National College of Ireland

Configuration Manual

MSc Research Project
MSc in Data Analytics

Devaki Naga Venkata Prasanthi Vudiga
Student ID: x23223677

School of Computing
National College of Ireland

Supervisor: Harshani Nagahamulla

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	DEVAKI NAGA VENKATA PRASANTHI
Student ID:	X23223677
Programme:	MSc IN DATA ANALYTICS **Year:** 2024-25
Module:	MSc RESEARCH PROJECT
Lecturer:	HARSHANI NAGAHAMULLA
Submission Due Date:	29/01/2025
Project Title:	SIGN LANGUAGE RECOGNITION: A COMPARATIVE STUDY OF DEEP LEARNING MODELS USING YOLOv8,RT-DETR, and Faster R-CNN
Word Count:	1534 **Page Count:** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: VDNVPRASANTHI

Date: 29/01/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

DEVAKI NAGA VENKATA PRASANTHI VUDIGA
Student ID: X23223677

1 Introduction

1.1 Brief Overview of the Project

This project implements an Indian Sign Language (ISL) Recognition system that compares three deep learning architectures:

- RT-DETR (Real-Time Detection Transformer)
- YOLOv8
- Faster R-CNN

The system processes a dataset of 4,410 ISL images spanning 35 classes (numbers 1-9 and letters A-Z) and evaluates each architecture's performance across different configurations and hyperparameters.

1.2 Purpose of the Configuration Manual

This manual is designed to:

1. Guide users through the complete setup and implementation process
2. Provide detailed configuration instructions for each architecture
3. Help users understand and optimize model performance
4. Serve as a reference for troubleshooting common issues

The manual covers:

- Environment setup and requirements
- Dataset preparation and organization
- Model configuration and training
- Hyperparameter tuning procedures
- Evaluation methods
- Inference and testing processes

This documentation ensures reproducible results and consistent implementation across different setups and configurations.

2 System Requirements

2.1 Hardware Requirements

- **Minimum Specifications**
  - **GPU**: NVIDIA A100 GPU with 40GB VRAM
  - **Storage**: 50GB free space on Google Drive
  - **Internet**: Stable broadband connection (10 Mbps minimum)
  - **RAM**: 32GB system memory recommended
  - **CPU**: Multi-core processor supporting CUDA operations
- **Recommended Specifications**

  - **GPU**: NVIDIA A100 GPU with 80GB VRAM for larger batch sizes
  - **Storage**: 100GB+ free space on Google Drive
  - **Internet**: High-speed connection (50+ Mbps)

- **RAM**: 64GB system memory for optimal performance
- **CPU**: Latest generation multi-core processor

2.2 Software Requirements

Operating System

- Any OS compatible with Google Chrome browser
- Google Chrome (Version 90+) recommended

Development Environment

- Google Colab Pro+ subscription
- Google Drive account with sufficient storage
- Python 3.8 or higher

Required Libraries and Versions

- Core Libraries
    - torch==2.1.0
    - torchvision==0.16.0
    - ultralytics==8.3.40
    - PyYAML==6.0.1

- mage Processing
    - opencv-python==4.8.0
    - Pillow==9.5.0

- Data Processing
    - pandas==2.0.3
    - numpy==1.24.3

- Visualization
    - matplotlib==3.7.1
    - seaborn==0.12.2
- Additional Tools
    - tensorboard==2.14.0
    - GPUtil==1.4.0

Additional Tools

- Google Drive File Stream for local access
- Jupyter Notebook environment (provided by Colab)

2.3 Environment Setup Requirements

1. Google Account with:
    - Access to Google Colab Pro+
    - Sufficient Google Drive storage

    - Permission to mount Drive in Colab
2. Browser Requirements:
    - JavaScript enabled
    - Cookie permissions for Colab
3. Network Requirements:
    - Stable internet connection
    - Ability to maintain long-running Colab sessions
    - Access to Google services

2.4 Compatibility Notes

- Models tested on NVIDIA A100 GPU
- Code compatible with PyTorch 2.0+
- CUDA 11.0 or higher required for GPU operations
- System designed for Google Colab environment

3 Connecting to Google Colab and Drive Setup for ISL Recognition System

3.1 Initial Setup

3.1.1 Colab Access

1. Visit Google Colab by navigating to colab.research.google.com
2. Sign in with your Google Account (Colab Pro+ subscription required)
3. Upload the notebook file of the

