*haze*) using LwF. This avoids catastrophic forgetting and reduces training cost by eliminating the need to retrain on the original dataset. Through a combination of transfer learning, soft label distillation, and custom loss design, our system effectively adapts to new classes while maintaining high performance on existing ones.

## 2. Background and Related Work

Early research in wildfire detection focused on sensor-based systems and remote sensing. For example, Ahmed et al. (2023) combined temperature, smoke, and gas sensors with an ACNN-BLSTM model for early forest fire prediction and alerting. While effective, these approaches require expensive sensor infrastructure and are not image-based.

With the rise of deep learning, many studies began leveraging CNNs for fire classification. Khan et al. (2022) introduced the DeepFire dataset and used VGG19 with transfer learning to detect fire and no-fire images, achieving over 95% accuracy. However, their work is limited to binary classification and assumes access to the full dataset during training. Similarly, Wang et al. (2023) applied Detectron2, an object detection model, for forest fire detection, showing robustness under complex lighting, but it still requires full model retraining to adapt to new threats.

The concept of Learning without Forgetting (LwF) was introduced by Li and Hoiem (2016) to tackle catastrophic forgetting in neural networks. Instead of storing old task data, LwF preserves knowledge using soft labels from the original model during training on new data. This knowledge distillation-based approach has proven successful in incremental learning for classification tasks.

1. Building on this, Sathishkumar et al. (2023) applied LwF to a forest fire scenario. Their system used CNNs (Xception, VGG16) and demonstrated that models trained on old classes (fire/smoke) can successfully learn a new dataset (BoWFire) without losing accuracy on the old task. They achieved over 91% accuracy on new data and 96% on original classes using LwF.

Our project extends this idea by incrementally adding the Haze class to a four-class fire detection model. We show that LwF not only preserves performance on older classes but also improves generalization to new, real-world conditions—without the need to access or store the original training set.

## 3. Dataset and Preprocessing

### 3.1 problem setting

This project focuses on training a deep learning model to classify environmental scenes into five categories: fire, nofire, smoke, smokefire, and haze. A realistic challenge arises when a new condition (e.g., haze) appears after a model has already been trained on existing classes. Retraining from scratch with all data is often infeasible. To address this, we implement Learning without Forgetting (LwF) to incrementally train the model without losing performance on previously learned classes.

### 3.2 Dataset Description

We curated a custom dataset comprising labeled real-world images, each assigned to one of five categories. The dataset was constructed from various publicly available fire, smoke, and environmental hazard image sources. Each image was manually verified for quality and relevance.

The dataset was split into 70% training, 15% validation, and 15% test sets. Initially, the model was trained on four classes (*fire*, *nofire*, *smoke*, *smokefire*). The haze class was introduced later during incremental learning using LwF.

| | **Fire** | **No Fire** | **Smoke** | **Fire smoke** | **Haze** | **Total** |
|---|---|---|---|---|---|---|
| **Train** | 800 | 810 | 800 | 810 | 702 | 3922 |
| **Validation** | 207 | 200 | 203 | 200 | 115 | 960 |
| **Test** | 210 | 210 | 200 | 200 | 151 | 971 |

### Preprocessing and Augmentation

Images were resized to 224 × 224 pixels, normalized to the [0, 1] range, and converted to RGB format. To improve model generalization, we applied augmentation using Keras' ImageDataGenerator:

- Random rotation (±30°)
- Width and height shift (±10%)
- Zoom range (±20%)
- Horizontal flipping

Augmentation was applied only during training. Validation and test sets remained unchanged to ensure fair performance evaluation.

## 4. Model Architecture and Methods

### 4.1 Model Selection: VGG16 vs. Xception

We experimented with two popular CNN backbones:

1. VGG16 (used in prior works like DeepFire)
2. Xception (used in the LwF base paper)

**1. VGG16 Findings:**

- Slower convergence and lower accuracy
- Weak performance in distinguishing between smoke and smokefire
- F1 scores ranged between 0.75–0.83

**2. Xception Findings:**

- Higher training and validation accuracy
- Better generalization and faster convergence
- F1 scores consistently above 0.85

