Image based Malware Classification

Comparative Analysis of YOLOv8 and YOLOv11 for Malware Image Classification

Sylan Padmakumar Pranjal Khare Anand Pandey Vikash Kumar Ayush Monga

Capstone Project 2025 - 2026

Table of Contents

1 introduction

2 Workflow

3 Results

Introduction

In our project we are using the YOLO ML models to classify malware. Our dataset consists of malware from 9 different families.

We have chosen to do a comparative study of the different YOLO models to identify their efficiency in classifying malware.

Table of Contents

1 introduction

2 Workflow

3 Results

Dataset Preparation

Dataset Acquisition:

- Dataset acquired from Microsoft malware classification challenge.
- Consists of bytes files acquired from malware belonging to 9 different families.
- The malware families consist of: Ramnit, Lollipop, Kelihos ver3, Vundo, Simda, Tracur, Kelihos ver1, Obfuscator.ACY, Gatak

Dataset Conversion

The bytes files had to be converted into images for classification by the YOLO image models, the following methods were used:

- The files are arranged into 9 different class directories.
- We have used the 3-gram rgb method proposed by arXiv:2207.00421.

Dataset Preparation

dataset was split in the following way to comply with YOLO training specifications:

Dataset split:

- 80% → train
- 10% → validation
- 10% → test

Software Workflow

Training Workflow

- Used GPU compute instance hired from a cloud compute provider.
- Installed all the requirements(Pytorch, Ultralitics, Matplotlib, numpy, pandas) on a python virtual environment.
- Verified CUDA and GPU availability using Torch.
- Downloaded pre-trained classification weights.

Hardware

Used a pytorch template VM with the following hardware specifications

- Used Nvidia RTX A5000 GPU (24GB GDDR6 with error-correction code (ECC)).
- 32GB Ram and 7 CPUs.

Note

GPU and compute was acquired from www.jarvislabs.ai.

Table of Contents

1 introduction

2 Workflow

3 Results

Results

Model complexity and final classification metrics:

Model	**Parameters**	**GFLOPs**	**Final Top-1 Acc**
YOLOv8m-cls (final)	15,774,185	41.6	98.5%
YOLO11m-cls (final)	10,353,161	39.3	98.9%
YOLOv8n-cls (smoke)*	1,446,409	3.3	93.7%
YOLO11n-cls (smoke)*	1,537,553	3.2	94.9%

* YOLOv11n-cls and YOLOv8n-cls stats are from a small smoke run. Exact small-model numbers vary by configuration.

Results

The conclusions are as follows:

- YOLOv8 performed well as expected for the classification tasks proving it as a strong baseline for future performance comparison.
- YOLOv11 had a minor accuracy increase over YOLOv8 but used drastically less parameters for training.

Reasons for YOLOv7 failing:

- YOLOv7 Classification scripts incompatible with current PyTorch version.
- Missing model YAMLs and checkpoint formats caused errors.
- Was unable to obtain YOLOv7 classification checkpoints for training.
- Not trainable without patching the code.

Note

YOLOv7 cannot be trained as support for it ended in 2023.

YOLOv8 Smoke Test

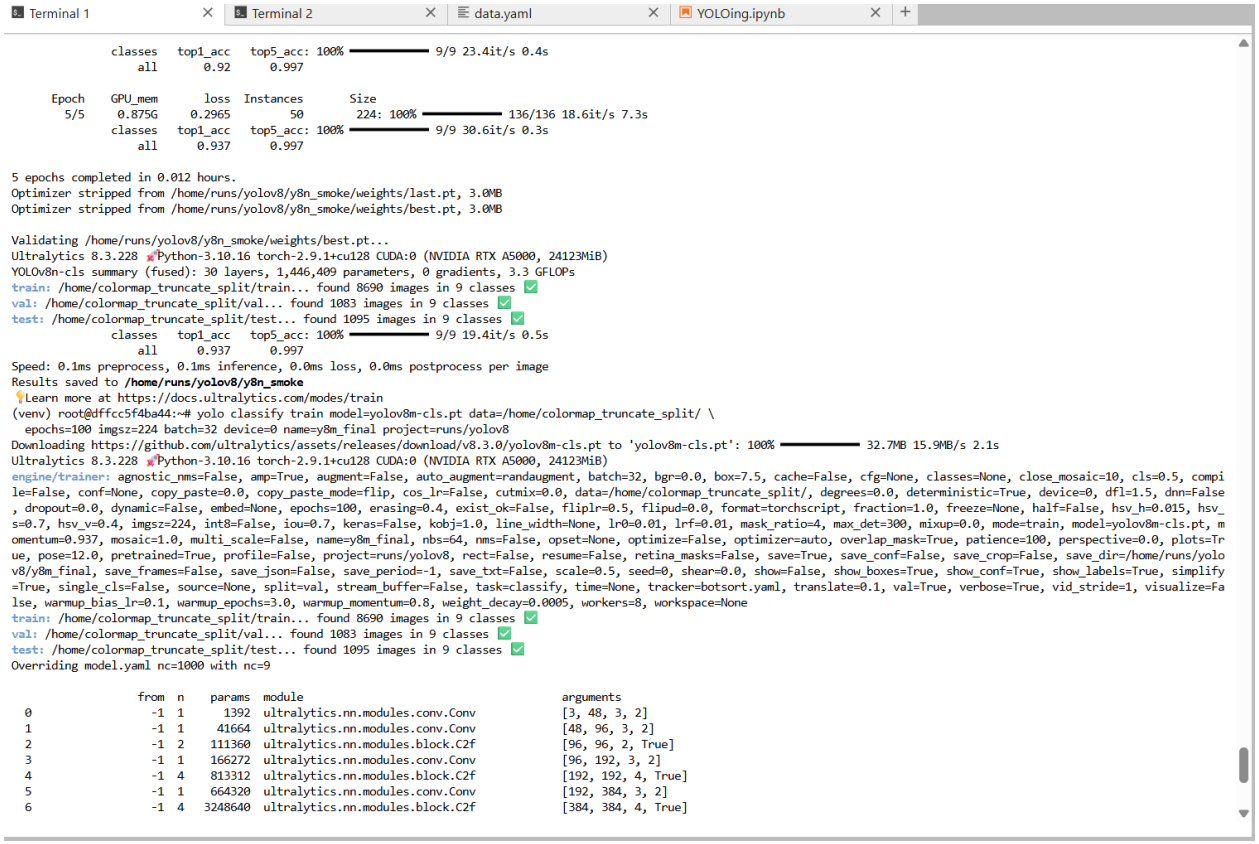
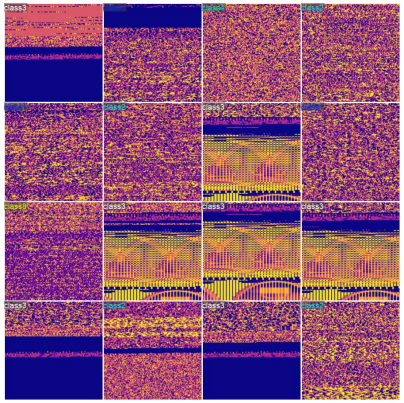