3.1.2 GPU Configuration

1. For each notebook, configure GPU settings:
    - From the top menu, select Runtime, then Change runtime type
    - In the hardware accelerator dropdown, select GPU
    - Choose A100 GPU with High-RAM configuration
    - Save your selection

3.2 Project Structure Setup

3.2.1 Google Drive Organization

Create a main project folder called "NCI_Sep_Thesis_2024" in your Google Drive. Inside this, create a subfolder "Devaki_NCI_RIC". Within this folder, create four essential subfolders:

- ISL_Devaki_Dataset: For storing all dataset files
- training_results: For storing training outputs
- trained_models: For storing trained model weights
- hyperparameter_tuning: For tuning results

3.2.2 Model File Organization

In the training_results folder, organize your model files in onen separate folders:

- RT-DETR folder containing best.pt
- YOLO folder containing best.pt
- Faster R-CNN folder containing best_model_epoch_10.pth

3.3 Dataset Organization

3.3.1 Dataset Structure

Create three main subfolders within ISL_Devaki_Dataset:

- train folder: Contains training images and labels
- valid folder: Contains validation images and labels
- test folder: Contains test images and labels Each subfolder should have an 'images' and 'labels' directory. Place a data.yaml file in the root of ISL_Devaki_Dataset.

3.3.2 System Verification

Essential Checks

1. Verify GPU availability:
    - Confirm A100 GPU access
    - Check 40GB VRAM availability
2. Verify storage space:
    - Ensure minimum 50GB free space
    - Confirm access to all model weights
    - Verify dataset accessibility
3. Verify drive mounting:
    - Check proper authentication
    - Confirm read/write permissions
    - Test path accessibility

3.3.3 Troubleshooting Guide

Common Issues and Solutions

1. If drive mounting fails:
    - Try re-authenticating
    - Check your internet connection
    - Verify your drive permissions
2. If GPU is not available:
    - Verify your Colab Pro+ subscription
    - Try switching the runtime type
    - Restart the runtime
3. For memory issues:
    - Clear your runtime
    - Try reducing batch sizes
    - Verify file integrity

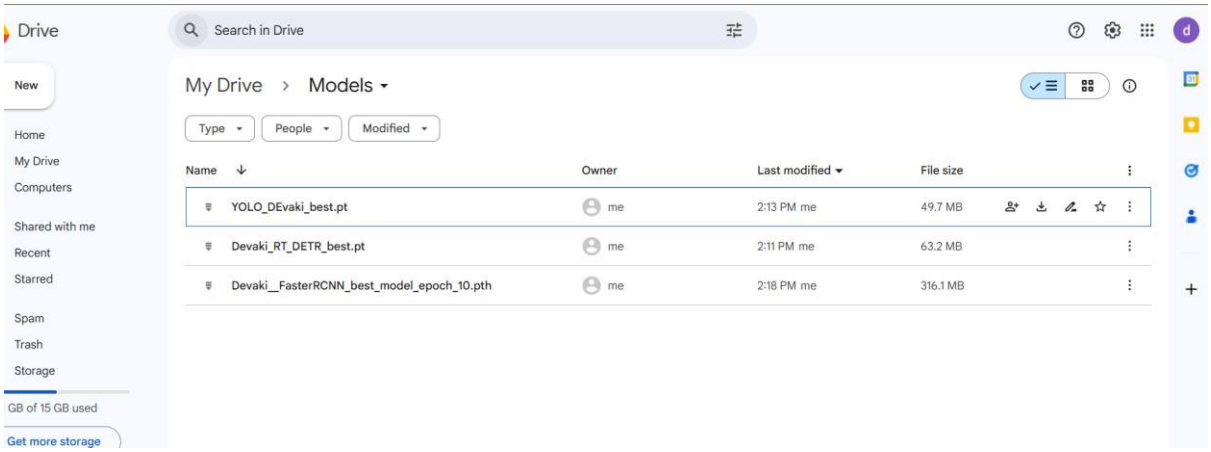
4 Update the Model path in the Demo.ipynb File

4.1 Running the Experiment

- **Step 1: Model File Setup**
- **Required Model Files**

Place the trained models in your Google Drive:

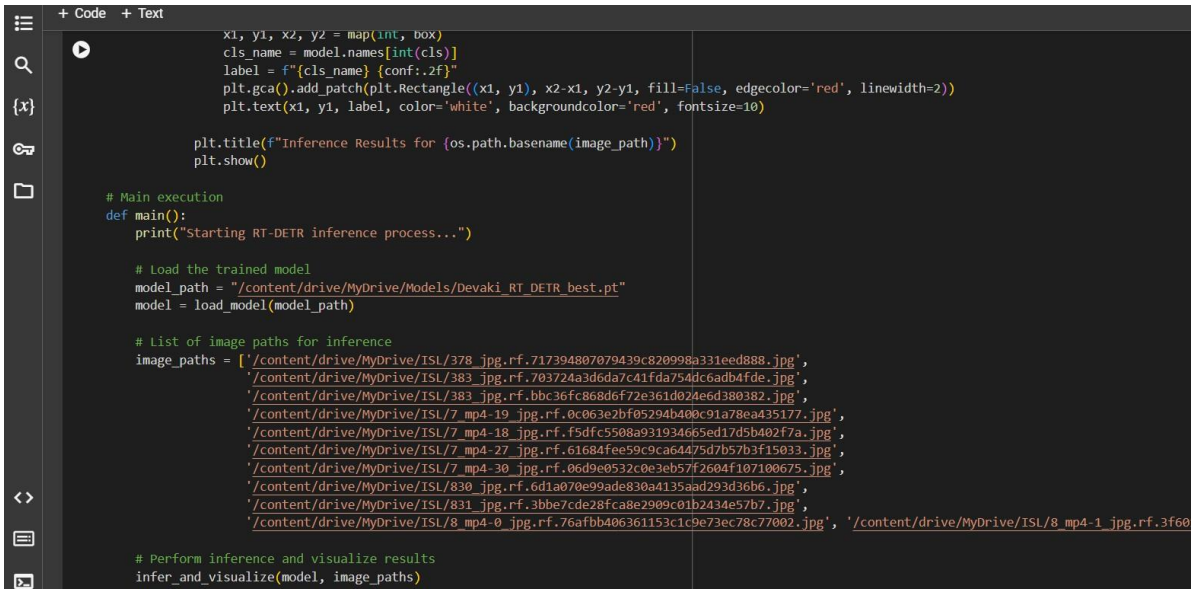
- RT-DETR model: Models/Devaki_RT_DETR_best.pt
- YOLOv8 model: Models/YOLO_DEvaki_best.pt
- Faster R-CNN model: Models/Devaki__FasterRCNN_best_model_epoch_10.pth

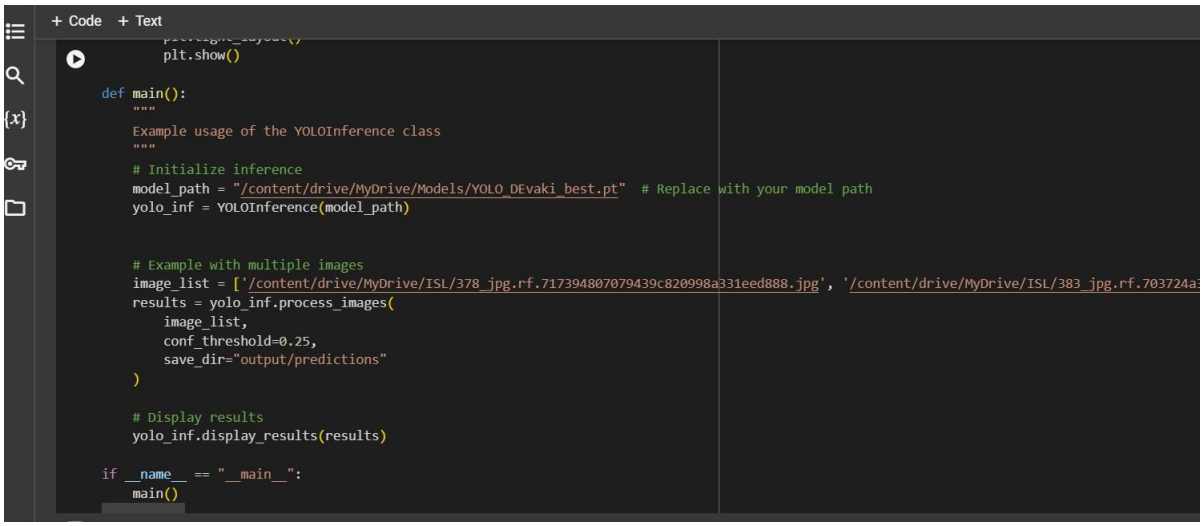


4.2 Model Path Configuration

Update the following paths in the code as per the directory placed.
For RT-DETR: model_path = "/content/drive/MyDrive/Models/Devaki_RT_DETR_best.pt"
For YOLOv8: model_path = "/content/drive/MyDrive/Models/YOLO_DEvaki_best.pt"
For Faster R-CNN: model_path = "/content/drive/MyDrive/Models/Devaki__FasterRCNN_best_model_epoch_10.pth"