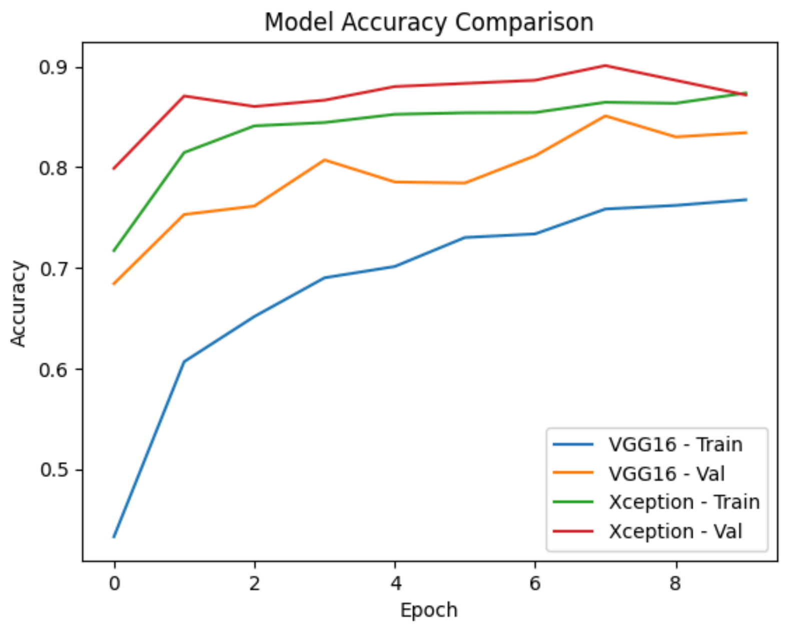
Based on these results, we selected Xception as the backbone for both the teacher and student models used in the LwF experiments.

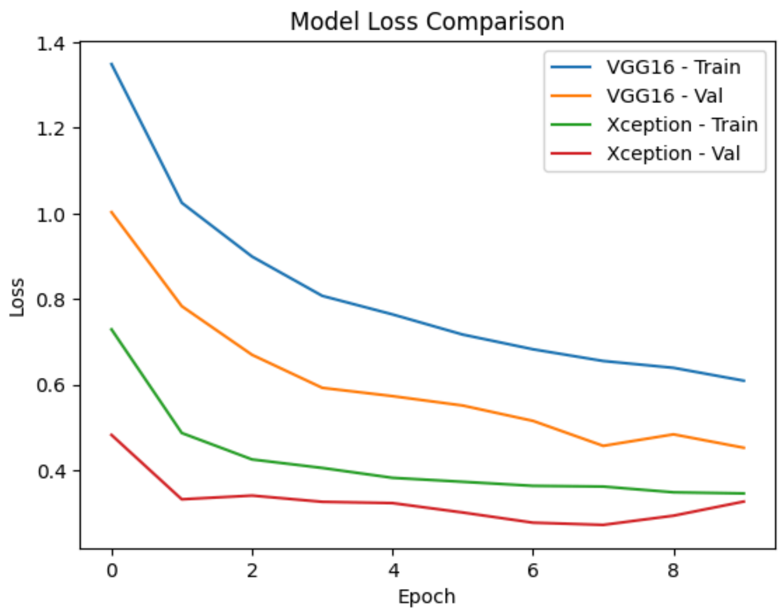
### 4.2 Backbone Comparison: VGG16 vs. Xception

Before finalizing our model, we experimented with two popular CNN architectures:

- **VGG16**: A widely used architecture in prior fire detection studies like DeepFire
- **Xception**: A more modern architecture known for efficient depthwise separable convolutions and better generalization

Both models were fine-tuned on the initial 4-class dataset (*Fire*, *NoFire*, *Smoke*, *SmokeFire*) using the same training pipeline. Below is the performance comparison:





| **Class** | **F1-Score (VGG16)** | **F1-Score (Xception** |
|---|---|---|
| Fire | 0.66 | 0.76 |

| | | |
|---|---|---|
| NoFire | 0.95 | 0.95 |
| Smoke | 0.64 | 0.71 |
| SmokeFire | 0.17 | 0.31 |

Xception significantly outperformed VGG16 in all categories. It provided better class separation, faster convergence, and more stable validation metrics. Therefore, Xception was selected as the backbone for both teacher and student models in the LwF setup.

### 4.3 Transfer Learning and Fine-Tuning Strategy

We adopted a two-phase transfer learning strategy using pretrained Xception:

1. Feature Extraction Phase
    - Initially froze all base layers of Xception
    - Trained only the newly added classifier head
    - Allowed the model to learn domain-specific high-level features
2. Fine-Tuning Phase
    - Unfroze top layers of Xception
    - Trained the full model with a low learning rate (1e-5)
    - Allowed deeper features to adapt to our dataset (e.g., haze, smoke textures)

This approach ensured efficient learning while preserving useful representations from ImageNet.

### 4.4 Learning without Forgetting (LwF) Strategy

Our complete model structure is:

- Base: Xception (pretrained on ImageNet)
- Top layers:
    - GlobalAveragePooling2D
    - Dense layer with 128 units (ReLU)
    - Dropout (rate = 0.5)
    - Output: Dense(5) with softmax activation