Figure: YOLOv8 Smoke Test to check if all systems are ready

YOLOv8 Smoke test Image Batches



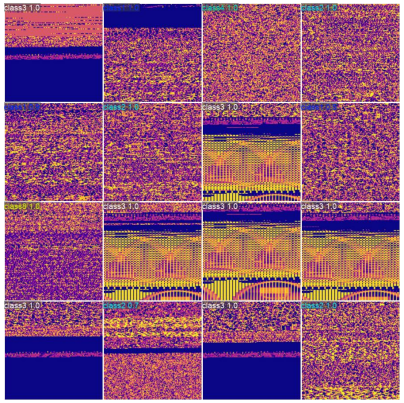
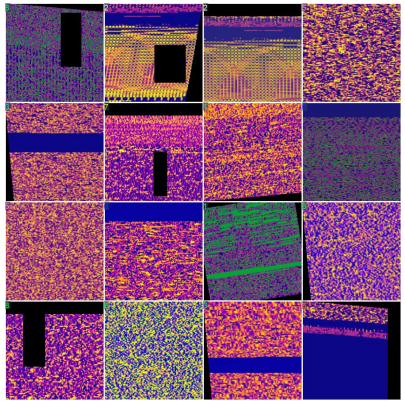


Figure: YOLOv8 validation batches.



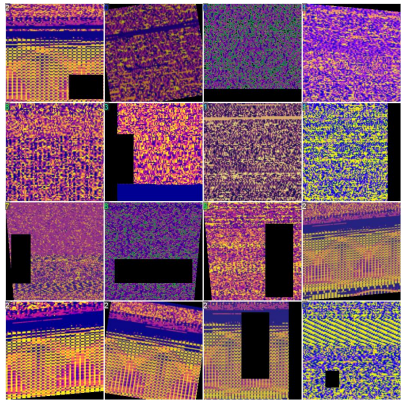


Figure: YOLOv8 training batches.

Confusion Matrix for YOLOv8 Smoke test

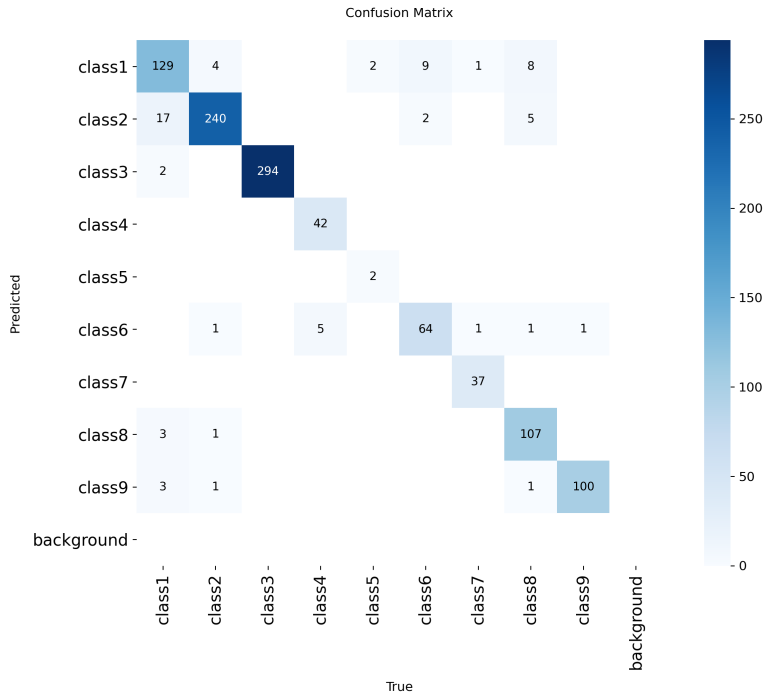


Figure: YOLOv8 confusion matrix normalized.

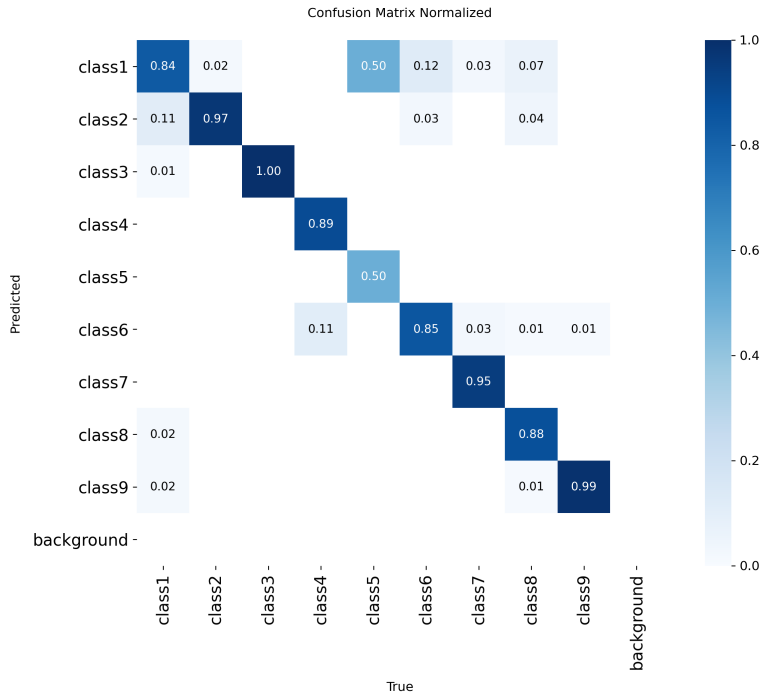


Figure: YOLOv8 confusion matrix normalized.

