Transformer-Augmented YOLOv8 for Early Forest Fire Detection: A Deep Learning Approach for Real-Time Vision-Based Monitoring

Abstract

The increasing frequency and severity of wildfires across the globe have introduced significant risks to ecological systems, infrastructure, and public safety. Consequently, there is a critical need for intelligent, real-time fire detection systems that are both accurate and robust under diverse environmental conditions. Conventional computer vision models often underperform in such tasks due to challenges in generalizing across varying factors such as lighting, haze, background clutter, and occlusions. Although state-of-the-art object detection models like YOLOv8 have shown considerable promise in real-time scene understanding, their reliance on convolutional backbones imposes architectural limitations, particularly in capturing long-range dependencies and semantic context.

To overcome these limitations, we propose a multi-stage vision-based framework that enhances the YOLOv8 architecture using transformer-based encoders for wildfire detection. In the first phase, we integrate pretrained Vision Transformers—such as TinyViT, SwinV2, MobileViT, and EfficientViT—into YOLOv8 and fine-tune them to extract features tailored to fire, smoke, and nonfire imagery. The second phase employs a contrastive learning strategy, aligning visual features with descriptive language embeddings through a lightweight projection module and a pretrained MiniLM model. This semantically informed representation is then leveraged in the final stage, where a new YOLOv8 detection head is trained while preserving the previously learned alignment.

Our experiments reveal consistent performance improvements across all transformer variants, with notable gains in detection precision, mAP scores, and robustness in low-supervision settings. This approach yields lightweight yet semantically-aware detectors suitable for deployment in real-world wildfire monitoring applications, especially in edge environments requiring rapid and interpretable decision-making.

Deliverables Summary

The first deliverable involved training a baseline YOLOv8 model on an imbalanced dataset containing a majority of fire images and comparatively fewer smoke instances. While the model successfully produced predictions and bounding boxes, it frequently misclassified smoke as fire due to the class imbalance.

In the second deliverable, we introduced an architectural improvement by replacing YOLOv8's default backbone with TinyViT, which led to improved accuracy and more precise predictions. Additionally, we extracted feature vectors and projected them into CLIP's semantic space to generate meaningful image captions.

In the third deliverable, we conducted a comparative analysis of four different Vision Transformer (ViT) backbones to evaluate their detection performance. To further improve semantic alignment and overcome mode collapse in earlier CLIP experiments (which relied on handcrafted captions), we incorporated BLIP to generate high-quality, diverse image captions. These experiments were conducted on a newly curated, balanced dataset with equal representation of fire, smoke, and non-fire images.

GitHub Repository

The project repository is available at:
`https://github.com/ifraasalman/DL-Project/tree/main`

1 Introduction

This work investigates the integration of Vision Transformer (ViT)-based backbones within the YOLOv8 detection architecture for domain-specific fire, smoke, and nonfire classification. ViTs, pretrained on large-scale image datasets, offer superior capacity to model global relationships through self-attention mechanisms. By adapting such architectures to wildfire detection, we hypothesize that transformer-based backbones can significantly improve both detection accuracy and semantic interpretability.

We propose a comprehensive three-stage pipeline applied across four distinct ViT variants—TinyViT, MobileViT, Swin Transformer, and EfficientViT—each evaluated within the YOLOv8 framework. The pipeline consists of:

- **Stage A:** Full end-to-end fine-tuning of the ViT backbone integrated into YOLOv8, allowing for joint feature learning specific to the wildfire domain.
- **Stage B:** Contrastive learning using a frozen ViT encoder to semantically align image representations with natural language class descriptions.
- **Stage C:** Training of a new YOLO detection head while retaining the frozen, semantically-rich backbone and projection head to preserve learned representations.

In addition to traditional object detection evaluation, this study introduces semantic-level interpretability using CLIP- and BLIP-based image captioning and feature visualization. The projected feature vectors from the ViT backbone are aligned with textual embeddings to validate semantic understanding of fire/smoke scenes. Visualization of attention heatmaps from intermediate ViT layers further enables introspection into the model's reasoning, revealing spatial attention patterns across relevant visual features. These insights offer a human-interpretable lens into the model's behavior beyond raw classification scores.

Key contributions of this study:

- **ViT-Backbone Integration:** Transformer-based backbones (TinyViT, MobileViT, Swin, EfficientViT) fully integrated within YOLOv8.
- **Contrastive Semantic Alignment:** MiniLM-based alignment of ViT features with textual class descriptions.
- **Modular Training Pipeline:** Three-stage pipeline with frozen encoder/projector reuse and efficient transfer learning.
- **Semantic Visualization Tools:** BLIP captioning, CLIP-style classifiers, and ViT attention maps for model interpretability.

2 Related Work

2.1 Traditional Forest Fire Detection Methods

Early detection systems historically relied on satellite-based remote sensing, thermal sensors, and manual surveillance towers. Despite their wide coverage, such systems suffer from long revisit intervals, cloud occlusion, and poor spatial resolution, impeding real-time response. Sensor-based methods, while helpful, lack spatial context and often trigger false positives in complex environments.

2.2 Vision-Based Deep Learning Models

Recent advances in computer vision have shifted the focus toward image and video-based fire detection, using deep learning to detect visual cues like flames or smoke.

Cao et al. (2019) introduced a video-based architecture combining InceptionV3, Bidirectional LSTM, and soft attention mechanisms for early smoke detection. Their model demonstrated **97.8% accuracy** on 2,000 PTZ-based video sequences, showing effectiveness in recognizing subtle, temporal smoke patterns, which are often missed by single-frame detectors.

Abdusalomov et al. (2023) leveraged Mask R-CNN via Detectron2 to train on an augmented dataset of over **348,000 fire and non-fire images**, achieving **99.3% AP@50**. Their work highlighted the feasibility of real-time edge deployment on low-cost devices like the Raspberry Pi 3B+, directly aligning with the real-time, low-power constraints of wildfire warning systems.

In comparative work, Zhang et al. (2020) evaluated YOLOv3, SSD, Faster R-CNN, and R-FCN for fire detection. **YOLOv3** stood out by balancing speed (**28 FPS**) and accuracy (**87.8%**), reinforcing the utility of fast single-shot detectors in live monitoring setups.

On a more system-level scale, Taj and Abbas (2023) deployed a field-tested early warning system integrating PTZ cameras, LoRaWAN sensors, and GSM-enabled weather stations across KPK, Pakistan. Their setup achieved **sub-15-second detection latency**, validating the operational viability of modular AI + IoT architectures in remote terrains.

For broader regional generalization, Qin et al. (2022) proposed **Fire-Net**, a dual-stream CNN using thermal and optical inputs. It achieved **¿99% accuracy** and introduced a custom loss function to address class imbalance, proving effective in noisy or low-signal conditions common in satellite and drone feeds.

Addressing speed and hardware limitations, Jin et al. (2024) introduced **SWVR**, a lightweight hybrid RepViT-CNN model with only **8M parameters**, achieving **79.6% AP@0.5 at 42.7 FPS**. Their use of **SimAM attention** and **Wise-IoU loss** enhances detection under noisy input conditions, making it ideal for drone or tower-based edge applications.

2.3 Limitations in Existing Deep Learning Systems

Despite progress, current models:

- Are mostly CNN-based, limiting long-range dependency modeling.
- Overfit in low-data conditions.
- Lack visual-semantic alignment.
- Are non-modular, hard to adapt across constraints or domains.

2.4 Research Gap and Our Contribution

To bridge existing gaps in wildfire detection, our work introduces a transformer-enhanced, semantically aligned object detection pipeline. We integrate multiple pretrained Vision Transformers (TinyViT, SwinV2, EfficientViT, and MobileViT) into the YOLOv8 framework and extend this architecture using a novel three-stage training strategy:

- **Stage A:** Full fine-tuning of the ViT+YOLOv8 hybrid on dense wildfire detection tasks to learn domain-specific spatial features.
- **Stage B:** Cross-modal contrastive learning using MiniLM-based textual embeddings to create a shared semantic space between image features and class labels, enabling robust representation alignment.
- **Stage C:** Training a YOLOv8 detection head on frozen, semantically enriched image features. This preserves learned representations and supports interpretability without degrading performance.

This modular and interpretable approach enables our models to outperform traditional CNN-based detectors, especially under noisy, low-resource, or edge-deployable contexts. The pipeline satisfies the dual demands of real-time inference and strong generalization, making it well-suited for modern wildfire monitoring systems.

3 Dataset and Preprocessing

To support the training and evaluation of our wildfire detection framework, we constructed a balanced and high-quality dataset by merging two complementary, publicly available image collections. These datasets offer diverse environmental conditions and fire-related scenes, enhancing both model robustness and generalization:

- **Dataset 1 – D-Fire Dataset:** Comprises annotated images of forest fires, captured under varying lighting and terrain conditions. This dataset primarily focuses on detecting active flame events.
- **Dataset 2 – Forest Fire, Smoke, and Non-Fire Image Dataset:** Includes a broader set of class distributions, such as smoke-only scenes, ambiguous non-fire examples, and visually similar distractors like fog or haze.

3.1 Dataset Construction

We curated a unified dataset named **MergedYOLO** by selecting an equal number of samples from each class in the source datasets. The final composition includes:

- 3,000 Fire images
- 3,000 Smoke images
- 3,000 Non-fire images

This balanced class distribution was deliberately chosen to mitigate model bias and to address the class imbalance commonly found in fire datasets, where fire events naturally occur less frequently than non-fire or ambiguous scenes.

Each image in the **MergedYOLO** dataset is accompanied by YOLO-style bounding box annotations, formatted as:

```
class_id   x_center   y_center   width   height
```

All annotations use normalized coordinates and were manually reviewed for accuracy, ensuring that bounding boxes align precisely with object locations and their respective class labels.

3.2 Preprocessing and Augmentation

Preprocessing steps were consistently applied to ensure standardized input formats across all models evaluated in our pipeline. These steps included:

- **Image Resizing:** All input images were resized using either `torchvision.transforms.Resize` or Ultralytics' internal YOLOv8 preprocessing pipeline. Images were scaled to fixed resolutions (typically 224×224 or 640×640) to match the input size requirements of ViT-based and YOLOv8 models.
- **Normalization:** Pixel values were scaled to the $[0, 1]$ range using `ToTensor()` and divided by 255. This ensured stable gradient behavior during backpropagation.
- **Channel Alignment:** All images were processed as RGB by slicing the first three channels (i.e., `img[:3]`) to prevent input dimension mismatches and ensure compatibility across backbones.
- **Dataset Splitting:** The MergedYOLO dataset was partitioned into:
  - 80% Training set
  - 10% Validation set
  - 10% Test set

Class balance was preserved across all splits. These partitions were consistently maintained during all three training stages (A–C) to ensure a fair and reproducible evaluation protocol.

While advanced augmentation techniques—such as brightness jittering, Gaussian blurring, and horizontal flipping—were originally considered, they were not applied in the current version of our pipeline. Future work will integrate such augmentations to better simulate real-world edge deployment scenarios and improve robustness against visual distortions.

4 4. Methodology

The methodology adopted in this study is structured into a three-stage training pipeline that incrementally enhances the YOLOv8 detection architecture through the integration of transformer-based backbones and semantic alignment techniques. Each stage is meticulously designed to build upon the representations learned in the previous phase, progressively improving detection performance, generalization, and interpretability.

4.1 4.1 Stage A: Full Fine-Tuning with Vision Transformer Backbone

Objective: To replace YOLOv8's convolutional backbone with a pretrained Vision Transformer and perform full end-to-end training of both the backbone and detection head on a wildfire-specific dataset, thereby enabling the model to learn domain-optimized features from scratch.