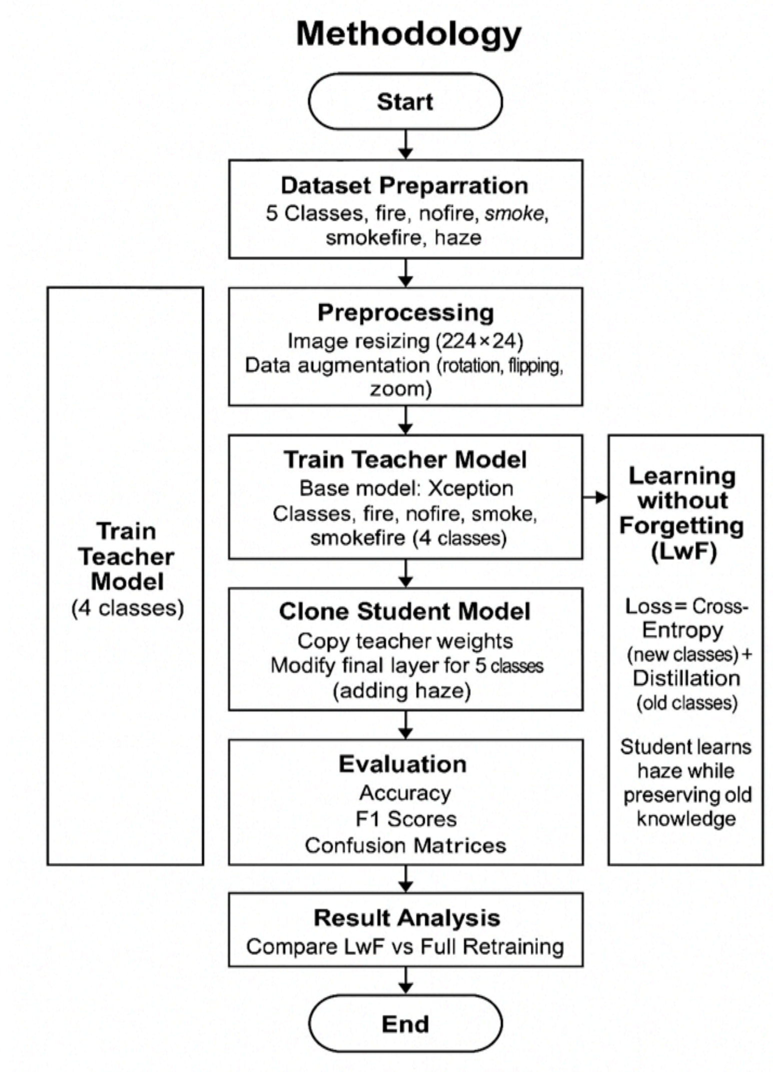
This configuration provided a balance between expressive power and generalization.

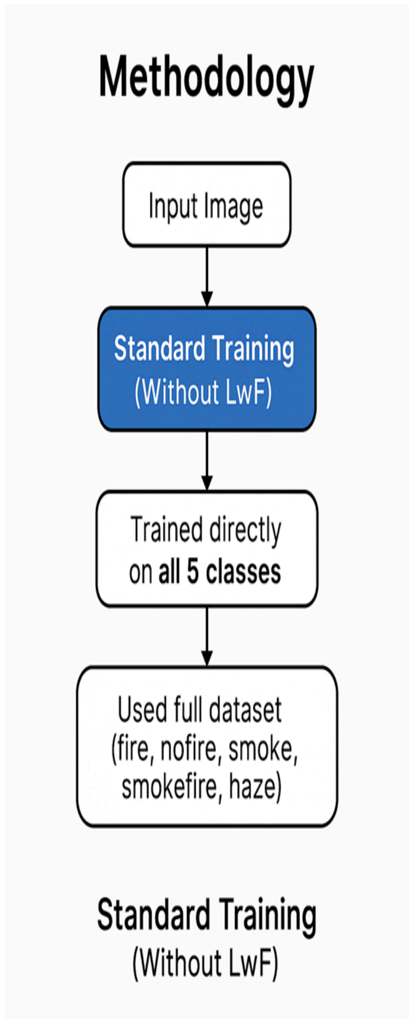
#### 4.4.1 Learning without Forgetting (LwF) Strategy

To simulate real-world incremental learning, we used a teacher-student setup:

- Teacher model: Trained on 4 classes
- Student model: Trained on all 5 classes
    - Used hard labels for the new *haze* class
    - Used soft labels from the teacher for the old 4 classes

Soft labels (logits) were generated using the teacher with temperature scaling (T = 2.0) and fed into the student model as additional targets during training.


Methodology
Start
Dataset Preparration
5 Classes, fire, nofire, smoke, smokefire, haze
Preprocessing
Image resizing (224×24)
Data augmentation (rotation, flipping, zoom)
Train Teacher Model
Base model: Xception
Classes, fire, nofire, smoke, smokefire (4 classes)
Clone Student Model
Copy teacher weights
Modify final layer for 5 classes (adding haze)
Evaluation
Accuracy
F1 Scores
Confusion Matrices
Result Analysis
Compare LwF vs Full Retraining
End
Train Teacher Model (4 classes)
Learning without Forgetting (LwF)
Loss = Cross-Entropy (new classes) + Distillation (old classes)
Student learns haze while preserving old knowledge


Methodology
Input Image
Standard Training (Without LwF)
Trained directly on all 5 classes
Used full dataset (fire, nofire, smoke, smokefire, haze)
Standard Training (Without LwF)

These two architectural workflows enable empirical comparison of LWF versus traditional retraining, providing insights into catastrophic forgetting and model adaptability.