YOLOv8 Final Result

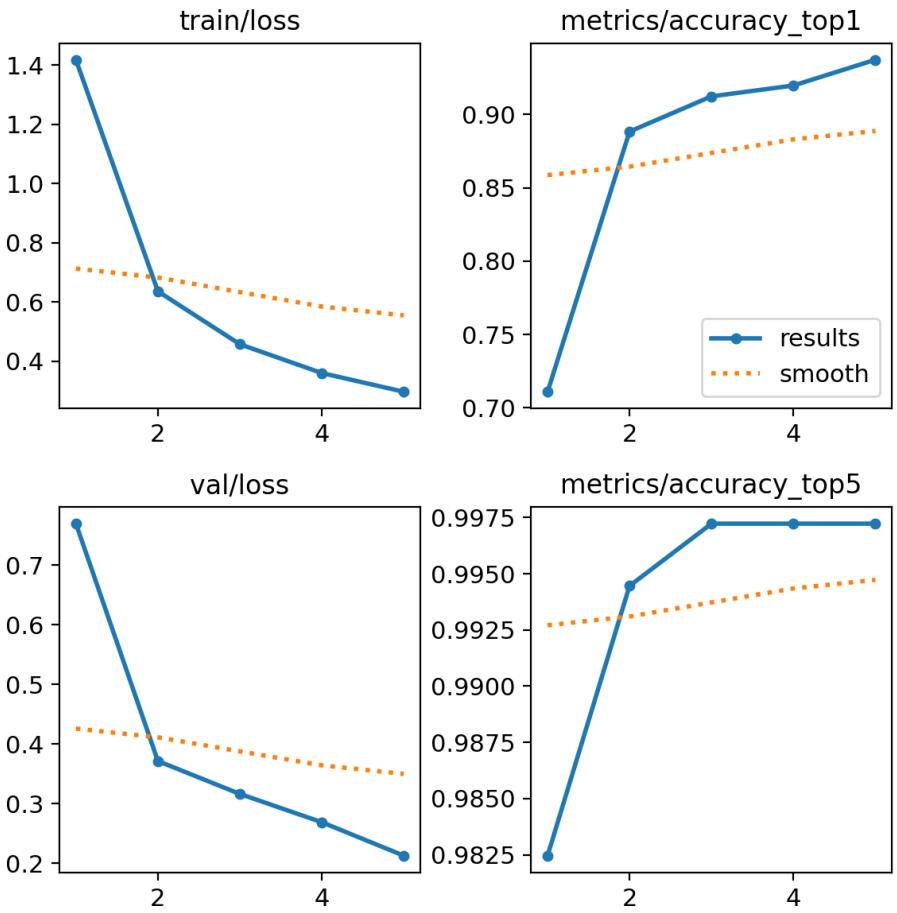


Figure: YOLOv8 smoke test result

YOLOv8 Final run

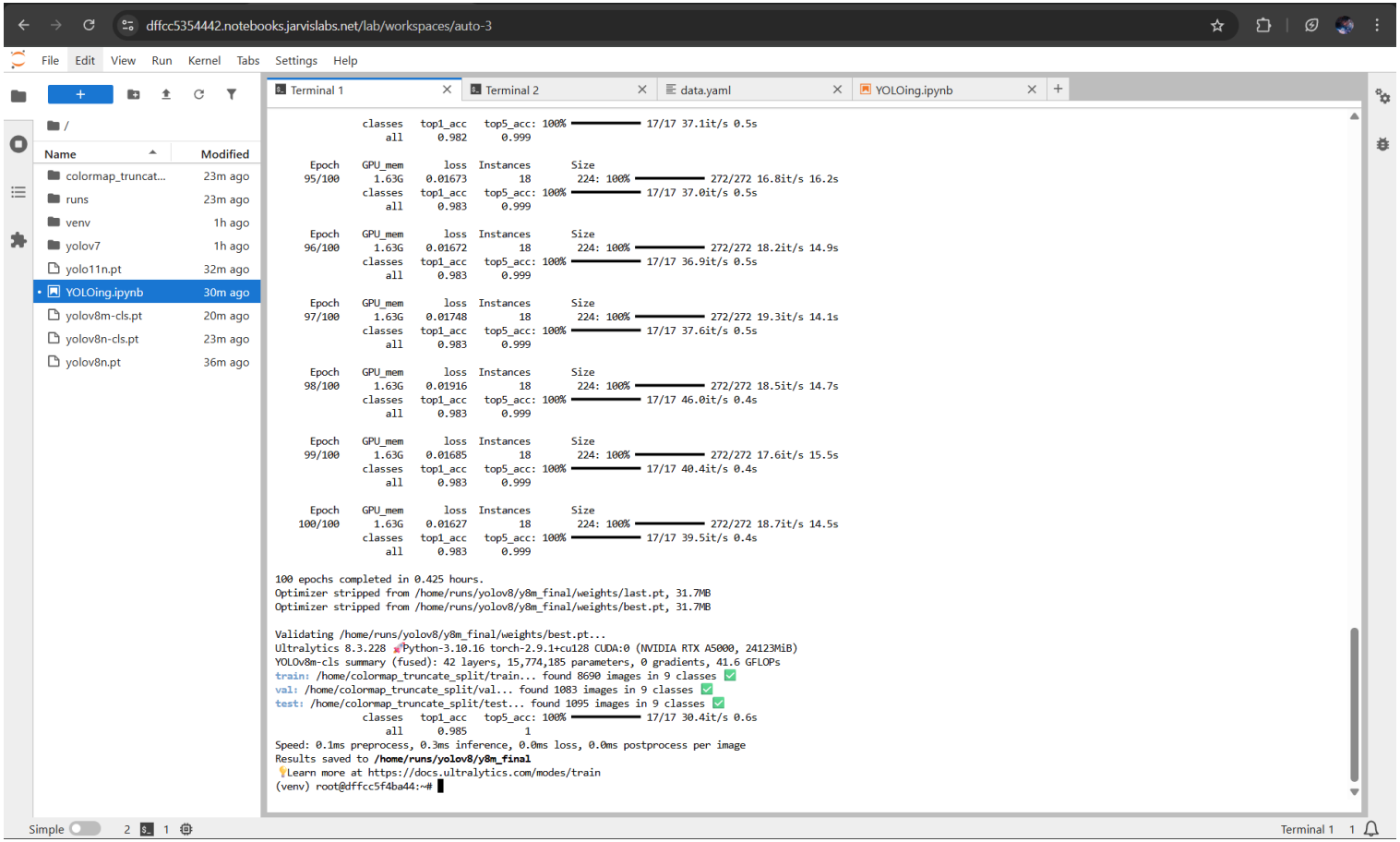
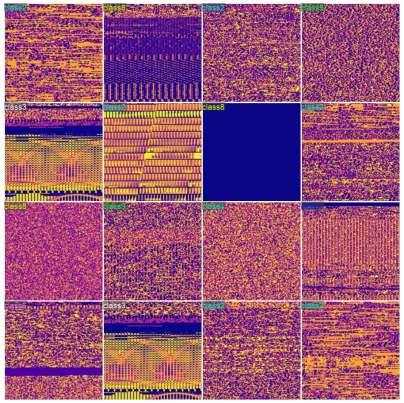