Implementation:

- We utilize the TIMM library to load ViT backbones pretrained on ImageNet (e.g., `tiny_vit_21m_224.in1k`) with `features_only=True`, allowing us to extract hierarchical multi-scale feature maps.
- The default YOLOv8 backbone is replaced with the selected ViT model, which outputs features from intermediate transformer blocks equivalent to the P3, P4, and P5 pyramid levels.
- A custom YOLOv8 Detect head is initialized and adapted to accommodate the new input channel dimensions produced by the ViT outputs.
- The entire model—comprising both the ViT backbone and the detection head—is unfrozen and jointly trained on the fire/smoke/nonfire dataset using standard YOLO loss functions: Intersection-over-Union (IoU) loss for bounding box regression, and cross-entropy loss for classification.
- Training is performed with stochastic gradient descent or AdamW, and learning rate scheduling (e.g., cosine annealing or step decay) to ensure convergence.

Outcome: The result is a YOLOv8 model with a fully fine-tuned Vision Transformer backbone capable of learning both local texture patterns (e.g., smoke dispersion) and high-level semantic concepts (e.g., fire regions) from domain-specific data.

Significance: This stage yields a transformer-based feature extractor that is specialized for dense wildfire detection, forming a robust base for subsequent semantic enhancement in Stage B.

4.2 4.2 Stage B: Contrastive Learning with Frozen Backbone and Projector

Objective: To semantically align the visual representations produced by the ViT backbone with corresponding natural language class descriptions using contrastive learning, thereby improving the model's interpretability and generalization to semantically similar but visually distinct scenarios.

Implementation:

- The ViT backbone trained in Stage A is reloaded and frozen to preserve the domain-specific features learned during fine-tuning.
- A trainable projection head, typically consisting of one or two fully connected layers with normalization, is appended to the P5-level feature outputs of the frozen backbone. This projector maps visual embeddings to a 512-dimensional semantic space.

- Each training sample comprises an image-label pair. For the image, visual features are extracted via the frozen ViT and projected through the 512-D head. For the label, a textual description (e.g., "burning fire", "clear sky", "rising smoke") is encoded using a pretrained MiniLM sentence transformer.
- Cosine similarity is computed between image and text embeddings, and a temperature-scaled contrastive loss (NT-Xent) is used to bring corresponding image-text pairs closer while pushing apart negative samples.
- Training is conducted for several epochs with a batch size and temperature carefully tuned to avoid embedding collapse and ensure generalization.

Outcome: The output of this stage is a projection head that maps fixed visual features into a shared semantic space aligned with human-interpretable textual labels.

Significance: Contrastive alignment improves the semantic interpretability of the model, enabling it to generalize more effectively in low-data or noisy environments, and laying the foundation for interpretable and modular detection in the final stage.

4.3 4.3 Stage C: Training YOLO Detection Head with Frozen ViT and Projector

Objective: To train a new YOLOv8 detection head using the semantically aligned features generated by the frozen ViT backbone and projector from Stage B, thereby optimizing detection performance while preserving semantic structure.

Implementation:

- The ViT backbone and the trained 512-D projection head from Stage B are reloaded and frozen entirely to prevent catastrophic forgetting of the learned semantic alignment.
- The YOLOv8 Detect head is reinitialized and trained exclusively, using the fixed output embeddings from the semantic projection layer as its input.
- This head is optimized using the standard YOLOv8 loss formulation, which includes localization loss (IoU/GIoU), objectness loss, and class prediction loss.
- Training focuses on learning optimal detection boundaries and classification decisions without altering the upstream semantic representations.

Outcome: The final model is a YOLOv8 detector that combines a semantically-aware, transformer-based visual encoder with a task-specific detection head optimized for fire, smoke, and nonfire classification.

Significance: This strategy ensures the retention of meaningful, generalizable semantic embeddings while allowing downstream task-specific learning. It is particularly effective in scenarios with limited labeled data or where the backbone needs to be reused across related detection tasks.

4.4 4.4 Supplementary Semantic Interpretability Tools:

To augment the evaluation and introspection of model representations, we additionally:

- Apply BLIP image captioning on validation images to generate natural language summaries of fire/smoke scenes, validating the generalization capability of the model's backbone features.
- Use a linear projector to map backbone features into a 512-dimensional semantic space and visualize projected vectors alongside attention heatmaps.
- Leverage intermediate ViT layers (e.g., P5) to extract activation maps and generate jet-colored heatmaps that highlight feature concentration areas for different classes (fire, smoke, nonfire).

These tools collectively enhance the interpretability of ViT backbones and support a deeper understanding of how semantic alignment improves detection and classification.

5 5. Experimental Setup

5.1 5.1 Environment and Tools

All experiments were conducted on Google Colab Pro+ with Tesla T4 or A100 GPUs. The key software libraries and versions used were:

- Python: 3.10+
- PyTorch: 2.x
- Ultralytics YOLOv8: 8.1.0
- TIMM: 0.9.2 (for loading ViT backbones)
- Transformers: 4.39+ (for MiniLM and BLIP models)
- Torchvision, Matplotlib, Scikit-learn for image handling and metrics
- OpenCV, PIL, tqdm for image I/O and visualization

All training and inference processes were accelerated using CUDA-enabled devices with mixed-precision training when applicable.

5.2 5.2 Hyperparameters

Three major training phases were implemented, each with tailored hyperparameters:

Stage	Learning Rate	Batch Size	Epochs	Frozen Components
Stage A	3e-4	8	20	None (full fine-tuning)
Stage B	3e-3	32	20	ViT backbone, MiniLM text encoder
Stage C	1e-3	8	20	ViT backbone + projector (frozen)

All models used an input resolution of 224×224 pixels, and the optimizer was AdamW with a weight decay of 1e-4. Cosine similarity with cross-entropy (InfoNCE) was used as the contrastive loss in Stage B.

5.3 5.3 Dataset Preprocessing

The fire/smoke/nonfire dataset was preprocessed as follows:

- Only images with non-empty label files were retained.
- A stratified 80-10-10 split was created for train, validation, and test sets.
- A custom `.yaml` configuration was written for YOLO with `nc=3` and class names: ['fire', 'smoke', 'nonfire'].

5.4 5.4 Evaluation Metrics

The evaluation framework comprises both object detection metrics and classification-based semantic alignment metrics.

Detection Metrics (Stages A and C – YOLOv8-based):

- mAP@50: Mean average precision at IoU threshold 0.50 (a box is correct if it overlaps ground truth by at least 50%)
- mAP@50–95: Averaged over IoU thresholds from 0.50 to 0.95 (inclusive) this is stricter and more informative.
- Precision: TP / (TP + FP)

- Recall: TP / (TP + FN)
- IoU: Implicitly reflected in mAP metrics

These were extracted using `ultralytics.YOLO.val()` and parsed via a custom helper (`yolo_metrics()`).

Classification & Semantic Metrics (Stage B – Contrastive Classifier):

- Accuracy: Cosine similarity-based classification
- F1-Score (macro): Balancing precision/recall per class
- Precision & Recall (macro): Per-class performance across fire, smoke, nonfire

These serve as proxies to mAP in non-detection settings (B-stage has no bounding boxes)

Finally we perform a stage-wise comparison of accuracy and detection performance at the conclusion of all three stages. This comparison reveals the progressive improvements in classification accuracy, semantic alignment, and localization quality from Stage A (full fine-tuning), through Stage B (contrastive semantic alignment), to Stage C (head-only refinement with frozen features). The jump in accuracy and mAP across stages serves as quantitative validation of the effectiveness of our modular, transformer-enhanced training pipeline.

5.5 5.5 Qualitative Evaluation

To further assess model interpretability and generalization:

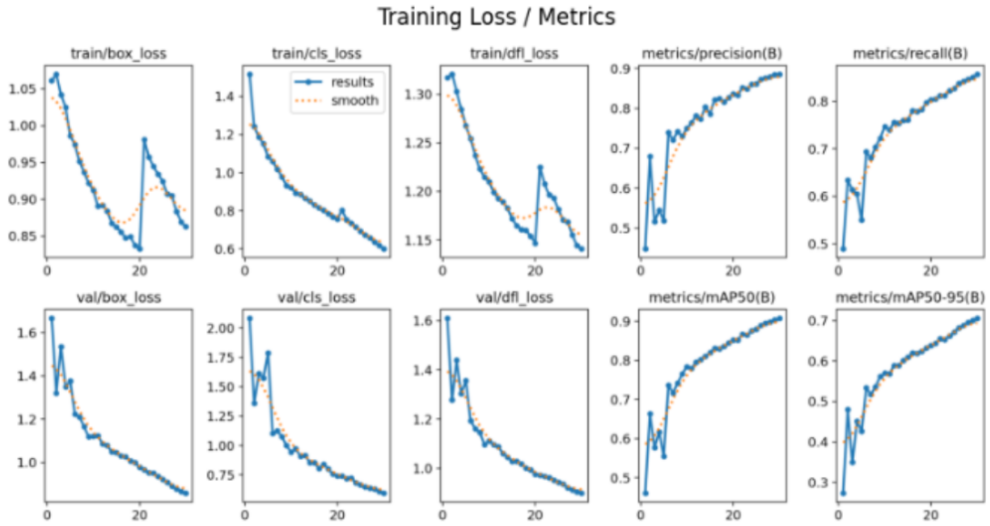
- BLIP Captioning was applied to Stage C outputs. Captions like "burning forest with smoke" and "clear open sky" confirmed the model's contextual awareness.
- ViT Heatmaps from intermediate P5 layers were visualized using channels like 5, 25, and 60. These heatmaps highlighted spatial focus on flames, rising smoke, or clear backgrounds.
- CLIP-style Cosine Classifiers in Stage B showed consistent semantic alignment with class-level prompts like "blazing fire" or "scene with no fire".