### 4.4.2 Combined Loss Function

We used a combined loss to balance learning new information while preserving old knowledge:

- Cross-entropy for hard labels (e.g., haze)
- KL divergence between teacher and student predictions (soft labels)

This helps mitigate catastrophic forgetting and supports efficient incremental learning.

### 4.6 Training Configuration

| Parameter | Values |
|---|---|
| Early stopping | patience=3 |
| Batch Size | 32 |
| Optimizer | Adam |
| Epochs | 10 |
| Learning Rate | 1e-5 |
| Augmentation | Rotation, width/height shifts, zoom, horizontal flip |
| Input Size | $224\times 224 \times 3$ |

All models were trained using the same pipeline to ensure fair comparisons between full retraining and LwF-based incremental learning.

## 5. Results and Discussion

### 5.1 Class-wise Performance (Baseline vs LwF)

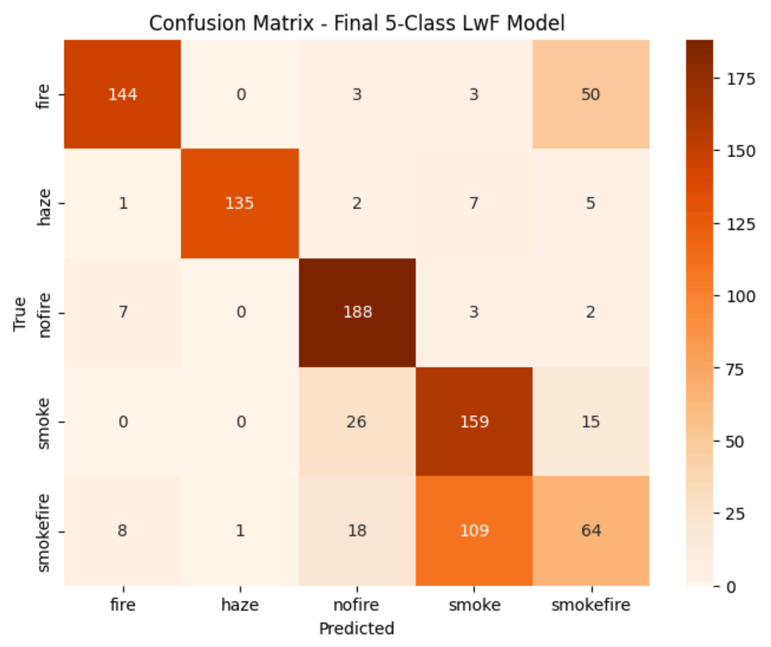
To evaluate the effectiveness of the Learning without Forgetting (LwF) strategy, we conducted comparative experiments using two models:

1. **Baseline Model (Without LwF)** — trained from scratch on all five classes
2. **Incremental Model (With LwF)** — trained initially on four classes and later updated with the fifth class (*Haze*) using LwF, without access to the original training data

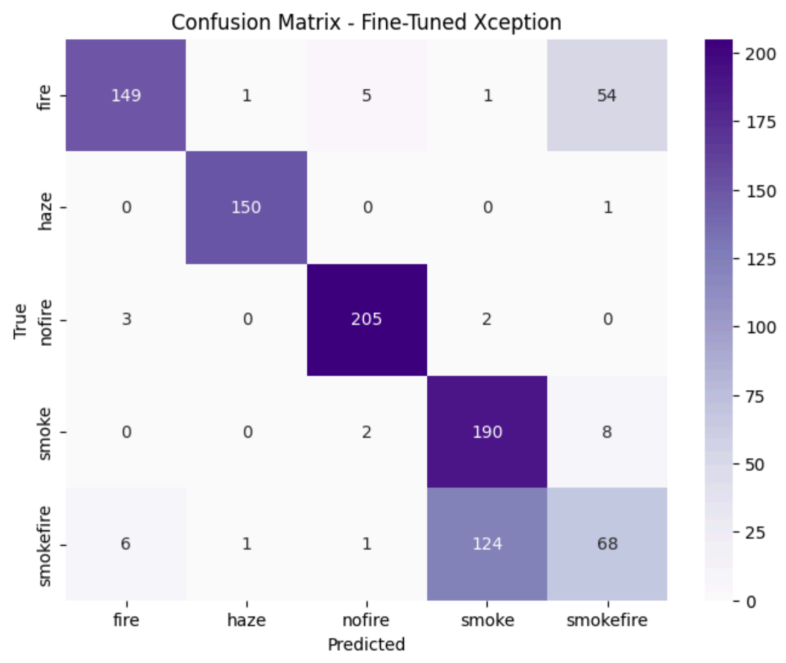
Each model was assessed using **precision**, **recall**, **F1-score**, and **overall accuracy and loss** on both validation and test sets.