Figure: YOLOv11 - Trained for 100 epochs and achieved an accuracy of 98.5%.

YOLOv8 Final Image Batches



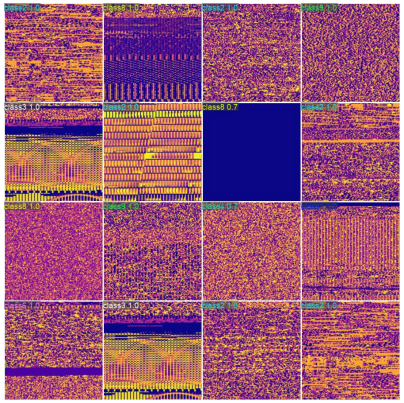
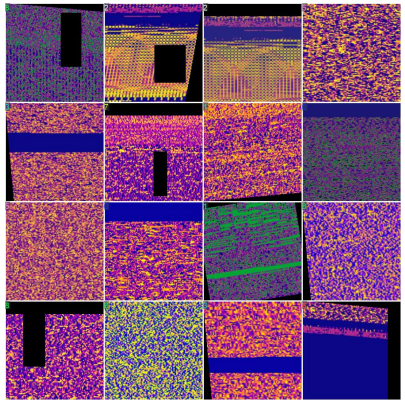


Figure: YOLOv8 validation batches.



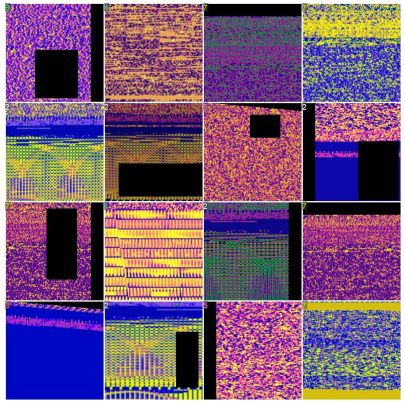


Figure: YOLOv8 training batches.

Confusion Matrix for YOLOv8

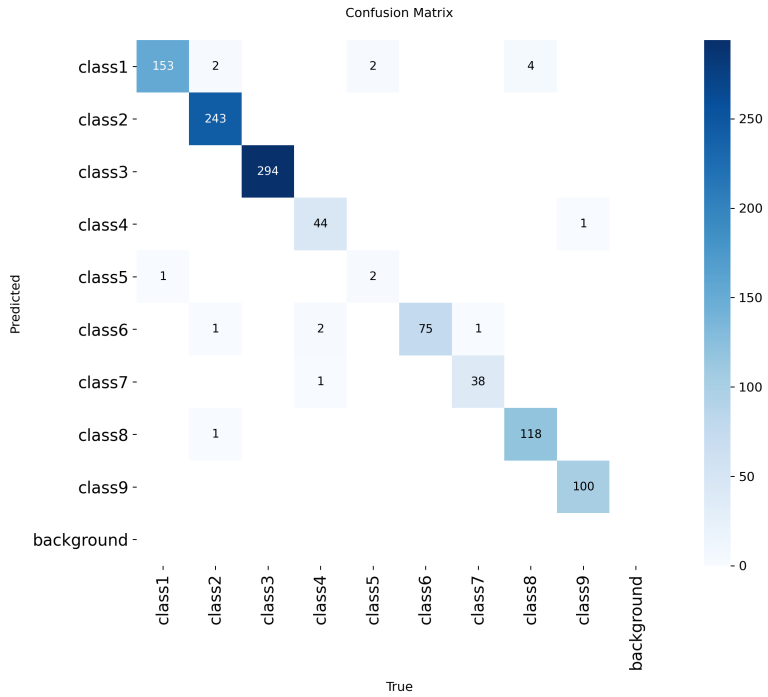


Figure: YOLOv8 confusion matrix normalized.

