

# Wildfire Detection Using End-to-End Object Detection with Transformers

Maria Radoslavova

InspiritAI

November 10, 2024

# Outline

- 1 Introduction
- 2 Background
- 3 Data
  - Dataset Overview
  - Data Preprocessing
- 4 Model
  - Training Procedure
  - Hyperparameters
  - Evaluation Strategy
  - Visualization
- 5 Results
  - Model Performance
  - Performance Metrics
  - Visualization of Results
  - Error Analysis
- 6 Conclusion

# Introduction

- Climate change is increasing the frequency and severity of wildfires.
- Traditional detection methods are slow and often inaccurate.
- Early detection is crucial for mitigating wildfire damage.
- **Objective:** Employ state-of-the-art computer vision techniques, specifically the DEtection TRansformer (DETR), to improve wildfire monitoring systems.

# Background

- **DETR**: A transformer-based deep learning model for object detection.
- Utilizes self-attention mechanisms to recognize patterns across larger contexts.
- Underutilized in environmental monitoring applications.
- Our research applies DETR to the task of wildfire detection.

# Data Components

- Dataset of 6,249 high-resolution images from RoboFlow.
- Images depict various wildfire scenarios with bounding boxes around fires.
- Landscapes vary in lighting, forest type, fire number, and fire size.
- Aimed to ensure model robustness across different wildfire contexts.

# Data Preprocessing

## ① Dataset Download and Organization

- Downloaded from RoboFlow and organized into directories.
- Annotations formatted in COCO JSON standard.

## ② Dataset Splitting

- 90% of images for training.
- 10% of images for testing.

## ③ Image Standardization and Resizing

- All images resized to 512 x 512 pixels.

## ④ Bounding Box Normalization

- Coordinates normalized to a range between 0 and 1.
- Ensures compatibility across varying image sizes.

## Example Annotation Entry

```
{  
  "images": [  
    {  
      "id": 5631,  
      "license": 1,  
      "file_name": "82_869_927_...jpg",  
      "height": 512,  
      "width": 512,  
      "date_captured": "2024-01-06T19:08:39+00:00"  
    }  
  ]  
}
```

# Model Architecture

- Utilized the DETR architecture with a ResNet-50 backbone.
- Leveraged pre-trained facebook/detr-resnet-50 weights.
- Employed transfer learning to accelerate training and enhance performance.



# Training Procedure

## ① Data Preparation

- Custom data loaders with `collate_fn` function.
- Handled variable-sized ground truth data.

## ② Model Configuration

- Trained on a CUDA-enabled GPU.

## ③ Training Loop

- Forward pass, loss calculation, backpropagation.
- Optimizer: AdamW.
- Loss tracking at regular intervals.

## ④ Model Saving

- Checkpoints saved after each epoch.

# Hyperparameter Tuning

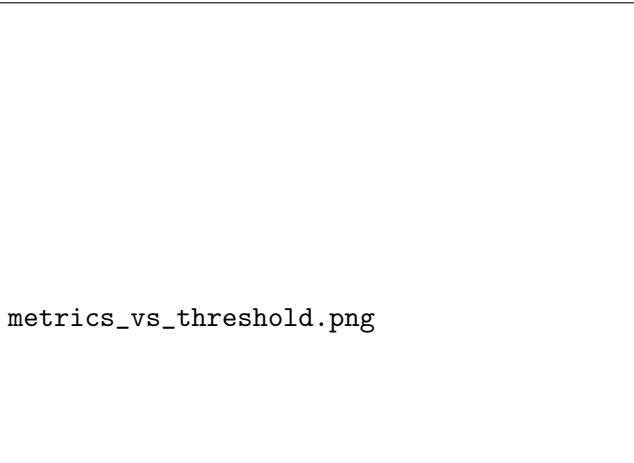
- **Batch Size:** 10
- **Epochs:** 15
- **Training Size:** 90%
- Optimal configuration after experimenting with different combinations.

# Evaluation Metrics

- **Mean Intersection over Union (mIoU)**
  - Measures overlap between predicted and ground truth bounding boxes.
- **Precision**
  - Proportion of correct wildfire detections.
- **Recall**
  - Proportion of actual wildfires correctly detected.
- Implemented parallel processing for efficiency.

# Visualization

- Used matplotlib for plotting results.
- Displayed mIoU, Precision, and Recall versus detection threshold.
- Provided insights into optimal threshold settings.



metrics\_vs\_threshold.png

# Model Performance

- **Training Loss**

- Steady decrease from 5.5 to 0.32 over epochs.

- **Validation Loss**

- Decreased from 0.98 to 0.42.
- Minor fluctuations but overall positive trend.

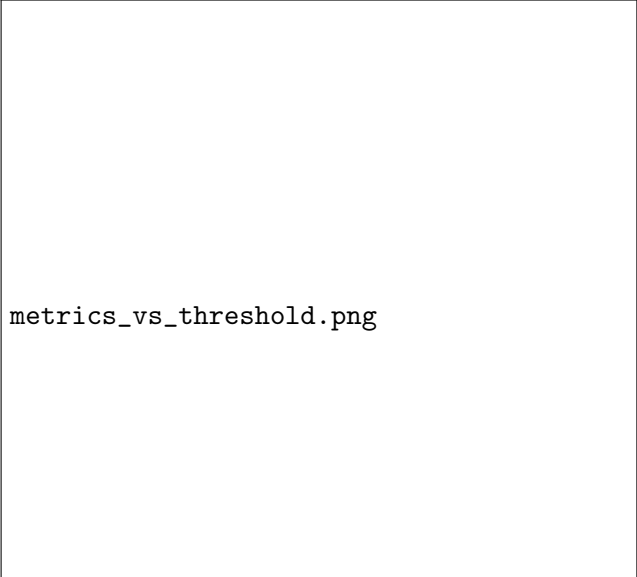
- Indicates successful learning and good generalization.

# Performance Metrics

Hyperparameters	Precision	Recall	mIoU
Batch size: 7, Epochs: 10	0.85	0.88	0.83
Batch size: 7, Epochs: 15	0.87	0.86	0.84
Batch size: 10, Epochs: 10	0.83	0.85	0.82
<b>Batch size: 10, Epochs: 15</b>	<b>0.86</b>	<b>0.87</b>	<b>0.83</b>

Table: Performance Metrics for Different Hyperparameters

# Visualization of Results



metrics\_vs\_threshold.png

# Reasons for Errors

- **Heavy Smoke**

- Obscures key features needed to identify a wildfire.

- **Similar Colors**

- Orange-leaved trees may be mistaken for fire due to color similarity.

- **Environmental Variability**

- Diverse conditions make consistent detection challenging.



# Conclusion

- Successfully demonstrated the potential of using DETR for wildfire detection.
- Model accurately identified fires across various environmental conditions.
- Both training and validation losses showed steady decreases.
- DETR could be integrated into real-world wildfire detection systems.
- **Future Work:**
  - Fine-tuning hyperparameters.
  - Addressing issues like smoke interference.
  - Optimizing the model for real-time deployment.

# Acknowledgments

- Special thanks to **Andrew Kent** for guidance and expanding my knowledge in machine learning.

# References



Carion, N., et al. (2020). End-to-End Object Detection with Transformers. *European Conference on Computer Vision*.



RoboFlow Dataset. Available at: <https://roboflow.com/>

**Thank You!**