GANFORGE
Assignment-2 Report

Dipsikha Rano
240365

June 8, 2025

Insights

Through this assignment, I gained hands-on experience with various image processing techniques using OpenCV, such as color space conversion, thresholding, filtering, and sharpening. I also learned how to apply object detection using YOLOv5 and understood the workflow of setting up and running pre-trained models in videos also its application in image processing. Reviewing the YOLO research paper further deepened my understanding of how modern object detection models are structured and optimized for speed and accuracy.

Task 1: Image Processing with OpenCV

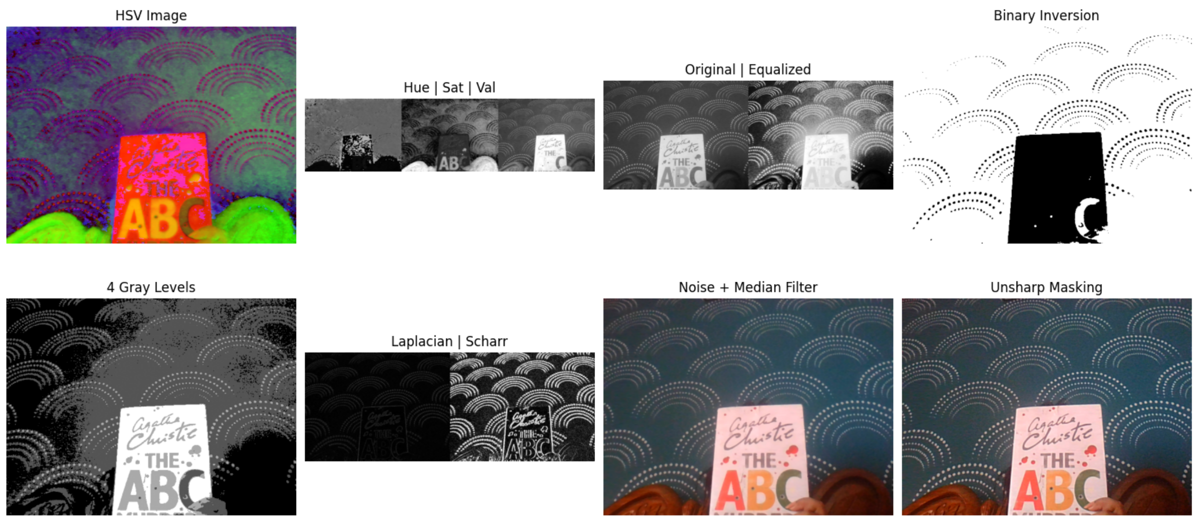

Figure 1: 2x4 Grid Showing Image Processing Results

Terminal Screenshot

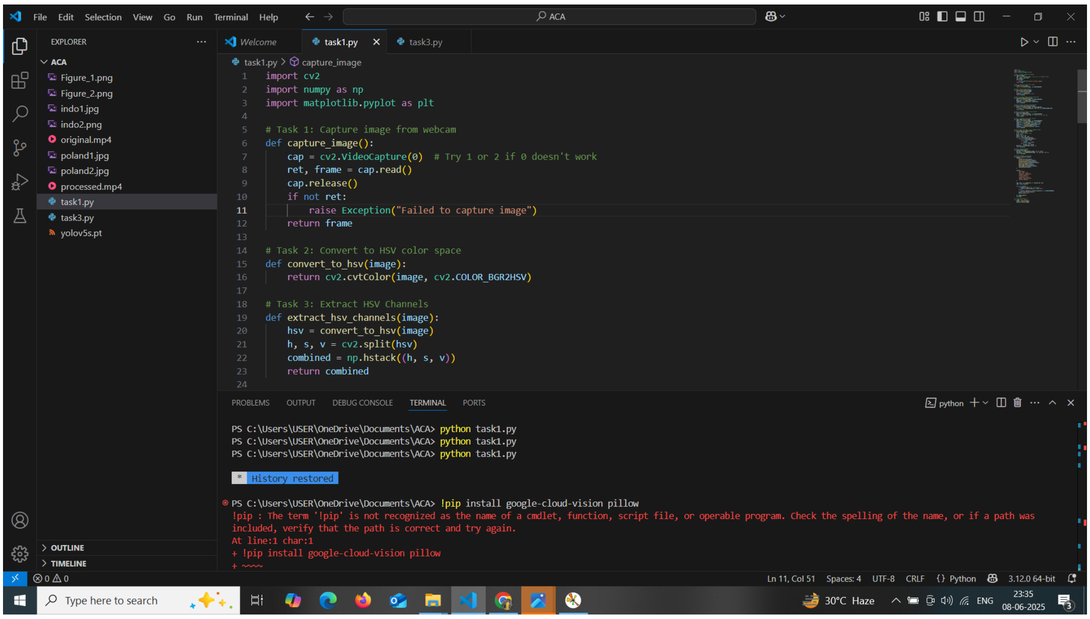

Figure 2: Terminal Showing Code Execution and Output

Task 2: YOLOv5 Object Detection

Before Detection

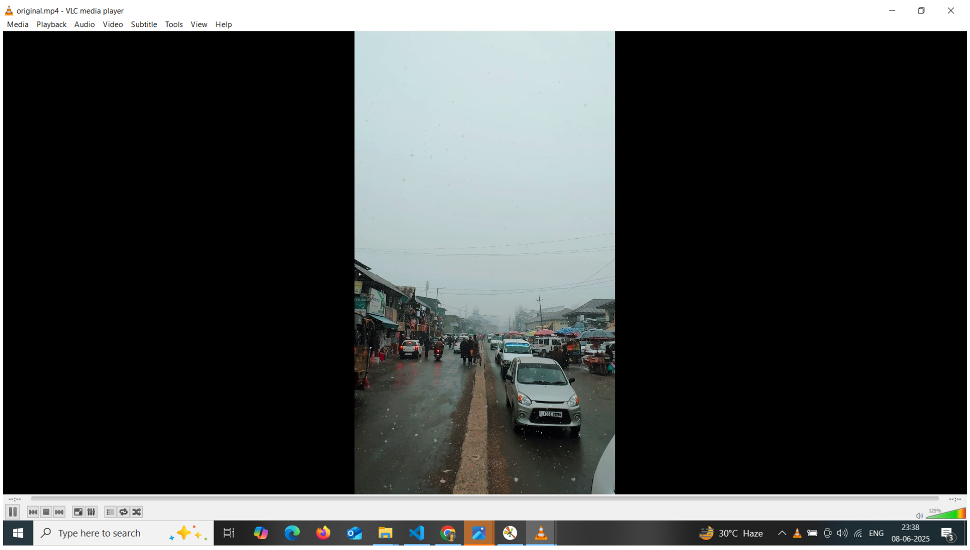

Figure 3: Input Image/Video

After Detection

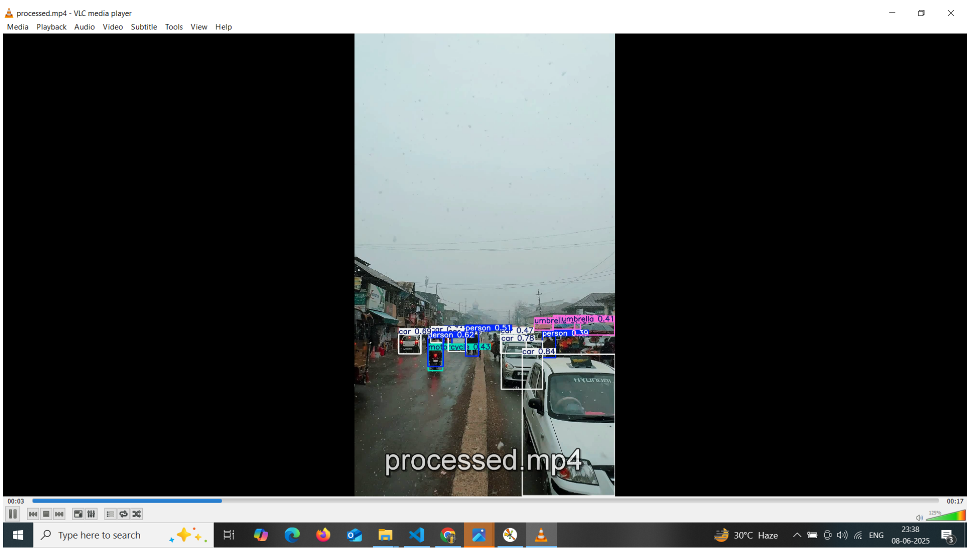