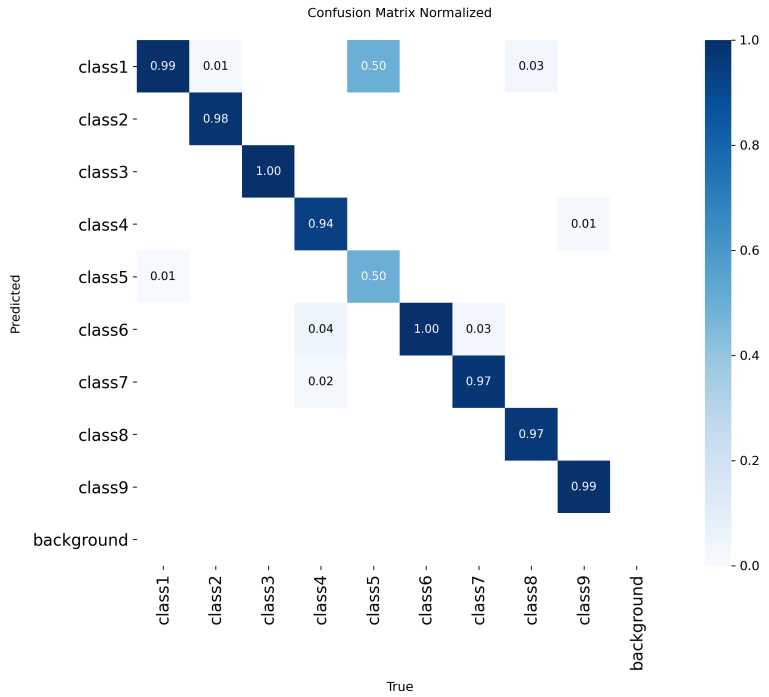


Figure: YOLOv8 confusion matrix normalized.

