AP FINAL PROJECT

Fire detection

Shoplikov Alisher

Abdymazhit Toretay

IT-2202

Оглавление

Introduction:

Problem:

The purpose of this research is to create a deep learning model that detects fire in the format of live video or recording. Fires are a serious threat to people and property, and early identification is crucial for effective firefighting and preventing its' harmful outcomes. Traditional fire detection technologies frequently rely on people detecting fire and informing about it, which may be very slow and not effective, particularly in large-scale fires.

To solve these issues, we suggest using deep learning and computer vision techniques to automatically identify fires in video feeds. The project's goal is to create a reliable and accurate fire detection system capable of real-time monitoring of video feeds from surveillance cameras, drones, and other sources of video recording.

Literature Review:

1. This project improves fire detection using neural networks over traditional detectors. It explores various architectures and datasets, focusing on edge device optimization. Challenges include managing fire scenarios and ensuring interpretability. It discusses preprocessing, augmentation, and performance metrics.
   https://github.com/robmarkcole/fire-detection-from-images

2. This project focuses on fire and smoke detection using YOLOv4 and YOLOv5 deep learning models. Key components include datasets, model training, and testing procedures for both YOLOv4 and YOLOv5. The project also introduces a new dataset for detecting reflective clothing and safety helmets.
   https://github.com/gengyanlei/fire-smoke-detect-yolov4

3. This project implements YOLOv5 in C++ to detect fire or smoke in the wild using Jetson Xavier NX and Jetson Nano. It achieves 33 FPS on Xavier NX. Requirements include Jetpack 4.5.1, Python 3, TensorRT, Torch, and OpenCV. It involves generating YOLOv5 models, training custom models, and optimizing with TensorRT for INT8 quantization.
   https://github.com/RichardoMrMu/yolov5-fire-smoke-detect

Current Work:

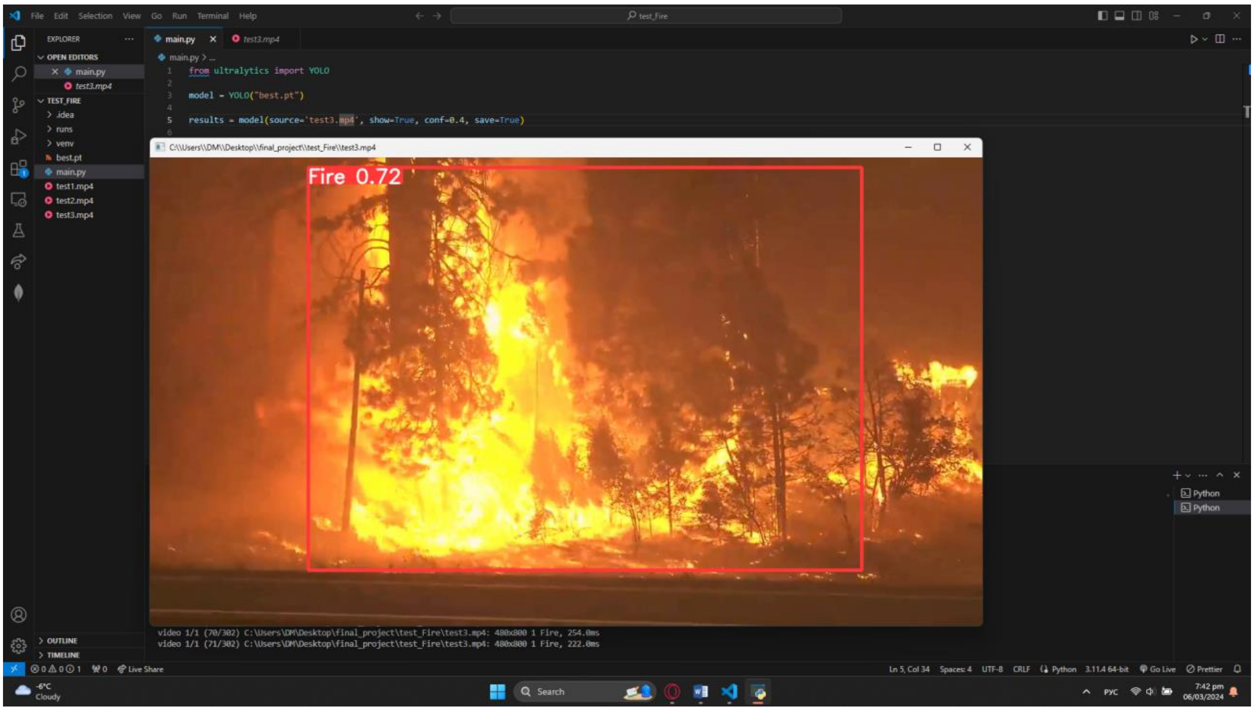
The project involves creating and testing an object detection model for wildfire detection that uses YOLOv8 technology. The project begins with data collection, which uses the Roboflow platform to access bunch of datasets designed for wildfire detection tasks. The YOLOv8 architecture is then implemented using the Ultralytics library, which allows for rapid model training and evaluation.