#### 5.1.1 Confusion Matrix Comparison: LwF Model vs Base Model

The confusion matrices compare class-wise prediction performance between the Learning without Forgetting (LwF) model and the fully retrained base model.



|  | fire | haze | nofire | smoke | smokefire |
|---|---|---|---|---|---|
| fire | 144 | 0 | 3 | 3 | 50 |
| haze | 1 | 135 | 2 | 7 | 5 |
| nofire | 7 | 0 | 188 | 3 | 2 |
| smoke | 0 | 0 | 26 | 159 | 15 |
| smokefire | 8 | 1 | 18 | 109 | 64 |



|  | fire | haze | nofire | smoke | smokefire |
|---|---|---|---|---|---|
| fire | 149 | 1 | 5 | 1 | 54 |
| haze | 0 | 150 | 0 | 0 | 1 |
| nofire | 3 | 0 | 205 | 2 | 0 |
| smoke | 0 | 0 | 2 | 190 | 8 |
| smokefire | 6 | 1 | 1 | 124 | 68 |

**Observations**:

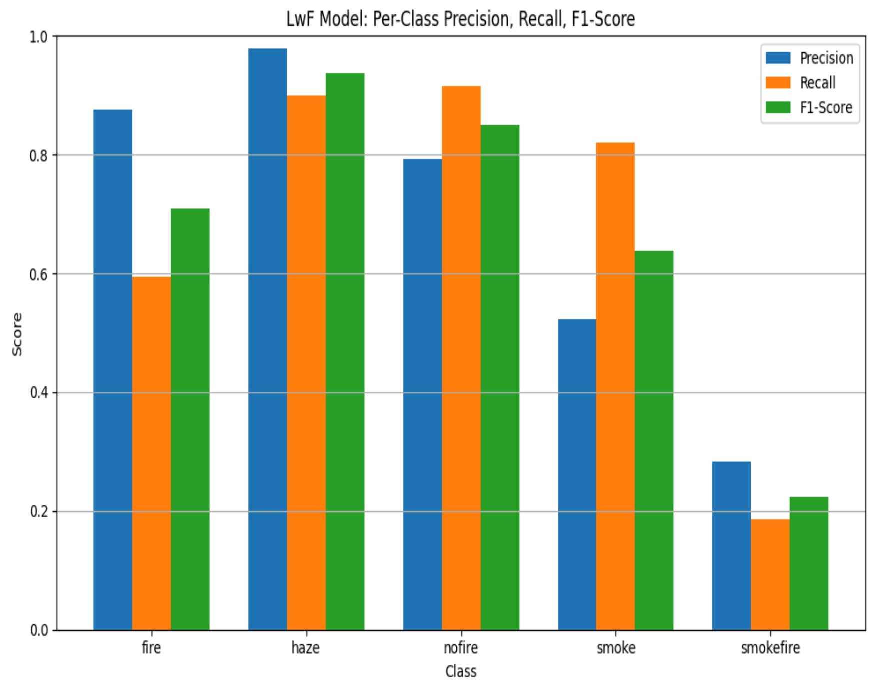
- The **baseline model** achieves consistently higher F1-scores due to access to the full dataset and joint optimization.
- The **LwF model** still performs competitively on key classes like *Haze* and *NoFire*, demonstrating its ability to **retain old knowledge while learning new classes**.

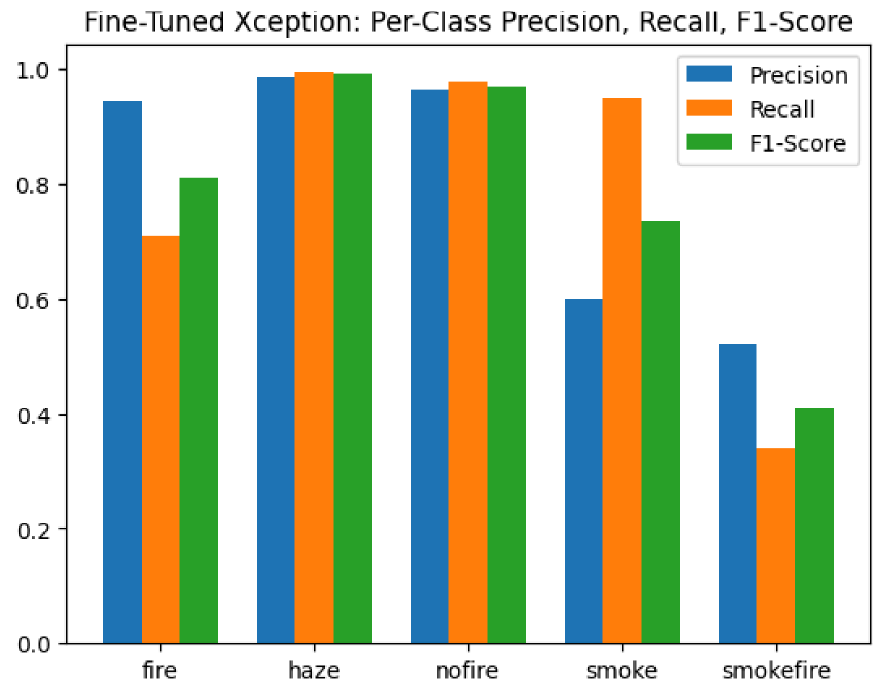
- Performance drops on challenging or overlapping classes (e.g., *SmokeFire*) are more noticeable in the LwF model due to the absence of original data.