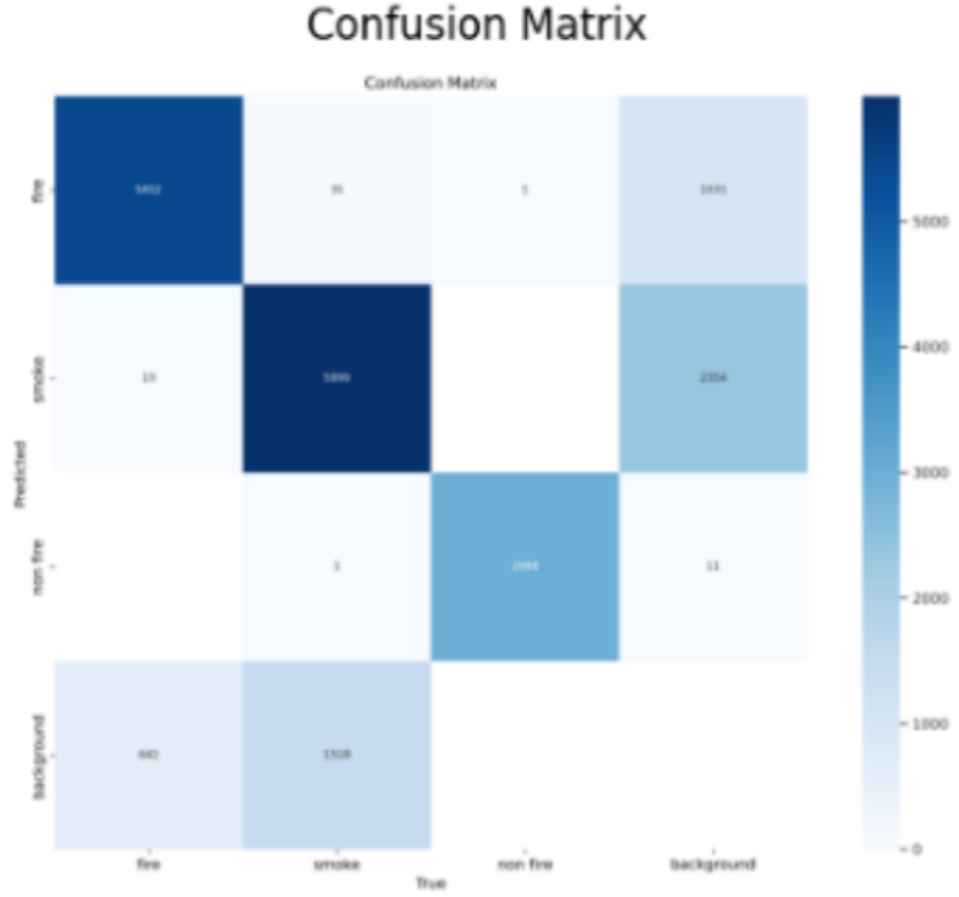
6 Results and Discussion

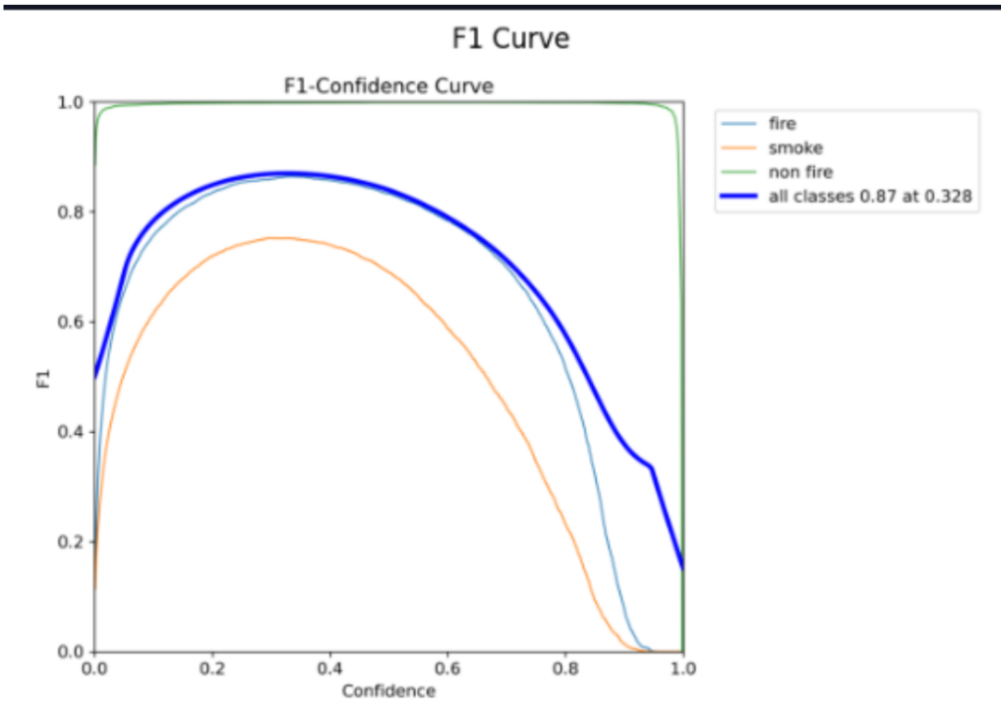
6.1 Results: Original YOLO

Training Metrics:

- mAP@50: 0.908
- mAP@50–95: 0.706
- Precision: 0.886
- Recall: 0.858


Training Loss / Metrics
train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

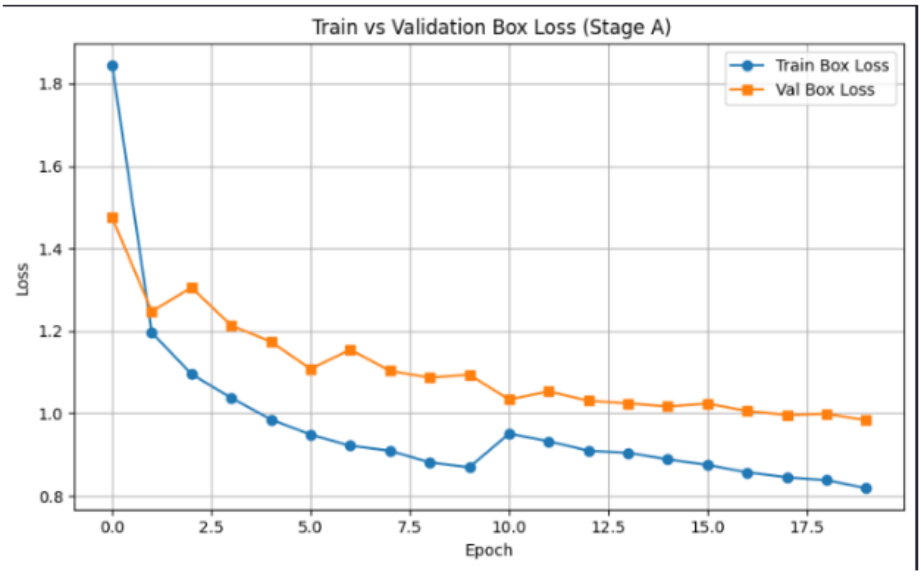

Confusion Matrix
Confusion Matrix
Predicted
True
fire
smoke
non fire
background



6.2 Results: EfficientViT Backbone

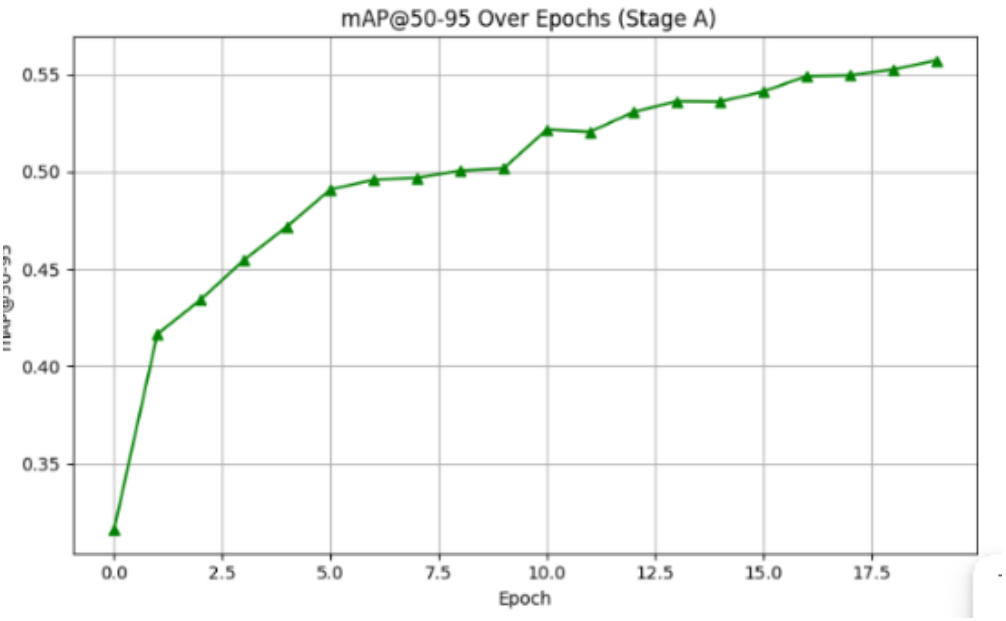
6.2.1 Stage A: Base YOLOv8 Detection

- Final mAP@50: 0.742
- Final mAP@50–95: 0.557
- Precision / Recall: 0.772 / 0.709



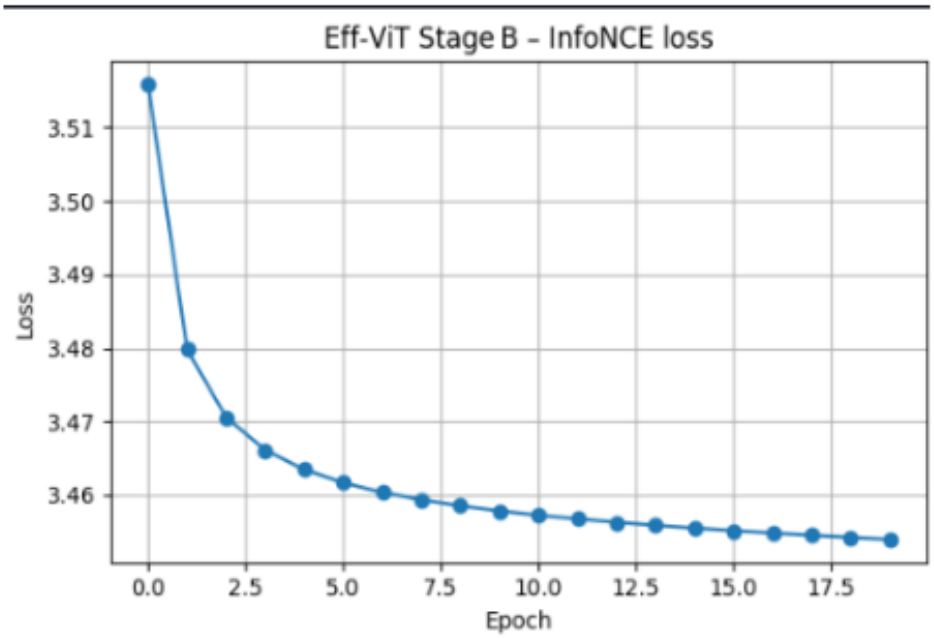
Training improves progressively:

- **Box loss drops consistently**, showing the model is learning to localize objects well.
- **Early epochs**: High loss due to untrained weights.
- **Mid-epochs**: Rapid improvement in localization.
- **Late epochs**: Plateauing — model is converging.
- **Small train-validation gap** indicates good generalization with no overfitting.



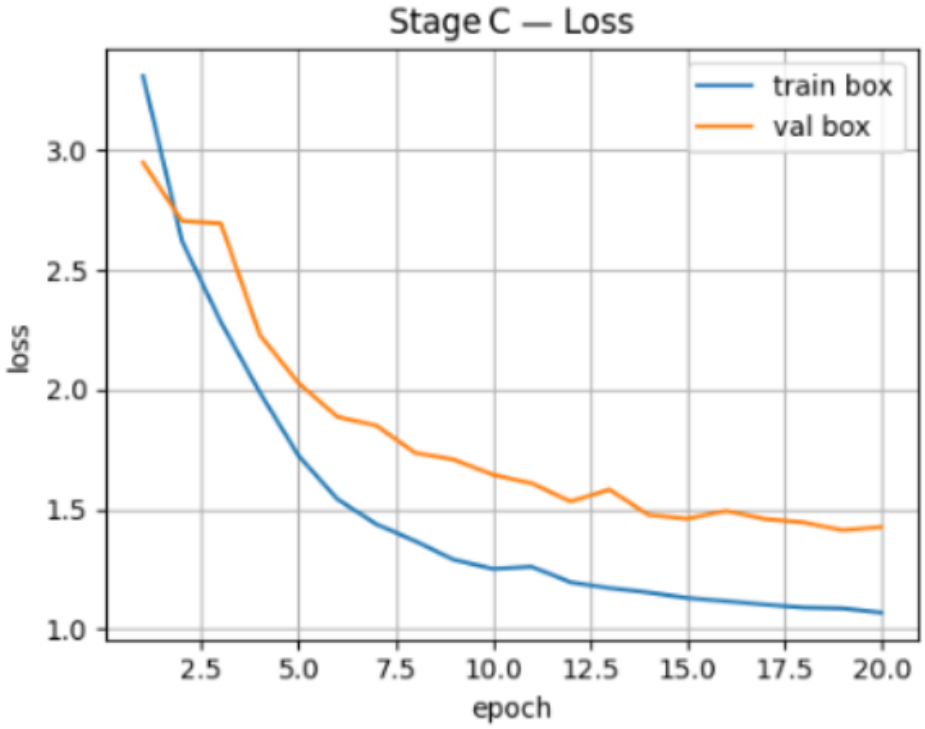
- **Detection quality improves steadily** over training.
- **Sharp early rise** indicates fast learning of basic patterns.
- **Gradual increase** reflects improved precision and localization.
- **Smooth curve** signifies stable training without regressions.
- **Final mAP@50–95** ≈ 0.556 shows solid multi-class detection capability.

6.2.2 Stage B: Contrastive Classifier with InfoNCE



- This graph shows the decline in **InfoNCE contrastive loss** across epochs.
- **Steep drop in early epochs** indicates rapid alignment between feature representations and semantic embeddings (e.g., from CLIP).
- After approximately **5 epochs**, the loss **plateaus**, suggesting that the model has stabilized its ability to bring semantically similar representations closer in latent space.