Figure 4: Output from YOLOv5

Terminal Output

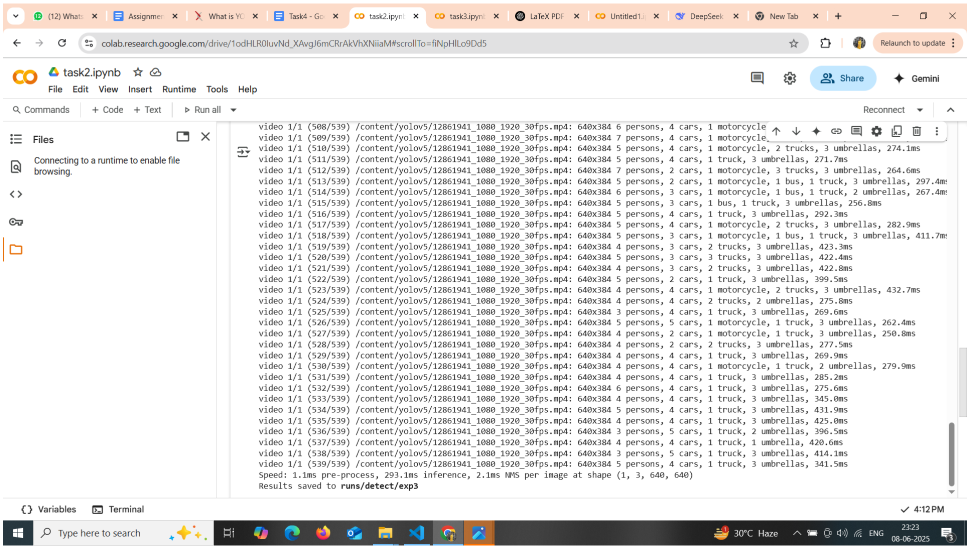

Figure 5: YOLOv5 Model Execution

Task 3: Indonesia vs Poland Flag Classifier

Paste your explanation here. For example:

This program uses OpenCV to analyze the color layout. It checks the color intensities of the top and bottom halves of the image. If the top half is predominantly red and the bottom is white, it classifies it as Indonesia. If the reverse is true, it's classified as Poland. HSV color space is used for robustness.