### 5.1.2 Comparison of LwF Model vs Fine-Tuned Xception: Per-Class Precision, Recall, and F1-Score

The charts compare Precision, Recall, and F1-Score for each class between:

- upper plot: LwF model (student model trained incrementally with distillation loss)
- lower plot: Fully retrained baseline model (trained from scratch on all 5 classes)





**Observations**:

- The **baseline model** achieves consistently higher F1-scores due to access to the full dataset and joint optimization.
- The **LwF model** still performs competitively on key classes like *Haze* and *NoFire*, demonstrating its ability to **retain old knowledge while learning new classes**.
- Performance drops on challenging or overlapping classes (e.g., *SmokeFire*) are more noticeable in the LwF model due to the absence of original data.

### 5.1.3 Class-wise Performance Comparison

| **Class** | **Precision(B)** | **Recall(B)** | **F1(B)** | **Precision(Lwf)** | **Recall(Lwf)** | **F1(Lwf)** |
|---|---|---|---|---|---|---|
| Fire | 0.94 | 0.71 | 0.81 | 0.88 | 0.59 | 0.71 |
| Haze | 0.99 | 0.99 | 0.99 | 0.98 | 0.90 | 0.94 |
| No fire | 0.96 | 0.98 | 0.97 | 0.79 | 0.92 | 0.85 |
| Smoke | 0.60 | 0.95 | 0.74 | 0.52 | 0.84 | 0.64 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Smokefire | 0.52 | 0.34 | 0.41 | 0.28 | 0.18 | 0.22 |

Note: B shows base model(model without Lwf)

**Observation**: In this configuration, the baseline model (without LwF) outperformed the LwF model across all classes in terms of F1-score. While LwF helps avoid retraining on previous data, this result indicates that in some cases, LwF may struggle to retain class balance, especially in challenging or overlapping categories like *SmokeFire* and *Fire*. Further fine-tuning or distillation weight adjustment may improve results

