Detecting People in Smoked Rooms

Introduction

This project focuses on developing a machine learning model capable of detecting humans in smoke-filled environments using data captured from CCTV cameras or firefighter equipment. By leveraging advanced machine learning architectures, we aim to enhance safety measures during firefighting operations and emergency responses.

Objective

The primary objective of this project is to create a machine learning model using Pytorch and YOLOv5 (You Only Look Once version 5) for real-time detection of humans in smoke-filled environments. The model will process images from a specified dataset, detect people, and provide confidence scores for each detection.

Requirements

- Python 3.6+
- PyTorch
- PIL (Pillow)
- Matplotlib

Dataset

1. Fire Images Dataset (Indoor and Outdoor): This dataset contains images of various indoor and outdoor fire scenarios, with and without people.
    - Fire Images Dataset
2. House Rooms Image Dataset: This dataset includes images of various rooms in houses. These images will serve as the base environment.
    - House Rooms Image Dataset
3. External Smoke Images: Additional images of smoke will be overlaid onto the house room images to create a realistic dataset of smoke-filled rooms.

Theory

The project employs YOLOv5, an object detection model known for its speed and accuracy. It divides the input image into a grid and predicts bounding boxes and confidence scores for potential objects within these grids. By training this model on a dataset that simulates smoke-filled environments, we aim to achieve reliable detection of humans under such conditions.

1. Model Structure

YOLOv5's architecture consists of three main parts:

- **Backbone**: The core of the network, utilizing the New CSP- Darknet 53 structure, a modification of the Darknet architecture.
- **Neck**: Connects the backbone and the head using SPPF (Spatial Pyramid Pooling-Fast) and New CSP-PAN structures.
- **Head**: Responsible for generating the final output, utilizing the YOLOv3 Head.

2. Data Augmentation Techniques

YOLOv5 employs various data augmentation techniques to improve generalization and reduce overfitting:

- **Mosaic Augmentation**: Combines four training images into one to handle different object scales and translations.
- **Copy-Paste Augmentation**: Generates new training samples by pasting random patches from one image onto another.
- **Random Affine Transformations**: Includes random rotation, scaling, translation, and shearing of images.
- **MixUp Augmentation**: Creates composite images by taking a linear combination of two images and their labels.
- **Albumentations**: Uses a library that supports a variety of augmentation techniques.
- **HSV Augmentation**: Random changes to the hue, saturation, and value of images.
- **Random Horizontal Flip**: Flips images horizontally at random.

3. Training Strategies

YOLOv5 uses several advanced training strategies to enhance model performance:

- **Multiscale Training**: Randomly rescales input images within a range during training.
- **AutoAnchor**: Optimizes anchor boxes to match the characteristics of the ground truth boxes in custom data.
- **Warmup and Cosine LR Scheduler**: Adjusts the learning rate to enhance performance.
- **Exponential Moving Average (EMA)**: Stabilizes training by averaging parameters over past steps.
- **Mixed Precision Training**: Reduces memory usage and enhances computational speed by performing operations in half-precision format.
- **Hyperparameter Evolution**: Automatically tunes hyperparameters for optimal performance.

4. Additional Features

- **Compute Losses**: Combines Binary Cross-Entropy (BCE) loss for classification and objectness with Complete IoU (CIoU) loss for localization.
- **Balance Losses**: Weighs objectness losses differently across prediction layers (P3, P4, P5) with balance weights of [4.0, 1.0, 0.4].
- **Eliminate Grid Sensitivity**: Updates box prediction strategy to reduce grid sensitivity and prevent unbounded box dimensions.
- **Build Targets**: Assigns ground truth boxes to appropriate grid cells and matches them with anchor boxes, ensuring each ground truth object is properly assigned during training.

Testing

The trained YOLOv5 model will be tested on a separate set of images to evaluate its performance. The testing phase will involve:

- Loading the YOLOv5 model
- Processing images from the test dataset
- Detecting people in these images
- Evaluating the detection accuracy and confidence scores

Conclusion

- Utilized YOLOv5 and PyTorch to detect humans in smoked rooms, enhancing detection in low-visibility conditions.
- Used PIL and Matplotlib for image preprocessing, augmentation, and visualization for effective model training.
- Combined fire scene and house room datasets, adding synthetic smoke using OpenCV to improve model robustness

This project aims to significantly improve the detection of humans in smoke-filled environments, thereby enhancing the efficiency and safety of firefighting and emergency response operations. By utilizing advanced machine learning techniques and comprehensive datasets, we strive to develop a reliable and accurate detection system.

References

https://docs.ultralytics.com/yolov5/tutorials/architecture_description/

https://pytorch.org/