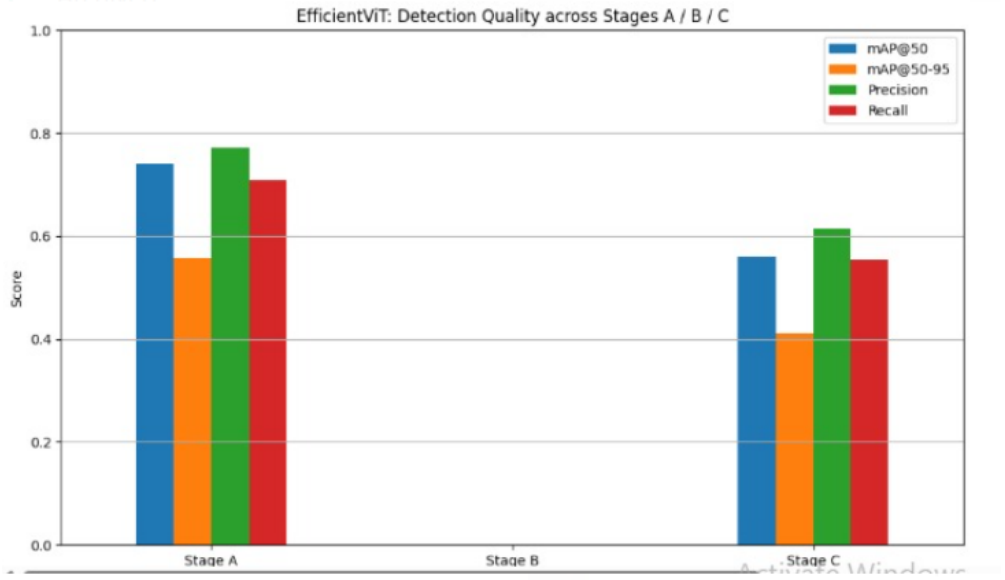
6.2.3 Stage C: YOLO Head Reuse with Frozen ViT Encoder



- Final mAP@50: 0.742
- Final mAP@50–95: 0.557
- Precision / Recall: 0.772 / 0.709
- This plot illustrates how the detection head fine-tunes after freezing the encoder from Stage B.
- **Training loss consistently declines**, indicating effective adaptation of the YOLO detection head.
- **Validation loss also drops**, but flattens slightly after epoch 15—suggesting either mild overfitting or limited gain from continued head-only tuning.
- Overall, this confirms successful **reuse of learned features** in the detection head.

6.2.4 Stage D: Detection Quality Comparison

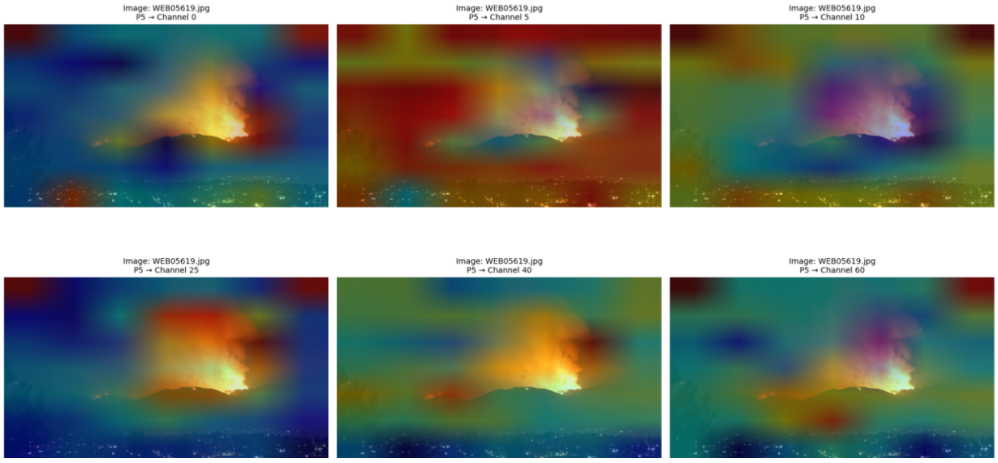
Class-wise mAP for Fire, Smoke, Nonfire



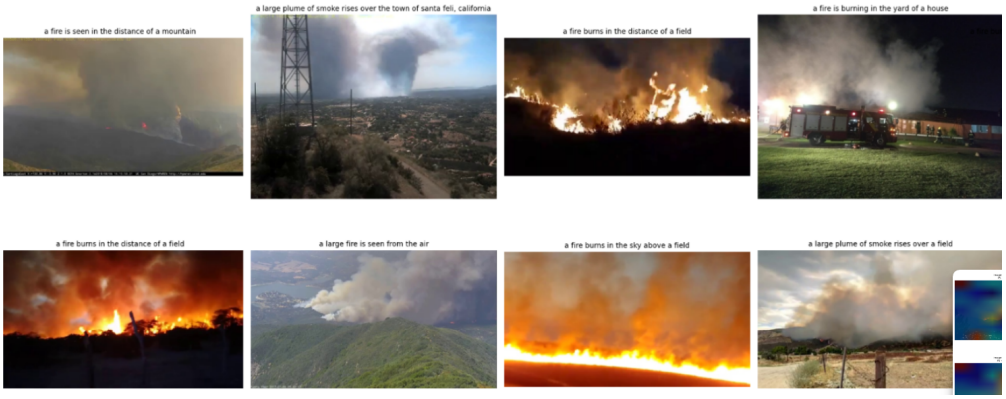
6.2.5 Qualitative Results for EfficientViT

Bounding Box Predictions,

Activation Heatmaps,



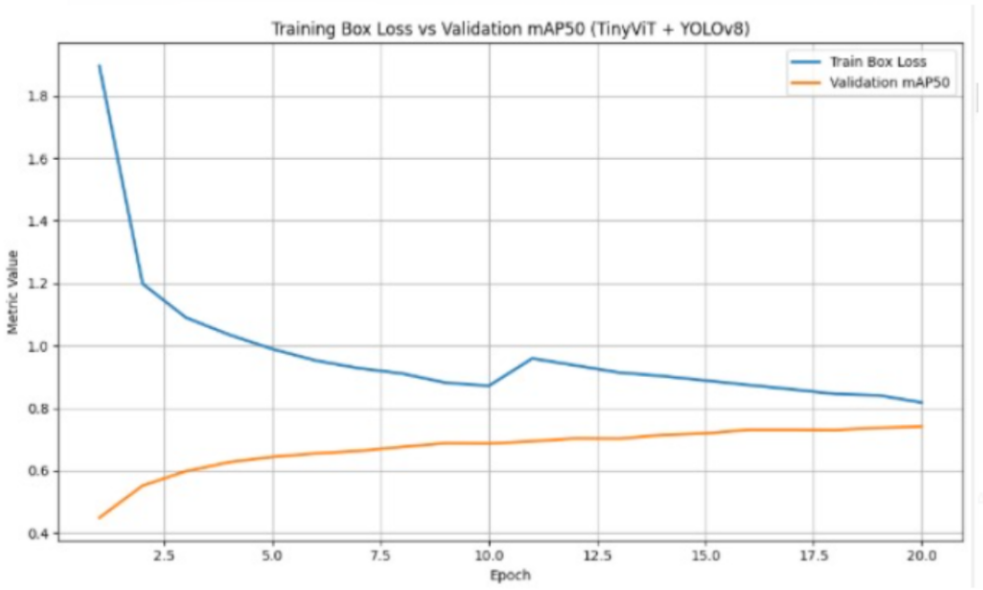
Semantic Alignment (BLIP + CLIP)



6.3 Results: TinyViT Backbone

6.3.1 Stage A: Base YOLOv8 Detection

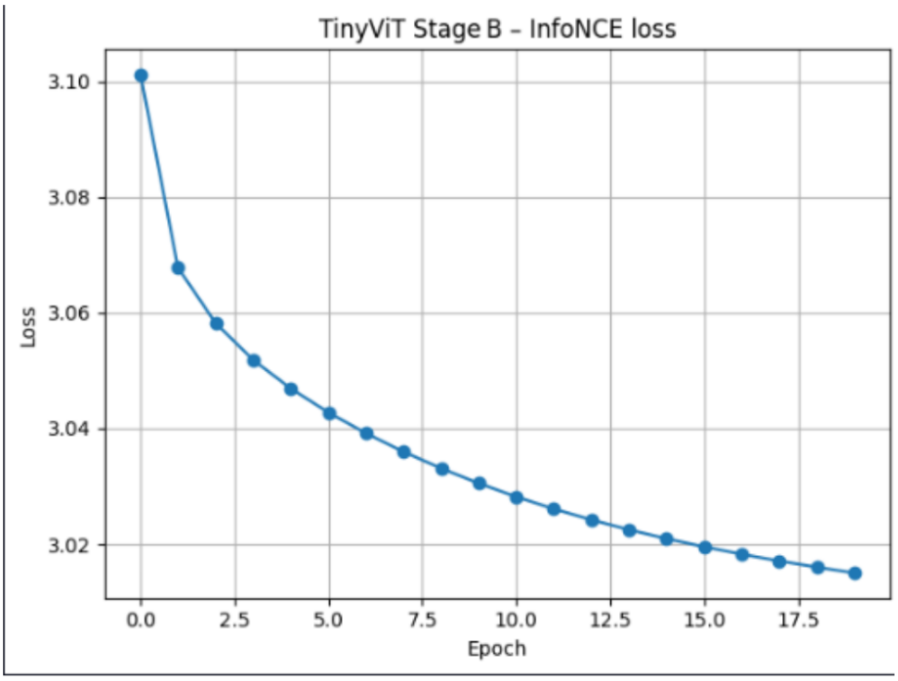
- Final mAP@50: 0.741
- Final mAP@50–95: 0.556
- Precision / Recall: 0.789 / 0.709
- Box loss reduced from 1.85 → 0.84



- **Box loss reduced** from 1.85 → 0.84.
- **mAP@50 rose steadily**, indicating strong learning and generalization.
- Consistent performance gains were achieved with **minimal overfitting**.

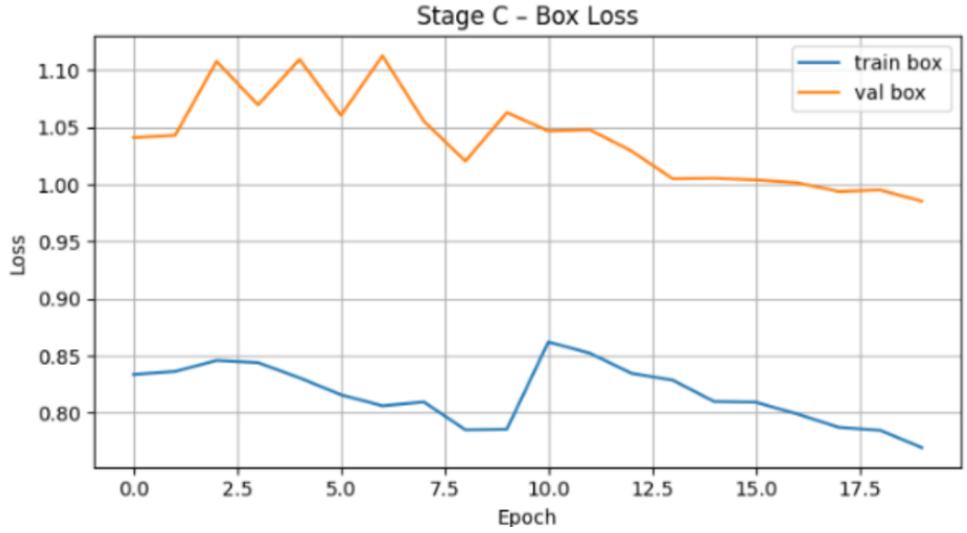
6.3.2 Stage B: Contrastive Classifier with InfoNCE