Training begins with the configured YOLOv8 model, which uses 50 epochs to constantly tune its parameters. Key training parameters such as image size, learning rate, and augmentation approaches are selected to tune and improve model performance. The trained model is then validated to determine its performance to reliably predict wildfires in previously unreported data, ensuring adaptability and generalization.

After the validation, the model performs prediction tasks on test images to show its performance in practical situations. The predictions are analyzed and presented to demonstrate the model's performance and potential for use in wildfire monitoring and management.

```
image 261/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-622_png_jpg.rf.67e57c96e804be10e230fd3b37b6c889.jpg: 800x800 3 Fires, 1 smoke, 11.7ms
image 262/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-674_png_jpg.rf.510c1ed8b0c68068d5ea3547d5c08a78.jpg: 800x800 1 Fire, 1 smoke, 11.7ms
image 263/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-687_png_jpg.rf.328408d1d37c137326d1dc53ac51a58d.jpg: 800x800 7 Fires, 2 smokes, 11.7ms
image 264/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-732_png_jpg.rf.c69f092d4bd5dbf8cd6849d756eca486.jpg: 800x800 1 Fire, 11.6ms
image 265/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-90_png_jpg.rf.0c428bb899826c872b5709544d26e8e5.jpg: 800x800 5 Fires, 11.5ms
image 266/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-93_png_jpg.rf.29ed1ec67e0dbee7f0a3585a05442907.jpg: 800x800 2 Fires, 1 smoke, 11.5ms
image 267/267 /content/datasets/Wildfire-Detection-3/test/images/new_fire_fire-97_png_jpg.rf.0acb0b333a84b53682d65c6a5f414bb9.jpg: 800x800 1 Fire, 1 smoke, 11.5ms
Speed: 4.2ms preprocess, 13.5ms inference, 3.8ms postprocess per image at shape (1, 3, 800, 800)
Results saved to runs/detect/predict
```

As part of the project, we have developed a mini script to detect fire from video footage using the YOLOv8-based object detection model created. The script is designed to process video input, frame by frame, and apply the trained model to identify instances of fire within the footage.



Data and Methods:

Data:

We used a dataset sourced from Roboflow consisting of annotated images for fire detection. The dataset includes many environmental conditions and wildfire footages. There are two classes in the given dataset: smoke and fire.

```python
import os
dataset_dir = '/content/Wildfire-Detection-3'

train_dir = os.path.join(dataset_dir, 'train/images')
test_dir = os.path.join(dataset_dir, 'test/images')
valid_dir = os.path.join(dataset_dir, 'valid/images')

num_train_images = len(os.listdir(train_dir))
num_test_images = len(os.listdir(test_dir))
num_valid_images = len(os.listdir(valid_dir))

print("Number of images in the train set:", num_train_images)
print("Number of images in the test set:", num_test_images)
print("Number of images in the valid set:", num_valid_images)
```

```
Number of images in the train set: 2095
Number of images in the test set: 267
Number of images in the valid set: 748
```