YOLOv8 Final Result

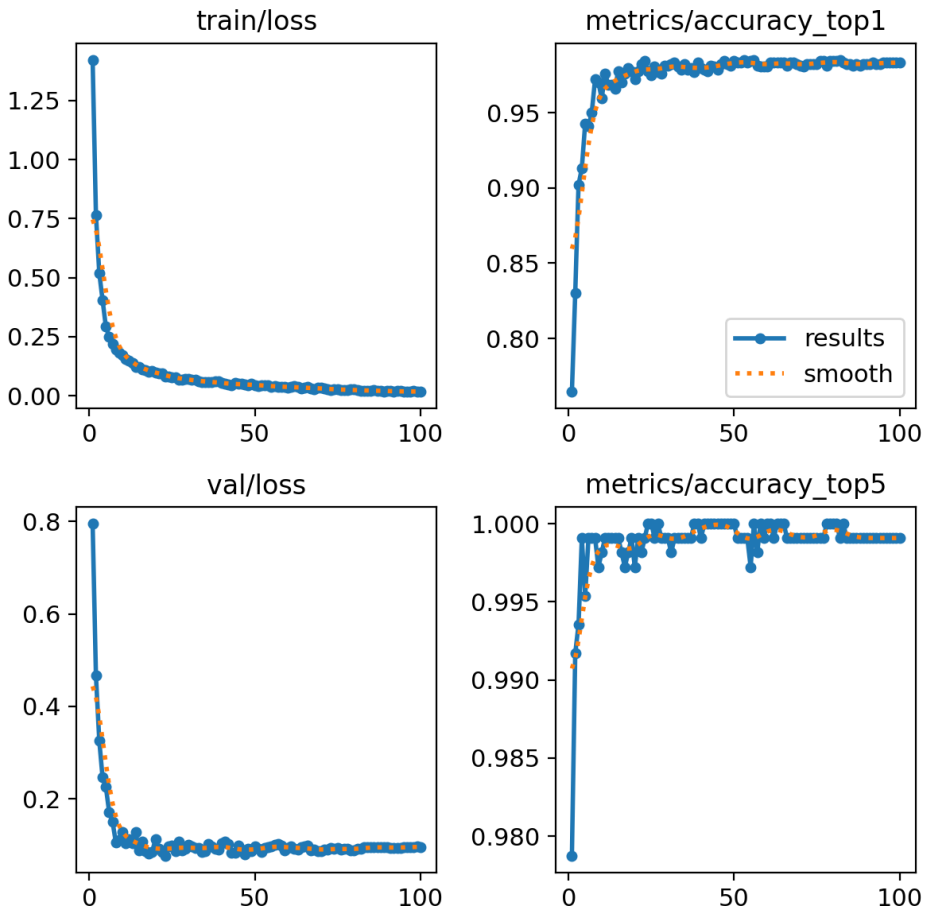


Figure: YOLOv8 Final result

YOLO 11 smoke test

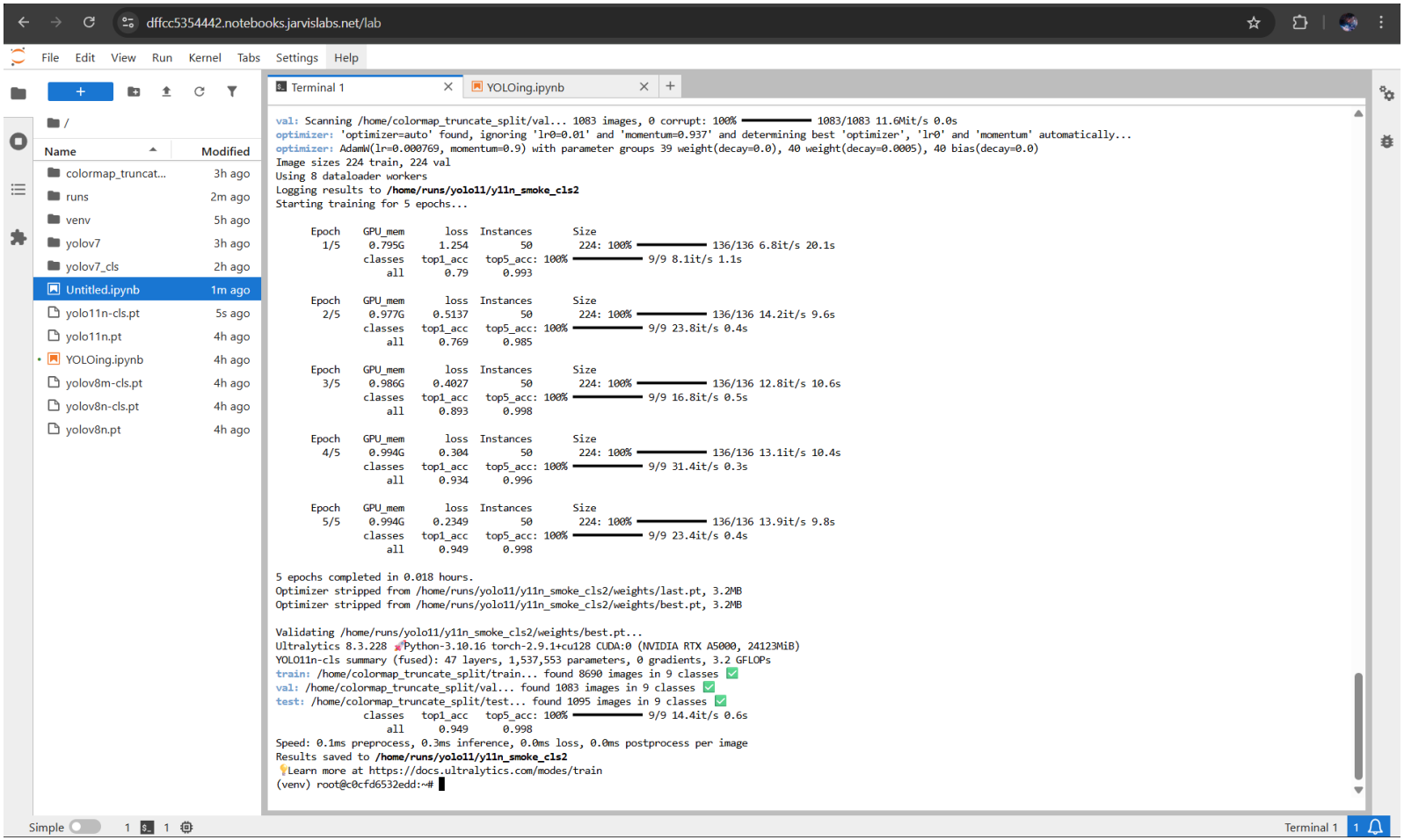
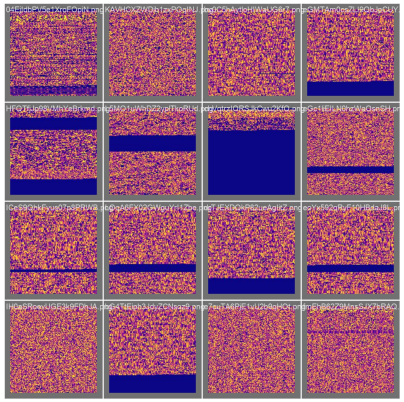


Figure: YOLOv11 smoke test to check if all systems are ready.

YOLOv11 Smoke test Image Batches



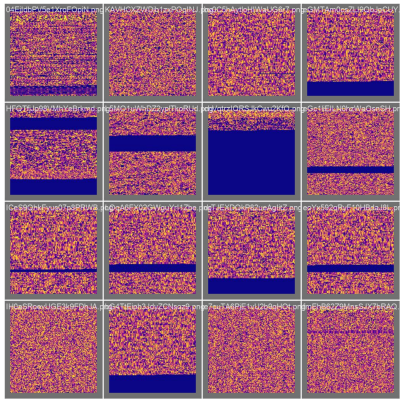
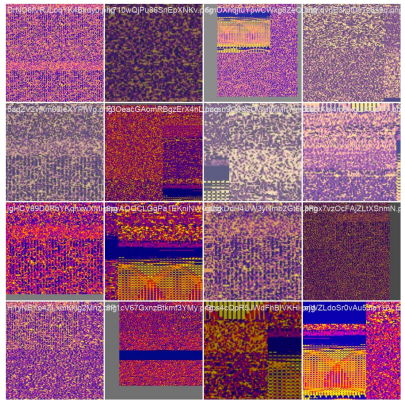


Figure: YOLOv11 validation batches.



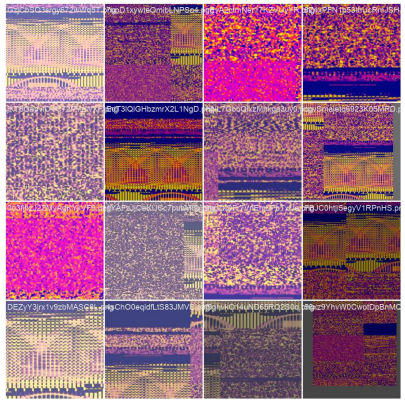


Figure: YOLOv11 training batches.