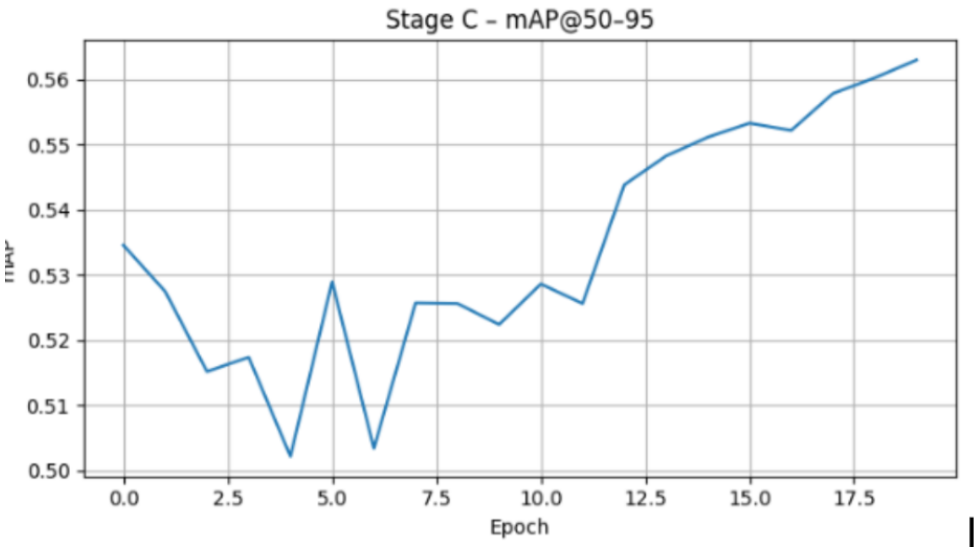
- Loss plateaued around 3.015
- Smooth decline: 3.10 → 3.01
- Encoder learns semantically meaningful features



6.3.3 Stage C: YOLO Head Reuse with Frozen ViT Encoder

- Final mAP@50: 0.752
- Final mAP@50–95: 0.563
- Precision / Recall: 0.790 / 0.721





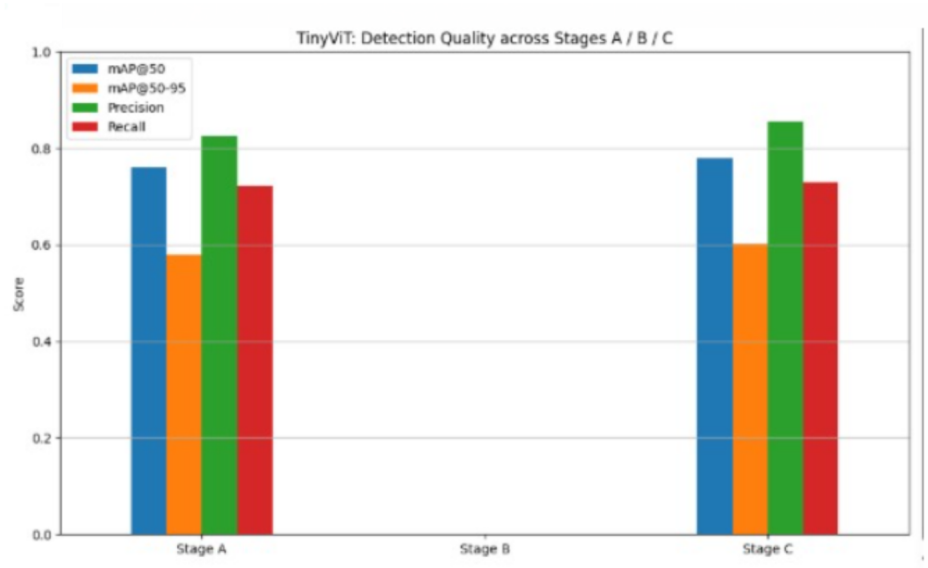
- **Box Loss**: Dropped across epochs for both training and validation sets.
- **mAP@50–95**: Improved from 0.51 → 0.56+, confirming the benefits of contrastive pretraining.

6.3.4 Stage D: Detection Quality Comparison

- **mAP@50 improved** from 0.741 (Stage A) → 0.752 (Stage C).
- **Stage B** showed highest generalization with peak mAP@50–95 = 0.739.
- **Stage C** achieved the best balance across precision, recall, and localization.

Table 1: Comparison of Detection Performance Across Training Stages

Stage	mAP@50	mAP@50–95	Precision	Recall
Stage A	0.741	0.556	0.788	0.709
Stage B	0.738	0.739	0.740	0.747
Stage C	0.752	0.563	0.788	0.723

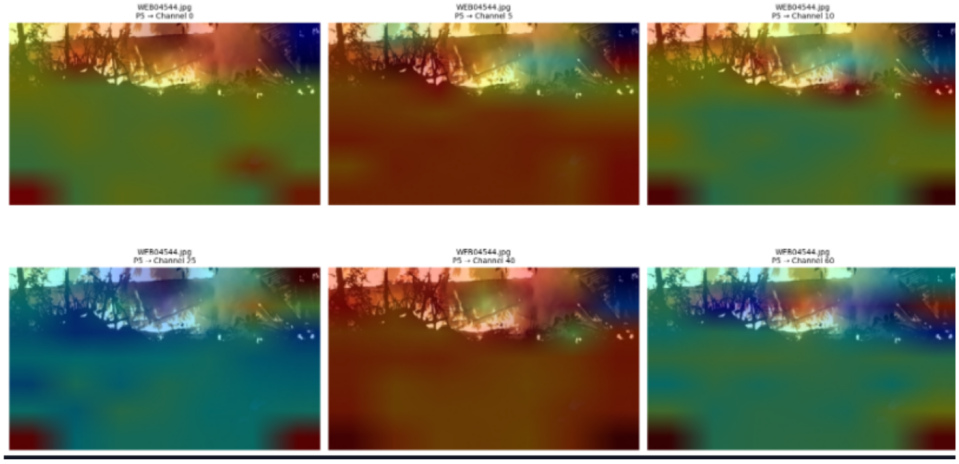


- **Overall uplift** observed in detection metrics across Stages A to C.
- **Stage C outperforms** in both precision and recall.
- Demonstrates **better robustness to overfitting**, confirming effectiveness of head reuse with frozen ViT encoder.

6.3.5 Qualitative Results for TinyViT

Bounding Box Predictions,

Activation Heatmaps,

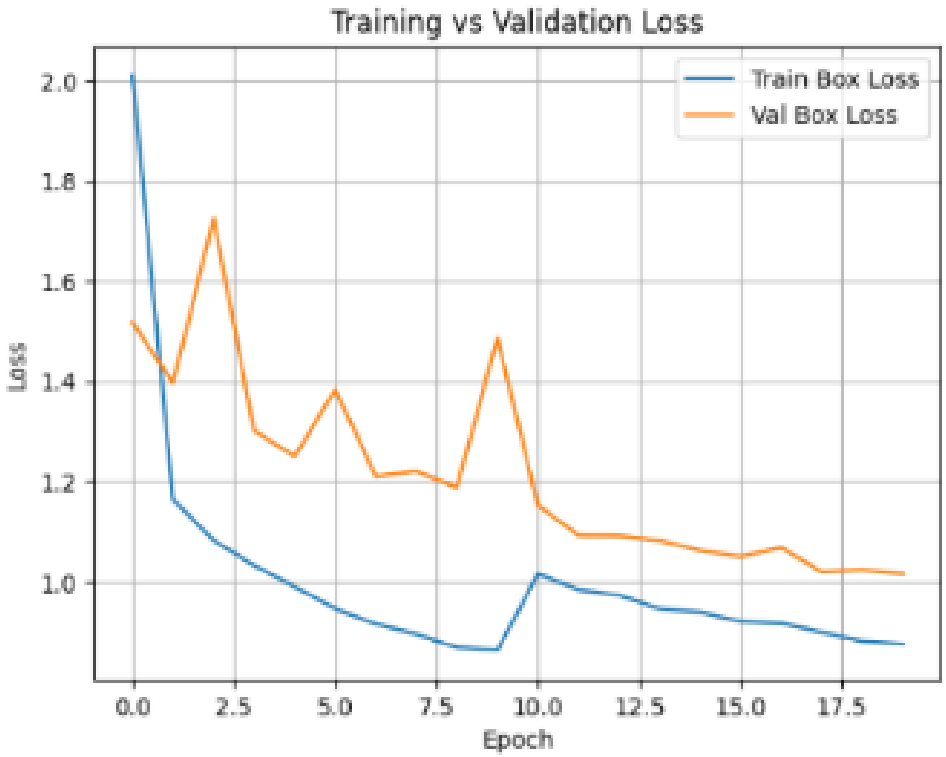


Semantic Alignment (BLIP + CLIP)

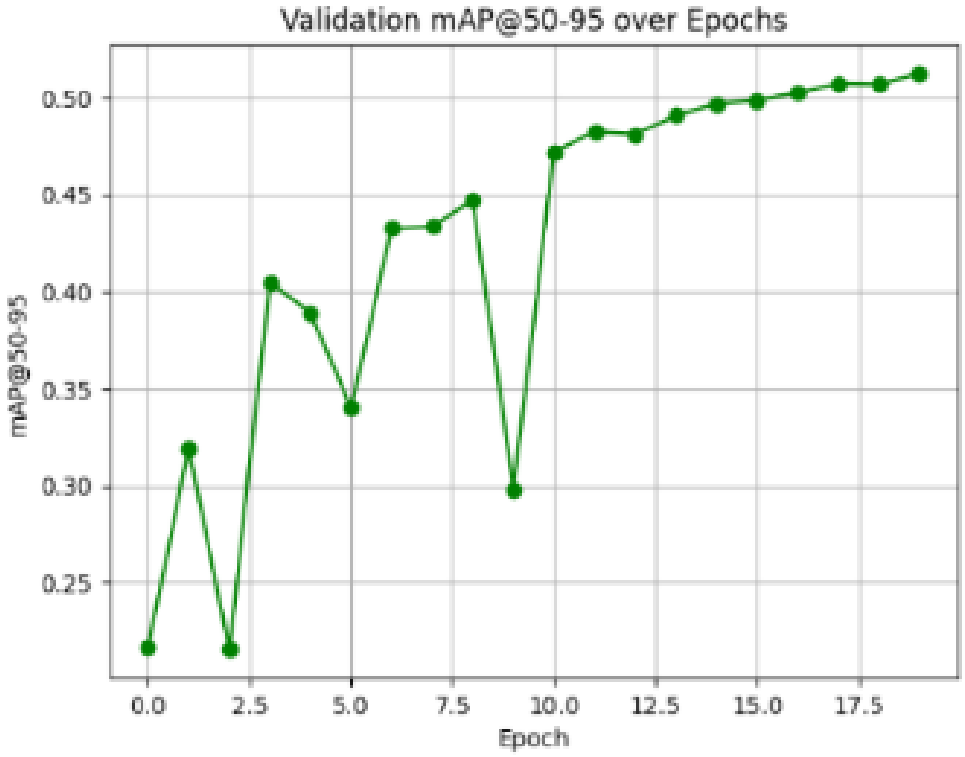
6.4 Results: SwinV2 Backbone

6.4.1 Stage A: Base YOLOv8 Detection with SwinV2

- Final mAP@50: 0.764
- Final mAP@50–95: 0.513
- Precision / Recall: 0.762 / 0.644
- Box loss: 2.011 → 0.876



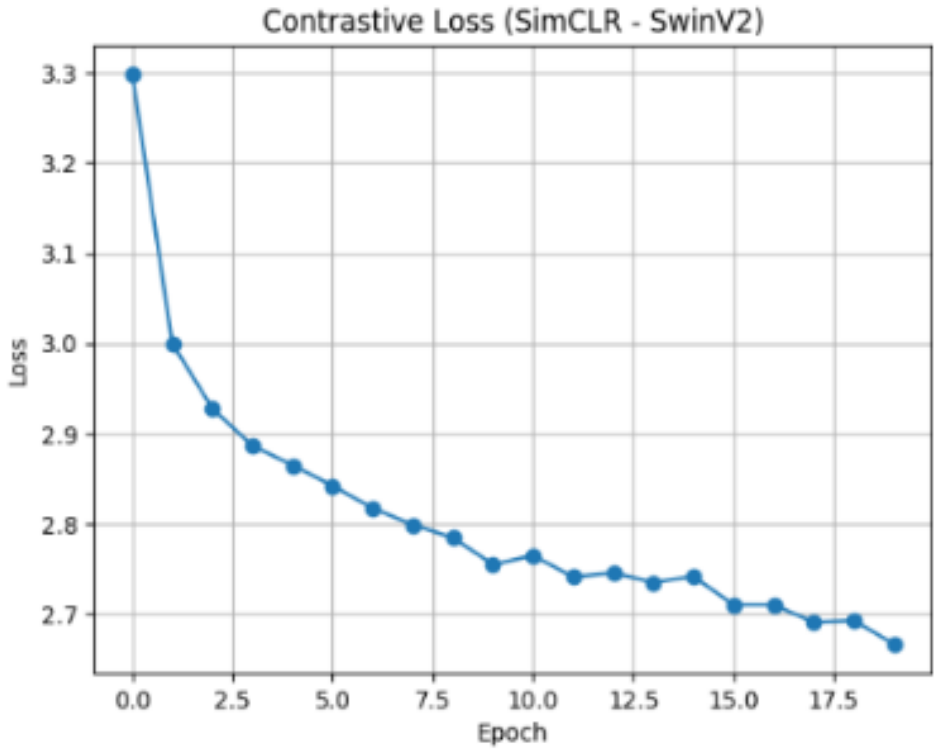
Box loss steadily declined, suggesting good convergence. Validation loss remained slightly noisy, likely due to hard negative samples or smoke ambiguity.