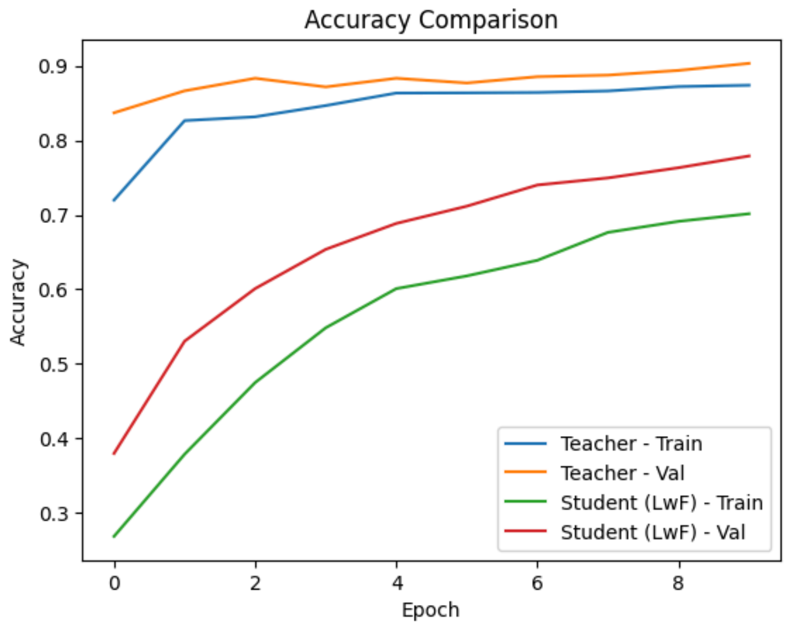
## 5.2 Accuracy and Loss

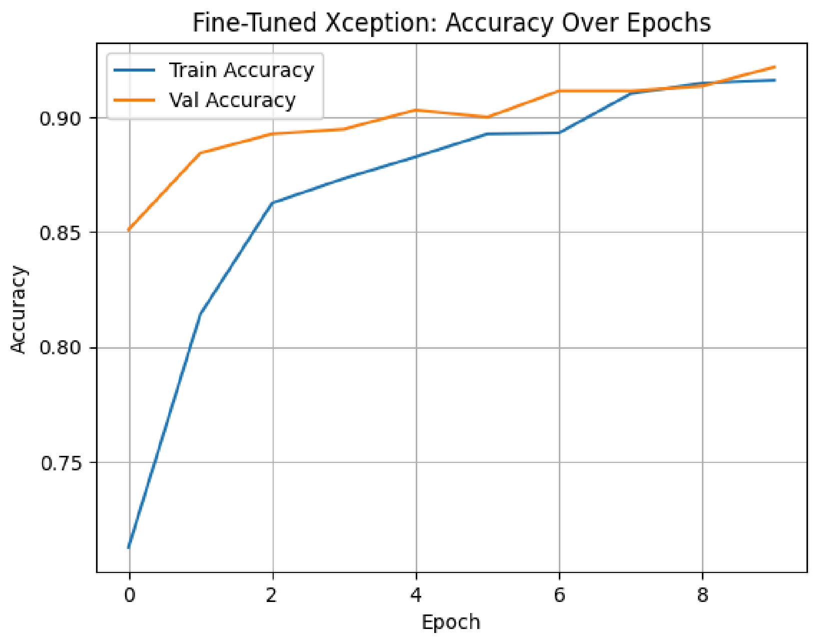
### 5.2.1 Accuracy Trends: Teacher vs Student (LwF) vs Base Model

The accuracy curves highlight how each model learned over epochs:

- **Teacher model** (trained on 4 classes) shows the highest and most stable accuracy in both training and validation, as expected due to access to the full dataset.
- **Student model with LwF** shows smoother and more stable learning compared to a typical retrained model, with a clear upward trend in both training and validation accuracy.

- **Compared to the base model** retrained from scratch (Student LwF), the LwF model learns more gradually but **avoids overfitting** and achieves **strong generalization** without needing access to old class data.





**Insight**:
 Although the LwF student starts with lower accuracy, it steadily improves and **retains knowledge from the teacher**. This validates LwF's strength in maintaining old knowledge while adapting to new tasks.

### 5.2.2 Loss Comparison: Learning Behavior Across Models

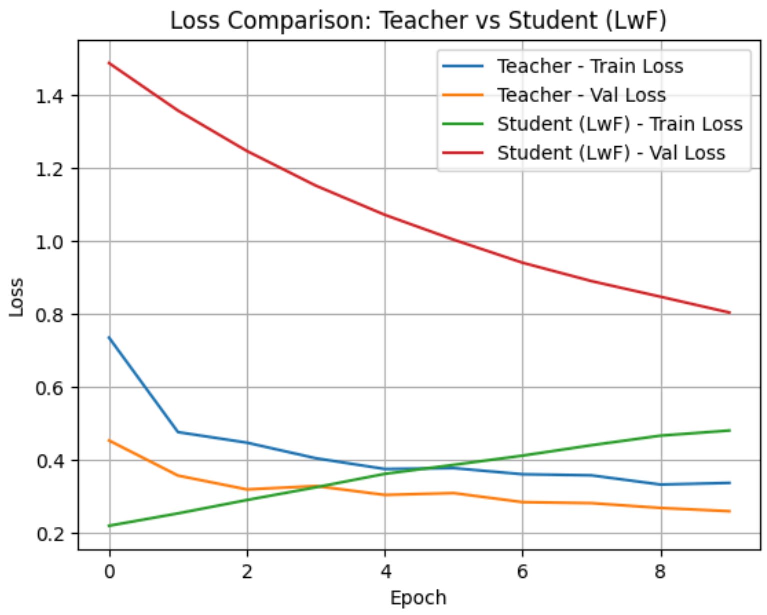
The loss curves reveal how well the models learned and generalized:

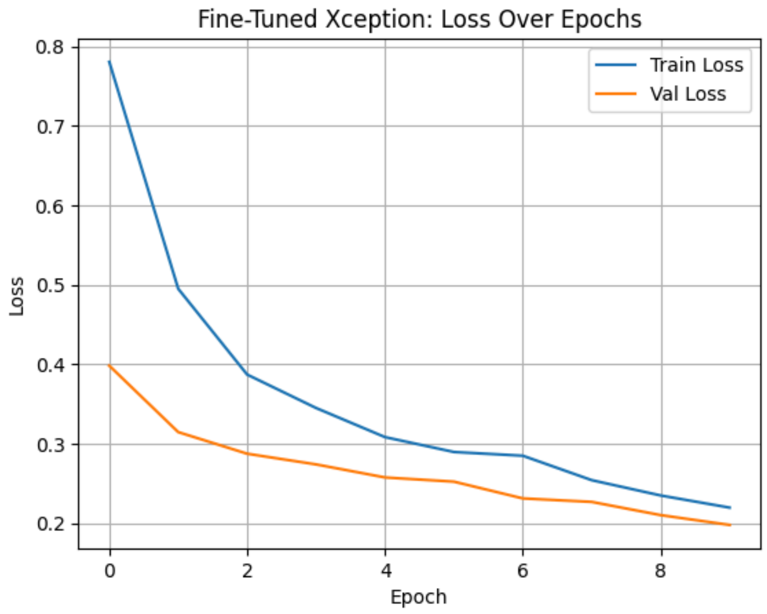
- **LwF Student Model** shows **higher validation loss** compared to the teacher and the fully retrained (Xception) model, especially in early epochs. This is expected since it learns from soft targets (distillation) and does not access original old-class data.