YOLO 11 Final Run

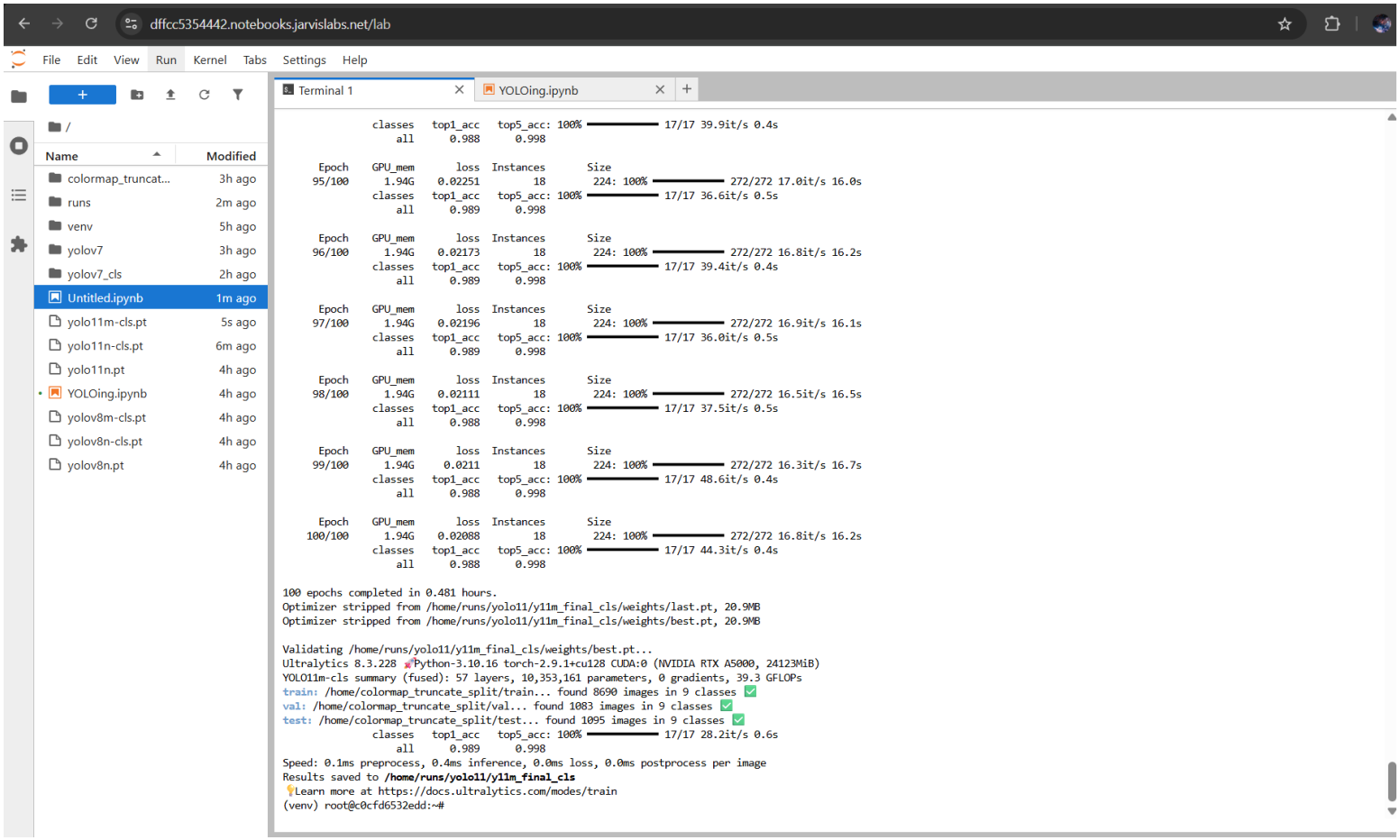
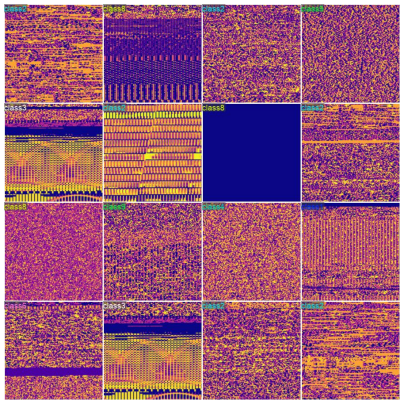


Figure: YOLOv11 - Trained for 100 epochs and achieved an accuracy of 98.9%.

YOLOv11 Final Image Batches



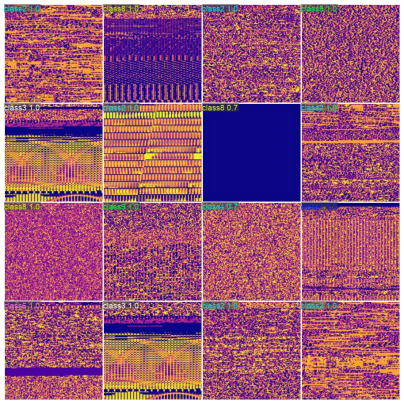
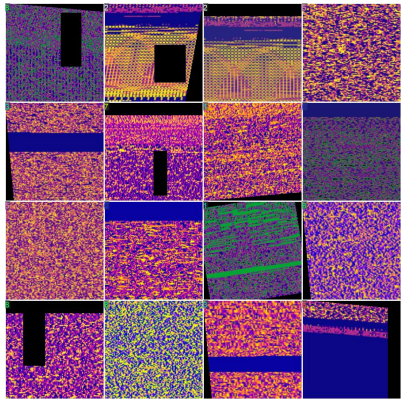


Figure: YOLOv11 validation batches.



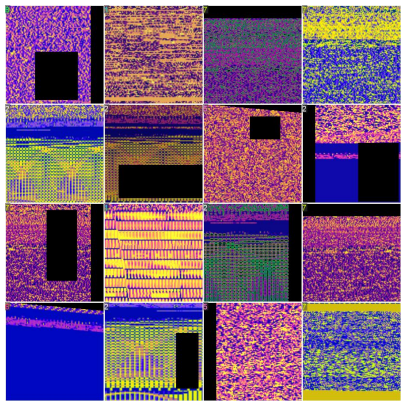


Figure: YOLOv11 training batches.