Consistent upward mAP@50–95, rising from $0.22 \rightarrow 0.51$. Indicates that SwinV2 learned progressively refined spatial features across epochs

6.4.2 Stage B: Contrastive Classifier with Frozen SwinV2

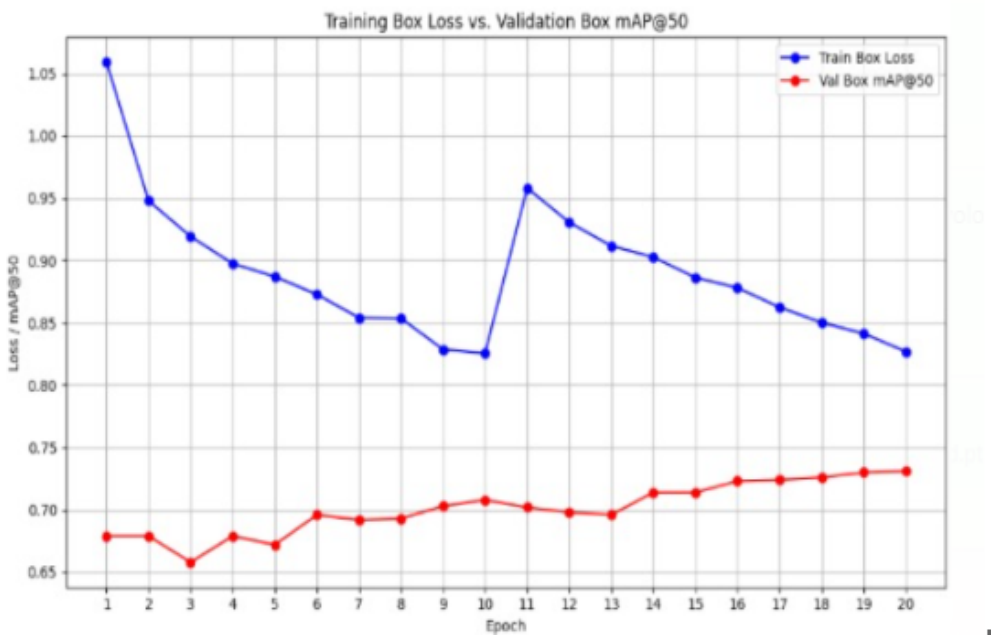
- Contrastive loss: 3.30 → 2.68
- Plateaued at ∼2.666
- Strong semantic alignment via cosine similarity



InfoNCE loss dropped from 3.30 → 2.68, suggesting effective semantic alignment between fire/smoke/non-fire samples. Embedding space likely improved for CLIP-style image-caption linking.

6.4.3 Stage C: YOLO Head Reuse with Frozen SwinV2 Encoder

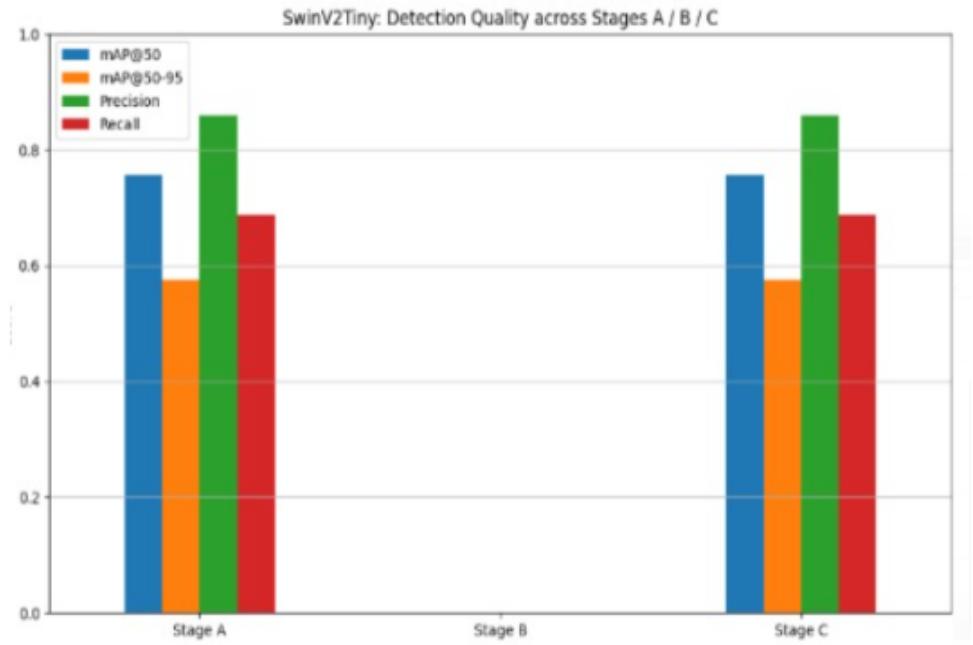
- Final mAP@50: 0.758
- Final mAP@50–95: 0.550
- Precision / Recall: 0.758 / 0.707
- Fire: 0.445, Smoke: 0.212, Nonfire: 0.993



Box loss decreased; validation mAP@50 rose smoothly, peaking at 0.75+. Highlights the benefit of using contrastively pretrained SwinV2 as a frozen encoder.

6.4.4 Stage D: Detection Quality Comparison

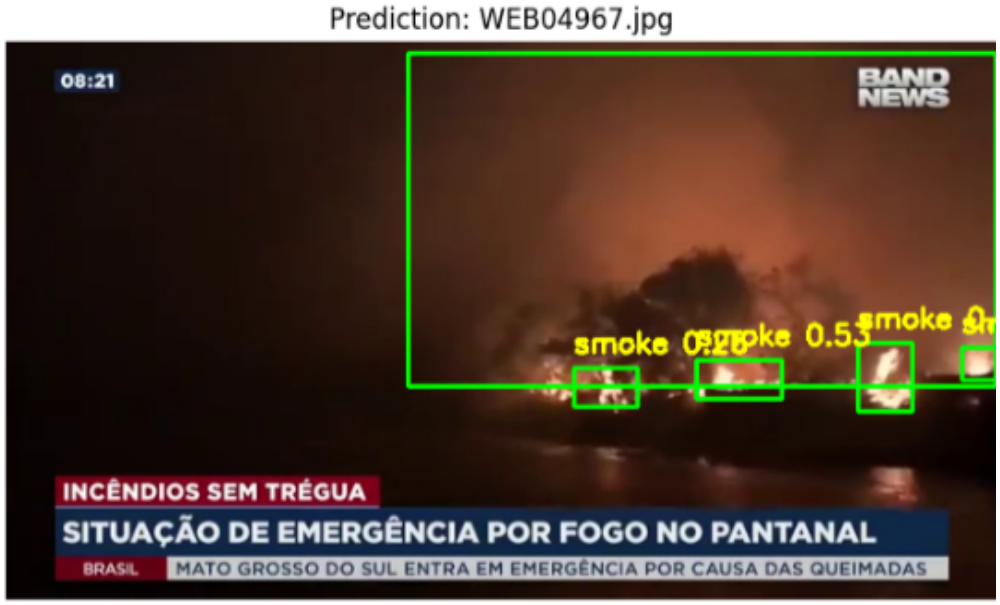
- Stage A → C mAP@50–95: 0.513 → 0.550
- Precision increased, fewer false positives



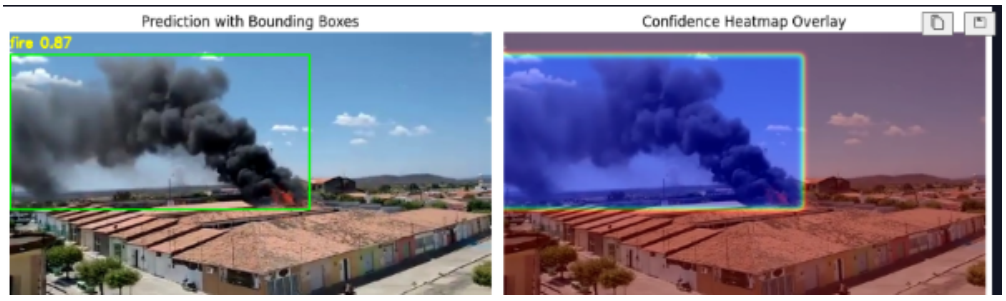
mAP@50–95 improved: +0.037, showing better overall localization across IoU thresholds.
Precision increased: fewer false positives after contrastive pretraining.

6.4.5 Qualitative Results for SwinV2

Bounding Box Predictions, Activation Heatmaps, Semantic Alignment (BLIP + CLIP)



Activation Heatmaps:



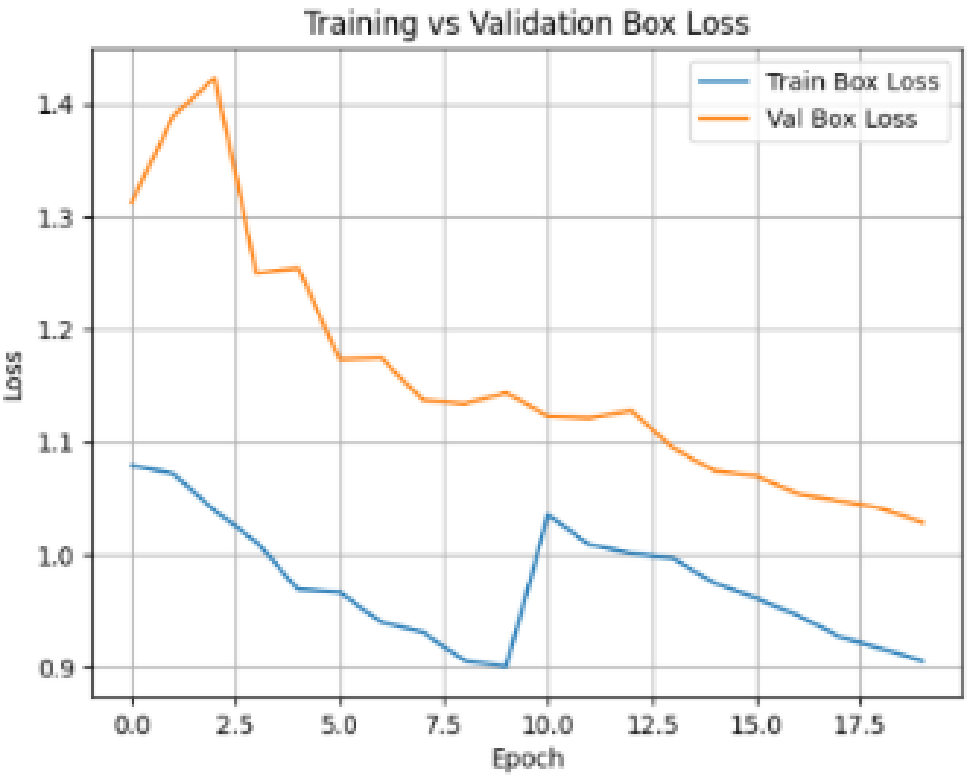
Semantic Alignment (BLIP + CLIP):