Detected Output

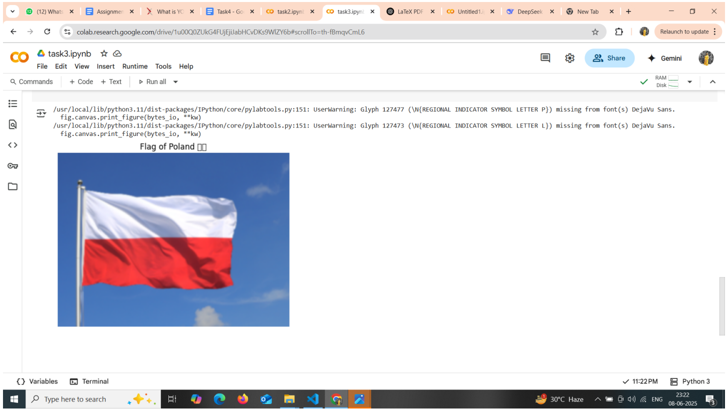

Figure 6: Classification Result

Terminal Screenshot

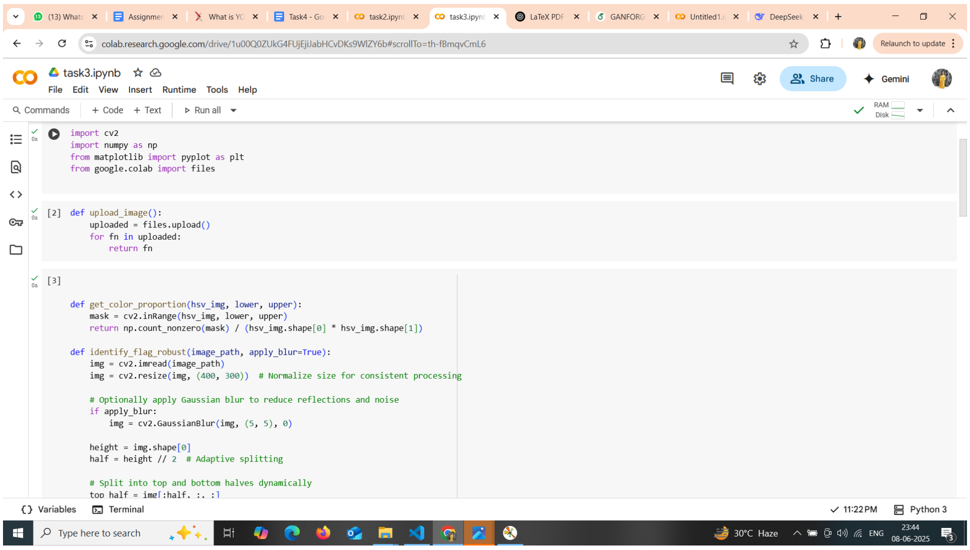

Figure 7: Terminal Showing Classification

Task 4: YOLOv5 Paper Summary

YOLOv5 is one of the most widely used object detection models due to its speed, flexibility, and performance. You Only Look Once (YOLO) examines the entire image just once to identify objects and their positions. In contrast to conventional methods employing two-stage detection processes, YOLO treats object detection as a regression problem. YOLOv5 is a single-stage object detector, which means it predicts bounding boxes and class labels directly from input images without any intermediate region proposals.

Architecture Overview

Backbone: CSPDarknet
CSP stands for Cross-Stage Partial networks. By decoupling feature maps into two main parts and recombining them, the CSP module effectively reduces computational cost and model complexity without compromising performance. This efficiency enhancement is particularly advantageous for the YOLO family.

Neck: PANet
The Path Aggregation Network (PANet) aims to boost information flow in a proposal-based instance segmentation framework. The feature hierarchy is enhanced with accurate localization signals in lower layers by bottom-up path augmentation. Adaptive feature pooling is also employed, allowing each feature level to contribute directly to proposal subnetworks.

Head:
The head makes the final predictions—bounding boxes, objectness scores, and class probabilities. YOLOv5 uses *anchor boxes*, which are predefined sizes used to detect objects of various shapes and scales.

Training Strategy

Data Augmentation – Mosaic
A novel technique that combines four images into four randomly sized tiles. It makes the model more robust to changes in object size, position, and background.

Anchor Box Optimization
YOLOv5 employs a novel approach to anchor box generation, using K-means clustering and genetic algorithms to derive anchor box dimensions directly from the dataset's bounding box distribution.

Loss Function
The loss function combines Binary Cross-Entropy (BCE) for class prediction and objectness, and Complete Intersection over Union (CIoU) for localization. The total loss is a weighted sum of these components.

16-Bit Floating Point Precision
YOLOv5 leverages PyTorch's support for 16-bit floating point numbers, reducing memory usage and increasing speed during both training and inference.

Model Variants

- **YOLOv5n (nano)** – Smallest, fastest, least accurate.
- **YOLOv5s (small)** – Good speed-accuracy balance.
- **YOLOv5m (medium)**
- **YOLOv5l (large)**
- **YOLOv5x (extra-large)** – Slowest but most accurate.

Evaluation

Architectural Advancements: YOLOv5 improves upon previous YOLO versions with the CSP backbone and PANet neck, boosting efficiency and accuracy.

Model Versatility: Its five variants offer flexibility for different hardware setups and application needs.

Training Innovations: Techniques like mosaic augmentation help with small object detection and reduce dataset requirements.

Performance and Impact: YOLOv5 achieves high mAP scores and low inference times, making it suitable for real-time detection. Its PyTorch-based implementation also increases accessibility for researchers and developers.

Conclusion

YOLOv5 is not just fast—it's intelligently designed to meet modern object detection demands with minimal computation and high accuracy.

GitHub Repository

You can access the full code and additional files here: https://github.com/your-username/ganforge-assignment2