Methods:

We used YOLOv8 model, in the form of transfer learning with the training on the custom dataset. YOLO is an object detection model known for its speed and simplicity. YOLOv8, one of its latest its' versions, it offers a balance between speed and accuracy, making it suitable for real-time applications, which is suitable for our purpose. It achieves competitive performance while being versatile and useful to various object detection tasks. Overall, YOLOv8 is widely used due to its efficiency and effectiveness in detecting objects in images and videos, this is the reason why we used the given model.

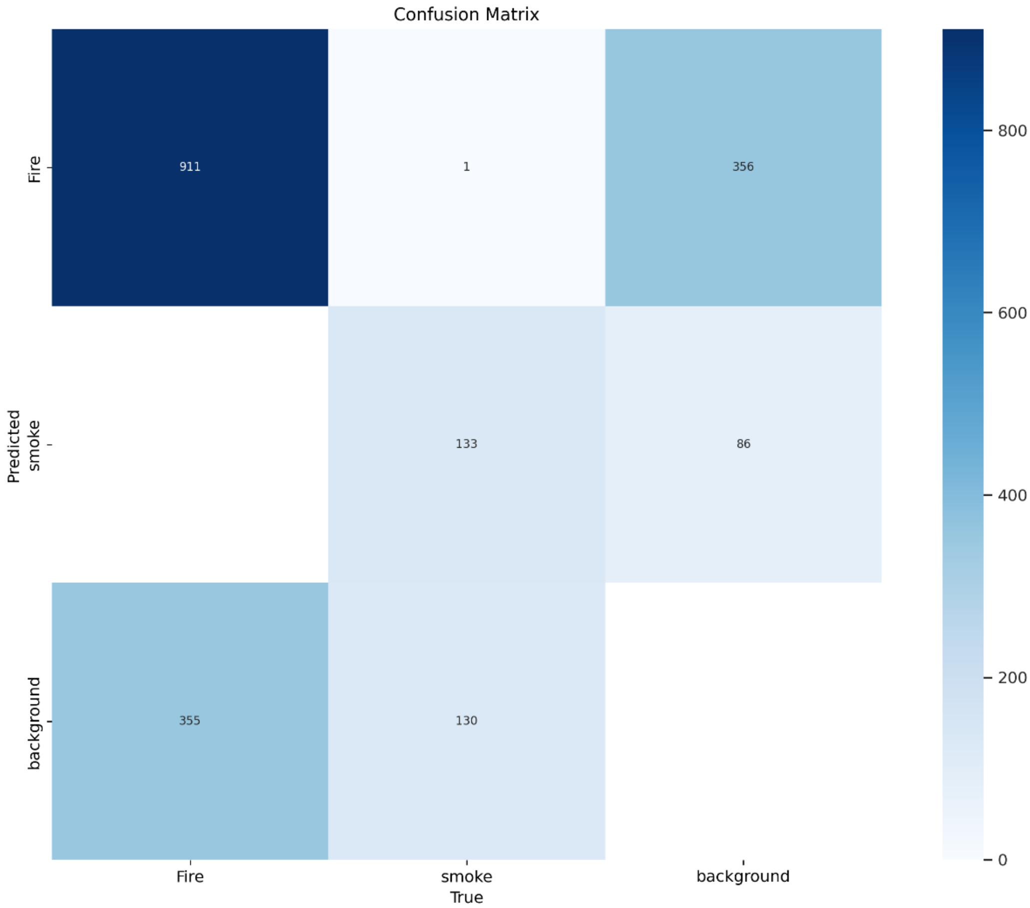
In our work, we specifically used YOLOv8 to detect wildfires in photo and video data formats. We used the Ultralytics package and the Roboflow platform to train and fine-tune the YOLOv8 model on a wildfire detection dataset. This allowed us to create a strong and efficient model capable of properly detecting wildfires in real-world situations.