Confusion Matrix for YOLOv11

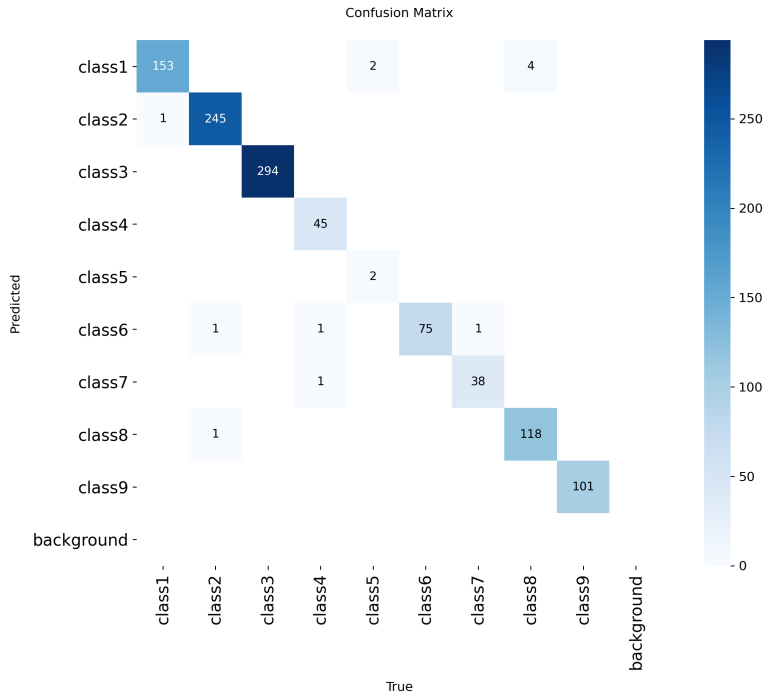


Figure: YOLOv11 confusion matrix normalized.

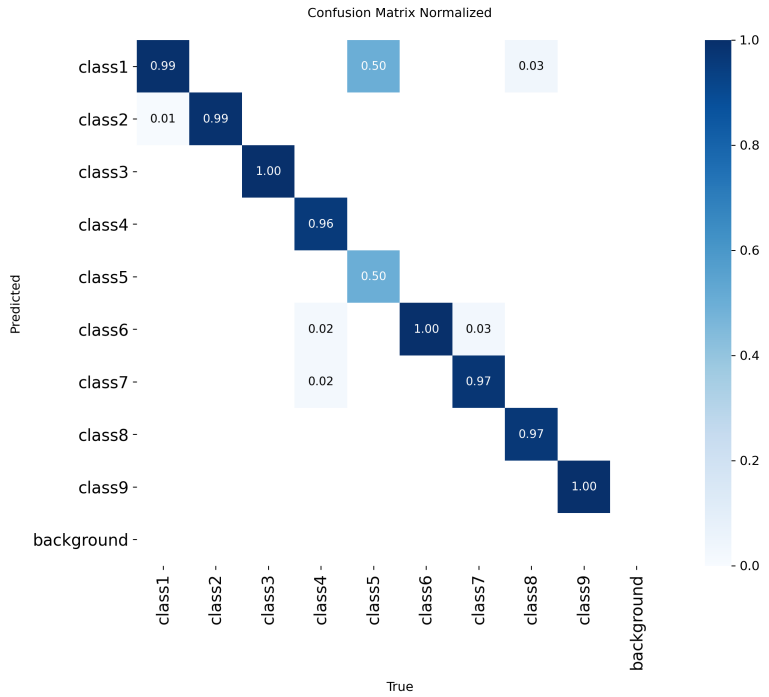


Figure: YOLOv11 confusion matrix normalized.

YOLOv11 final result

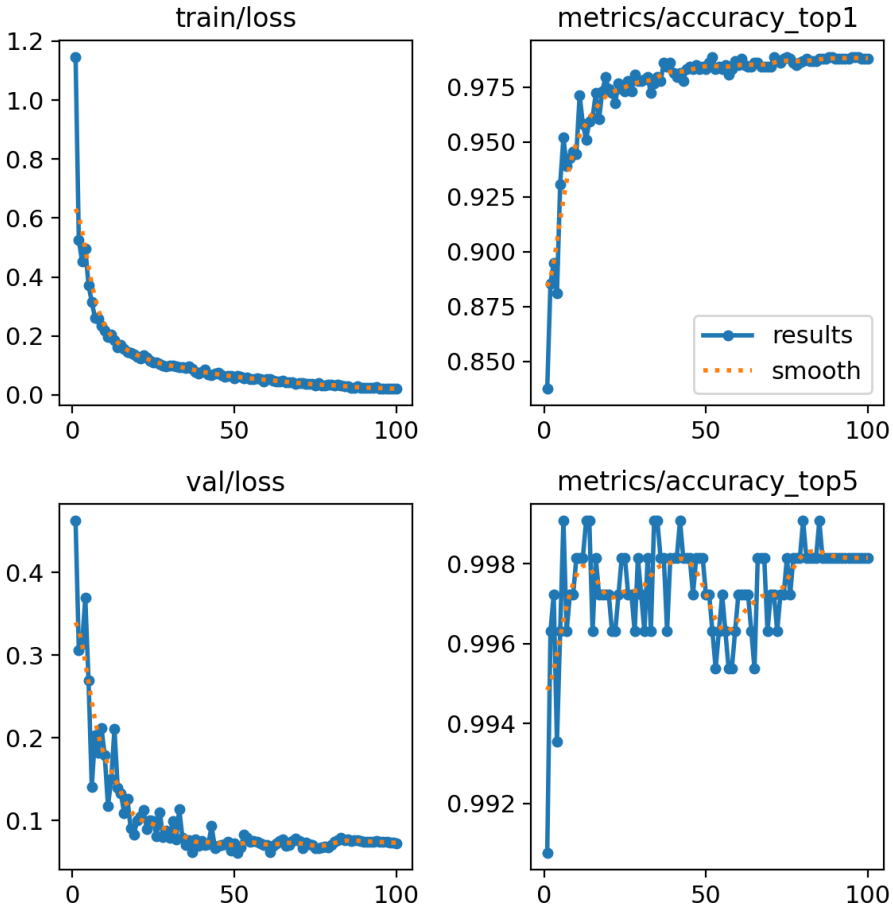


Figure: YOLOv11 results graphs