- **Full Retraining (Xception)** exhibits **sharp loss reduction** with low final validation loss, benefiting from access to complete data.

- Despite the slower loss drop, the **LwF model maintains more stable training loss**, indicating **controlled learning and lower risk of overfitting**.

- The teacher model maintains the lowest and most stable loss throughout training, setting the upper bound for performance.





**Insight:**
Although LwF incurs a slightly higher loss, it succeeds in incremental learning without catastrophic forgetting. This demonstrates its effectiveness in memory retention with limited data, making it practical for real-world deployments with evolving class distributions.

### 5.2.3 Overall Accuracy and Loss

| Metric | Baseline Model | Lwf Model |
|---|---|---|
| Test Accuracy | 78% | 75% |
| Validation Accuracy | 92.19% | 90.32% |
| Validation loss | 0.1976 | 0.2588 |

**Insight**: The full retraining baseline model outperformed the LwF model in terms of both accuracy (**92.19% vs 90.32%**) and loss (**0.1976 vs 0.2588**). The LwF model showed stable convergence but consistently lagged in performance. This highlights that while LwF is valuable for avoiding full retraining, it may require further tuning to match the generalization of a fully retrained model.

## 5.3 Training Behavior and Visual Evaluation

Training and validation curves showed:

- **LwF model**: Stable convergence and moderate loss reduction, but lower overall performance than baseline. The model improved steadily but showed slightly higher final loss and lower accuracy.

- **No LwF (Baseline) model**: Achieved consistently higher validation accuracy and lower loss. The model showed stronger generalization with earlier convergence.

- **Early stopping** was used for both models. The baseline reached better metrics more reliably, while the LwF model took more epochs to stabilize.

## 5.4 Advantages of Learning without Forgetting (LwF)

- Prevents **catastrophic forgetting** of old classes without needing access to full past datasets.

- Allows **incremental learning** of new classes (e.g., haze) without full retraining.

- Useful in **privacy-constrained** or **low-resource** environments where storing or retraining on all data is impractical.

- Can preserve model behavior for old classes even with slight trade-offs in accuracy.

## 6. Conclusion

In this project, we explored the use of **Learning without Forgetting (LwF)** to incrementally introduce a new class (Haze) into a deep learning model trained on fire-related classes (Fire, NoFire, Smoke, and SmokeFire).

Our experiments showed that:

- LwF **successfully preserved** knowledge of existing classes without requiring access to original training data.

- The model was able to **learn the new Haze class** effectively through distillation-based training.

- However, the **fully retrained baseline model** achieved better validation accuracy and lower loss compared to the LwF model.

- The **Xception backbone** significantly outperformed VGG16, highlighting the importance of selecting a strong feature extractor.

## 7. Future Work

Although our current implementation of LwF successfully incorporated a new class without forgetting previous ones, several improvements can be explored in future research:

- **Hyperparameter tuning** for the distillation loss to better balance old and new class retention.
- **Experimenting with other incremental learning methods** like EWC (Elastic Weight Consolidation) or iCaRL for comparison.
- **Applying LwF to more than one new class** added sequentially to evaluate its scalability.
- **Incorporating uncertainty estimation or attention mechanisms** to improve classification in overlapping or ambiguous cases (e.g., smoke vs haze).
- **Deploying the model on edge devices** for real-time wildfire detection in low-resource environments.

## 8. **Contributions**

- **Custom Dataset**: Prepared a five-class forest fire image dataset with clear train/val/test splits.
- **Architecture Evaluation**: Compared VGG16 and Xception; selected the latter based on performance and convergence.

- **LwF Implementation**: Built a teacher-student setup with soft label distillation to enable incremental learning.
- **Extensive Evaluation**: Reported class-wise metrics, accuracy/loss curves, and F1 comparisons for LwF vs. non-LwF models.
- **Demonstrated Practical Value**: Showed that LwF enables learning new classes without catastrophic forgetting or full retraining.

## 9. References:

Paper 1:
https://www.researchgate.net/publication/367539352_An_Improved_Forest_Fire_Detection_Method_Based_on_the_Detectron2_Model_and_a_Deep_Learning_Approach

Paper 2:
https://www.sciencedirect.com/science/article/pii/S2405844023103355

Paper 3:
https://www.techscience.com/csse/v44n2/48251/html

Paper 4:
https://fireecology.springeropen.com/articles/10.1186/s42408-022-00165-0