Results:

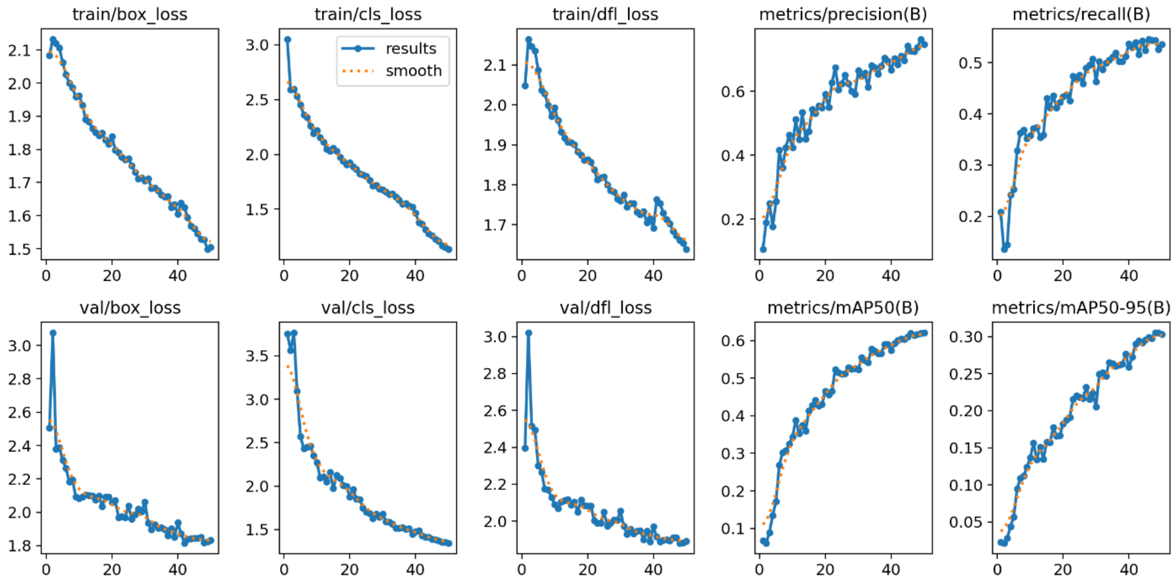
Model Performance:

We trained the YOLOv8 model for 50 epochs with an image size of 800, a learning rate of 0.001, and augmentation techniques including rotation, translation, and shear.

Confusion Matrix:



Validation Results:



Sample Predictions:

Discussion:

The results of the project demonstrate the successful implementation and training of the model for wildfire detection. We reached a decent level of accuracy in identifying wildfires in photos and video footage by meticulously using a non-small dataset with 2 classes, training the model, and validating them. The model's power to detect wildfires in real-time circumstances shows that it can be used in active fire monitoring and management.

Critical Review of Results:

While our algorithm has demonstrated not bad performance in spotting fires, there are room additional improvement. One issue is the model's precision, particularly in terms of identifying wildfire-related classes such as smoke, which is the secondary class in our dataset. Furthermore, fine-tuning the model on larger and more diverse datasets, such as video recordings, may improve its generalization ability and stability under different environmental situations.

Next Steps:

1. Training on video datasets: to improve our model to include video datasets, allowing for early detection of wildfires in complex situations.
2. Improved class precision: improve the model's class parameters and annotations to obtain higher accuracy, particularly when separating between many causes of wildfires, such as smoke, flames, and burned areas, which we also want to add.

Sources:

- YOLOv8: How to Train for Object Detection on a Custom Dataset
  https://www.youtube.com/watch?v=wuZtUMEiKWY
- Wildfire Detection Computer Vision Project
  https://universe.roboflow.com/ipv-investigacao/wildfire-detection-5odum