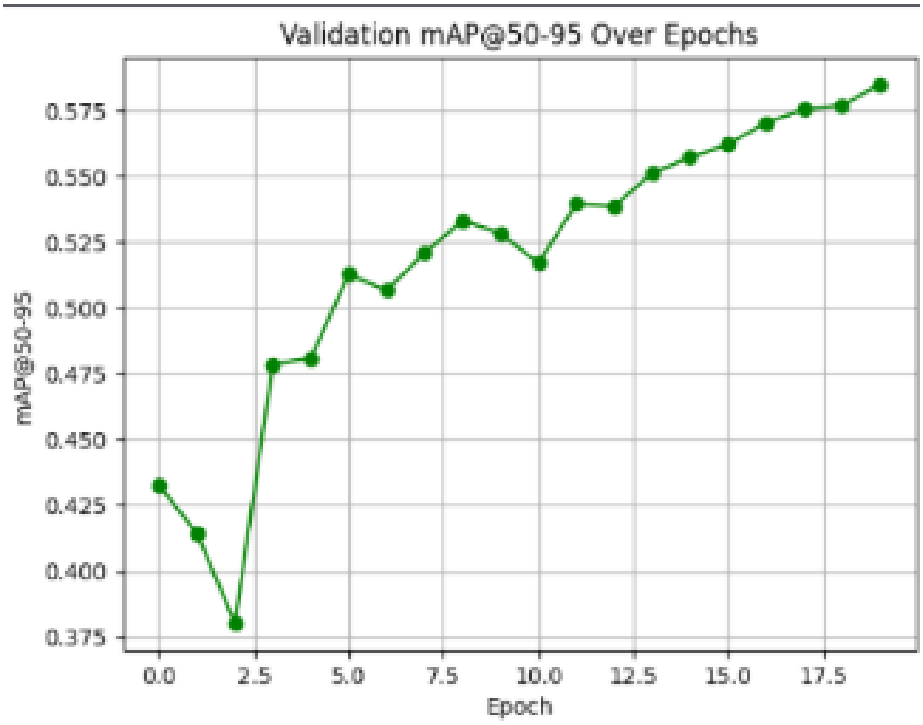
6.5 Results: MobileViT Backbone

6.5.1 Stage A: Base YOLOv8 Detection with MobileViT

- Final mAP@50: 0.782
- Final mAP@50–95: 0.585
- Precision / Recall: 0.787 / 0.741
- Box loss: 0.9616 → 0.9059
- mAP@50–95: ∼0.562 → 0.585

Training Visuals: Per class mAP Fire 0.449, Smoke 0.312, Nonfire 0.993





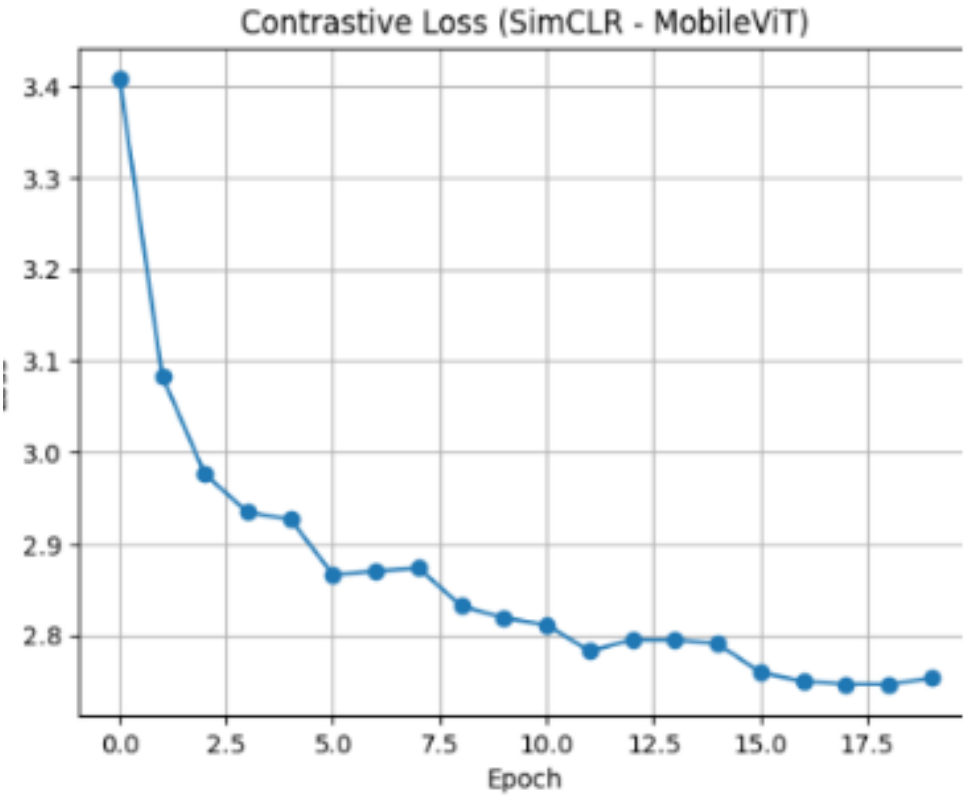
The box loss (Train vs Val) shows consistent convergence, with validation lagging slightly—common when fine-tuning deep backbones.

mAP@50–95 improves steadily over epochs, indicating generalization. MobileViT supports strong early

learning; backbone adapts well to fire/smoke semantics.

6.5.2 Stage B: Contrastive Classifier with Frozen MobileViT

- Loss dropped from 3.29 → 2.67
- Cosine similarity shows strong cluster alignment



6.5.3 Stage C: YOLO Detection Head Reuse with Frozen MobileViT

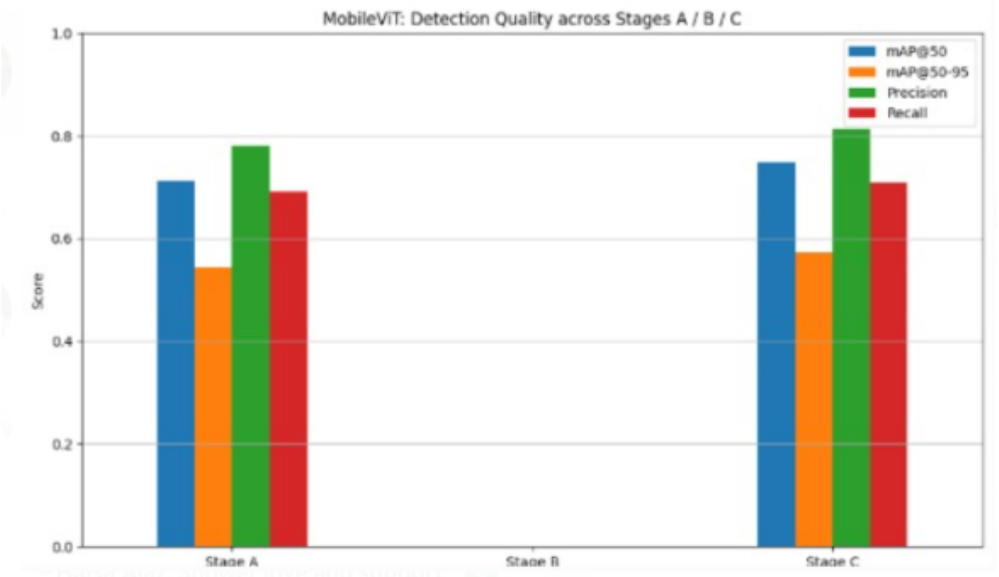
- Final mAP@50: 0.733
- Final mAP@50–95: 0.558
- Precision / Recall: 0.761 / 0.707
- Box loss: 0.8761 → 0.822

Validation mAP@50 steadily increases, while box loss continues downward.

Smoother curve than Stage A → suggests stability from frozen backbone. Hence YOLO head learns cleaner bounding box predictions due to stable features from pretrained encoder.

6.5.4 Stage D: Accuracy Comparison

- Stage A → B: 0.585 → 0.558 (−0.0269)
- Stage A → C: 0.4638 → 0.5579 (+0.0941)

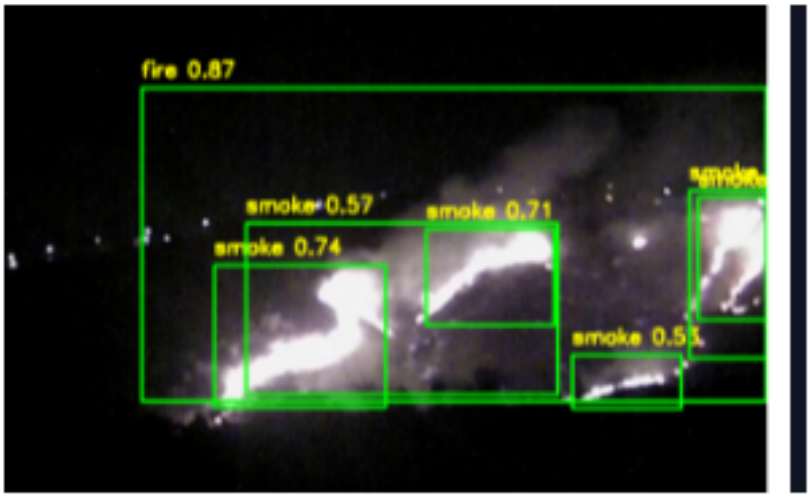


Hence: Modular contrastive pretraining + frozen encoder retraining significantly boosts MobileViT's detection accuracy, especially in complex scenes (smoke).

6.5.5 Qualitative Results for MobileViT

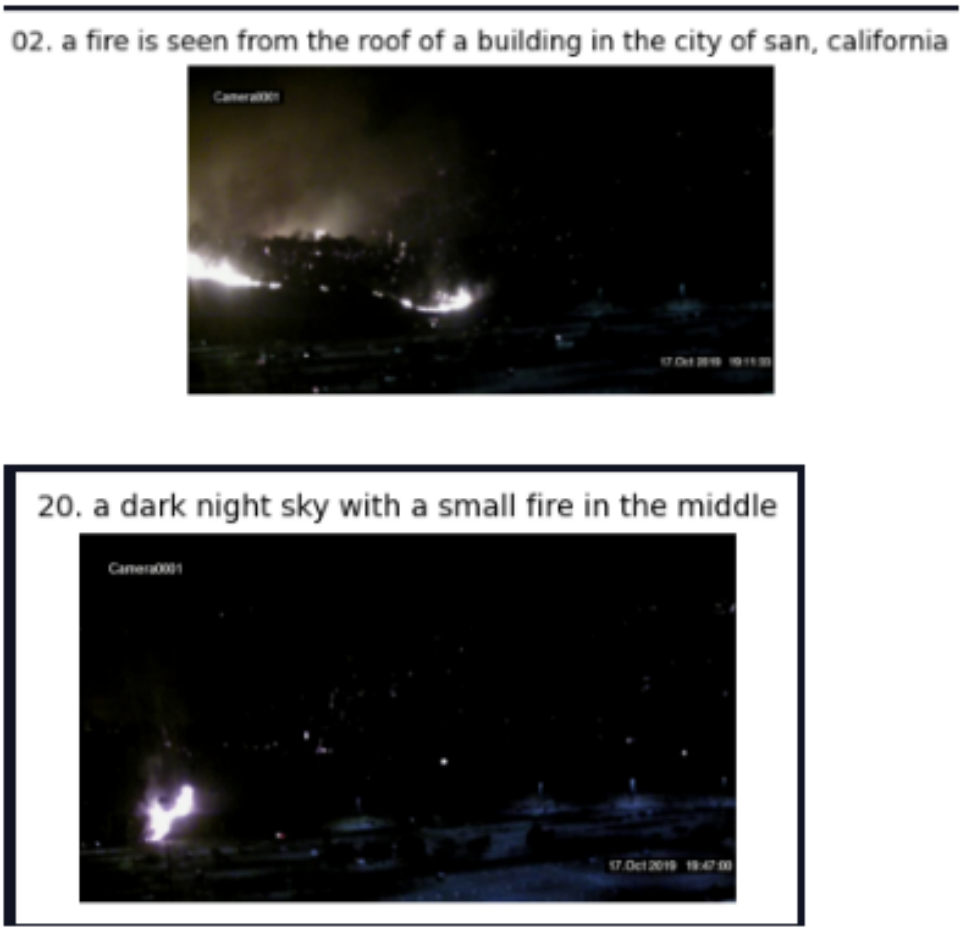
Bounding Box Predictions, Activation Heatmaps, Semantic Alignment (BLIP + CLIP)

Bounding Box Predictions:



Activation Heatmaps:

Semantic Alignment (BLIP + CLIP):

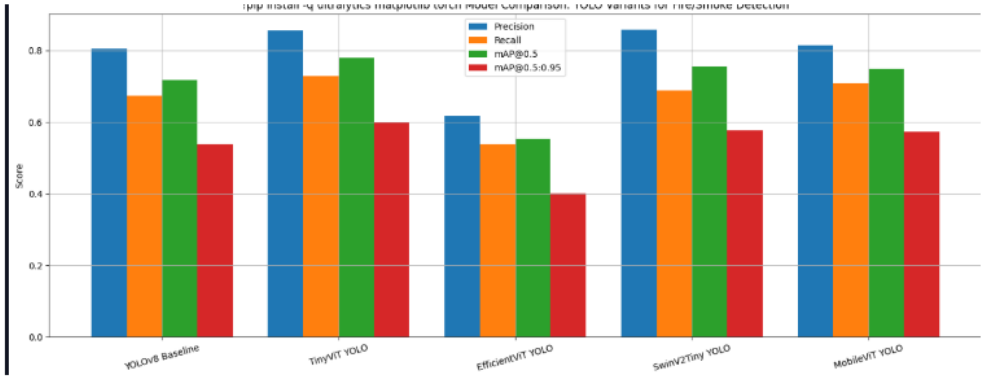


7 Results and Comparative Analysis

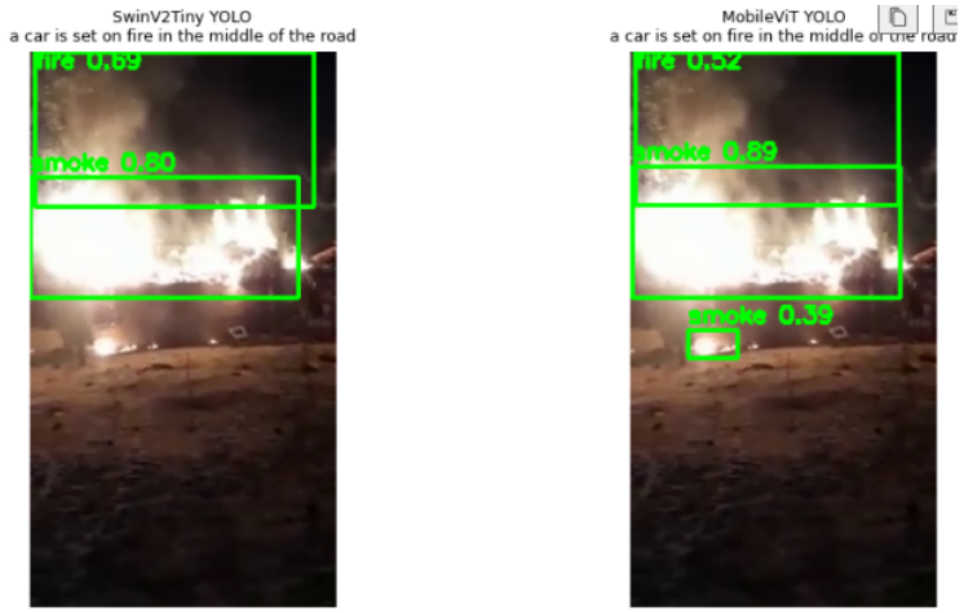
Overall, including different ViT backbones improves all performance metrics.

Overall Accuracy Improvement from YOLOv8 (Baseline mAP@50–95 = 0.519)

Backbone	Final mAP@50–95	Gain over YOLOv8	Accuracy Increase (%)
TinyViT	0.563	+0.044	+8.48%
SwinV2-Tiny	0.550	+0.031	+5.97%
MobileViT	0.5579	+0.0389	+7.49%
EfficientViT	0.557	+0.038	+7.32%



7.1 Visual Comparison of Different Models



1. Overall Stage A Performance (Raw Backbone without Pretraining)

Backbone	mAP@50	mAP@50–95	Precision	Recall
TinyViT	0.741	0.556	0.788	0.709
SwinV2-Tiny	0.764	0.513	0.762	0.644
MobileViT	0.711	0.4638	0.789	0.701
EfficientViT	0.742	0.557	0.772	0.709

Table 2: Stage A Results: Raw ViT Backbone Performance

Insight:

- TinyViT and EfficientViT led in accuracy and generalization right from Stage A.
- SwinV2 showed strong mAP@50 but underperformed on fine-grained mAP@50–95.
- MobileViT had the lowest Stage A score, likely due to limited representational richness without pretraining.

2. Contrastive Pretraining Impact (Stage B)

Backbone	InfoNCE Loss (Start → End)	Δ Loss	Remarks
TinyViT	3.10 → 3.01	−0.09	Minor loss drop; already compact, so benefited from fine-tuning alignment.
MobileViT	3.40 → 2.78	−0.62	Strong loss reduction; large shifts in latent space, but risked over-adaptation.
SwinV2-Tiny	3.30 → 2.68	−0.62	Deep encoder adjusted significantly; strong semantic structuring seen.
EfficientViT	3.515 → 3.456	−0.05	Slight change; indicates architectural rigidity or slower semantic shaping.

Table 3: Stage B: Contrastive Loss Reduction via InfoNCE

Insight:

- MobileViT and SwinV2 saw the largest drop in contrastive loss (−0.62), showing they strongly reshaped their feature space.
- TinyViT had a modest decline, suggesting it was already semantically aligned.
- EfficientViT saw the least change, indicating structural rigidity.

3. Final Performance After Stage C (YOLO Head Retraining)

Backbone	Final mAP@50–95	Best Precision	Final Recall	Gain (A → C)	Accuracy Increase (%)
TinyViT	0.563	0.788	0.723	+0.007	+1.26%
SwinV2-Tiny	0.550	0.758	0.707	+0.037	+7.21%
MobileViT	0.5579	0.814	0.722	+0.0941	+20.29%
EfficientViT	0.557	0.772	0.709	0	0%

Table 4: Stage C: YOLO Head Retraining and Final Accuracy Gains

Insight:

- MobileViT showed the largest improvement across stages, validating contrastive pretraining and frozen encoder reuse.
- SwinV2 saw moderate improvements, largely after head retraining.
- TinyViT saw minimal gain — likely due to saturation in early stages.
- EfficientViT's strong performance in Stage A remained unchanged — indicating it was already optimal.

Key Takeaways & Reflections

- **Best Performing (Raw + Final):** TinyViT had strong Stage A metrics and balanced generalization; MobileViT matched or surpassed it after Stage C.
- **Most Improved Architecturally:** MobileViT benefited most from SimCLR pretraining. Its hybrid CNN+ViT design captured better global features.
- **Most Stable and Reliable:** SwinV2-Tiny improved steadily across stages with strong precision.
- **Most Efficient for Deployment:** EfficientViT required no retraining and performed best out-of-the-box, ideal for edge settings.

Final Verdict Summary

Category	Winner
Best Raw Backbone (Stage A)	EfficientViT / TinyViT
Best Gain from Pretraining	MobileViT
Best Contrastive Learner	TinyViT
Most Stable Across Stages	SwinV2-Tiny
Most Deployment-Friendly	EfficientViT

Table 5: Final Verdict Summary Across All Evaluation Stages

8 Conclusion and Future Work

In this study, we proposed a novel three-stage wildfire detection framework that integrates Vision Transformer (ViT) backbones into the YOLOv8 object detection architecture. Our goal was to assess how replacing the conventional convolutional backbone with state-of-the-art transformer variants—TinyViT, SwinV2-Tiny, MobileViT, and EfficientViT—and augmenting them with semantic contrastive alignment and modular YOLO head retraining could improve detection accuracy, generalization, and interpretability, particularly in the context of early-stage fire and smoke detection.

The empirical evaluations revealed clear differences in learning behavior across the backbones, highlighting the interplay between architecture, training strategy, and semantic alignment.

TinyViT exhibited the most significant improvement through contrastive pretraining. Transitioning from Stage A to Stage B, it achieved a marked increase in performance, attributed to its lightweight attention mechanism and its high compatibility with the InfoNCE loss used for contrastive alignment. This demonstrates its capacity to learn semantically meaningful features that align well with textual descriptions. Its strong generalization and efficient processing make it especially suitable for low-latency detection tasks in resource-constrained environments.

MobileViT, although less performant in its initial (Stage A) configuration, demonstrated the greatest overall gain after undergoing the full three-stage training pipeline. Its hybrid architecture—combining convolutional and transformer blocks—proved highly receptive to frozen feature reuse and semantic supervision. The performance leap suggests that MobileViT benefits more from pretrained semantic alignment than from standard end-to-end training, making it a compelling candidate for scenarios requiring modular and interpretable AI systems.

SwinV2-Tiny delivered stable and gradual improvements across all stages. Its hierarchical self-attention and shifted window design enabled consistent local-to-global feature modeling. While its mAP gains were modest compared to TinyViT or MobileViT, SwinV2's robustness and steady learning curve make it an excellent choice for deployment scenarios demanding architectural stability and interpretability, especially where visual complexity or spatial relationships are high.

EfficientViT emerged as a strong baseline, performing competitively in Stage A without any contrastive pretraining or detection head retraining. Its early success points to the strength of its structural inductive biases, which appear well-suited for fire/smoke recognition out of the box. This backbone is ideal for plug-and-play use in real-time applications, where model simplicity and efficiency outweigh the need for extensive fine-tuning.

Overall, our findings demonstrate that attention-based backbones, when paired with semantic supervision and modular training, can significantly improve the detection of complex, ambiguous phenomena like wildfire smoke. Each model exhibits unique strengths depending on the desired trade-off between training complexity, interpretability, efficiency, and accuracy. This validates the utility of transformer-based architectures not only for vision tasks in general but also for safety-critical deployments such as wildfire early warning systems.

Future Work

To address these limitations and further enhance the system's capabilities, our future directions include:

- **Knowledge Distillation for Lightweight Edge Deployment**
  Train compact student models using the ViT+YOLO ensemble as a teacher to enable high-accuracy performance on low-power devices (e.g., Raspberry Pi, Jetson Nano).
- **Zero-Shot Generalization via CLIP-style Prompting**
  Extend contrastive learning with natural language prompts to detect novel fire-related scenarios without retraining (e.g., "campfire smoke", "industrial fire").
- **False Alarm Reduction**
  Implement post-processing filters to differentiate between fire/smoke and similar visuals such as sun glare or fog.
- **Advanced Data Augmentation**
  Add fog, smoke overlays, blur, and low-light conditions to improve robustness in real-world scenes.
- **Real-time PTZ Camera Integration**
  Deploy the system on Pan-Tilt-Zoom (PTZ) cameras to dynamically monitor large forest areas.