update the model_path as per the path if changed above directory names.





```
+ Code + Text
        plt.show()

    def main():
        """
        Example usage of the YOLOInference class
        """
        # Initialize inference
        model_path = "/content/drive/MyDrive/Models/YOLO_DEvaki_best.pt"  # Replace with your model path
        yolo_inf = YOLOInference(model_path)


        # Example with multiple images
        image_list = ['/content/drive/MyDrive/ISL/378_jpg.rf.717394807079439c820998a331eed888.jpg', '/content/drive/MyDrive/ISL/383_jpg.rf.703724a
        results = yolo_inf.process_images(
            image_list,
            conf_threshold=0.25,
            save_dir="output/predictions"
        )

        # Display results
        yolo_inf.display_results(results)

    if __name__ == "__main__":
        main()
```

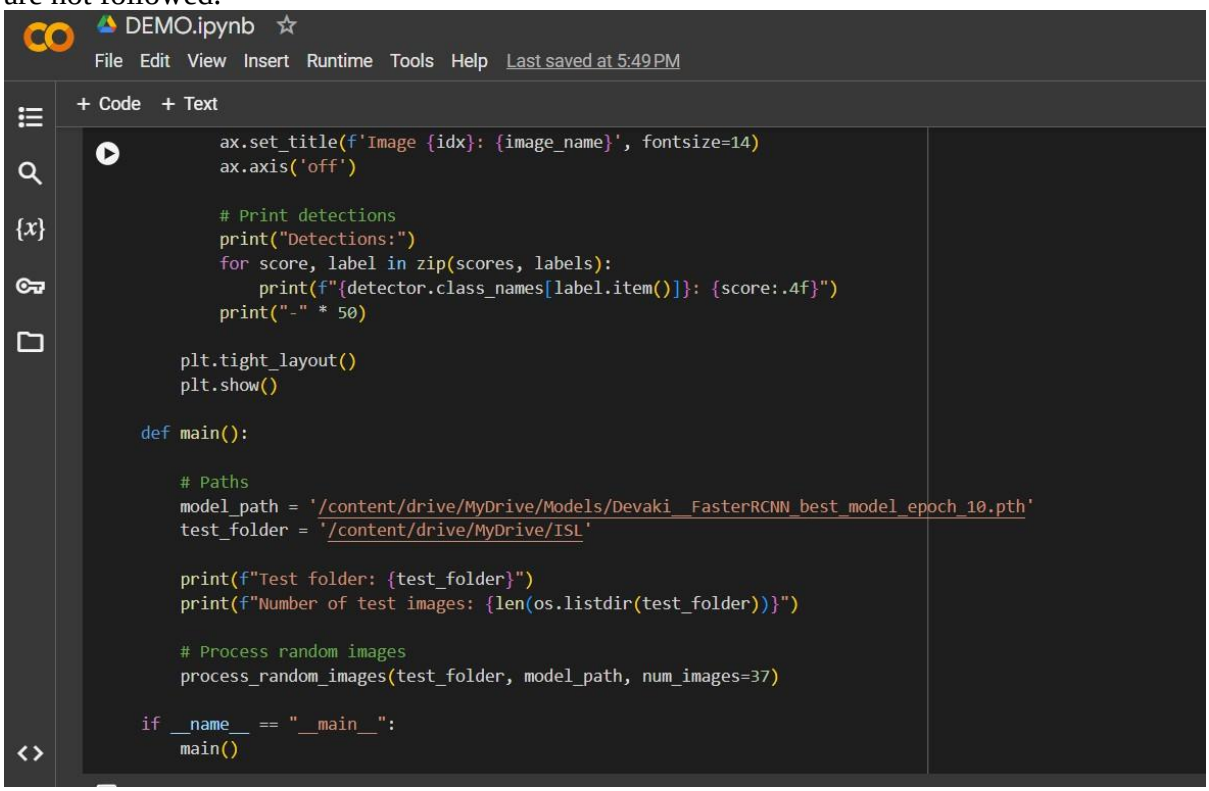
4.3 Image Input Setup

Individual Images

1. Create a folder named 'ISL' in your Google Drive
2. Upload test images to this folder
3. Update image paths in the code:

folder_path = '/content/drive/MyDrive/ISL'{sample Path}

same here,chang ethe model path and image directory path as per their location,if same names are not followed.



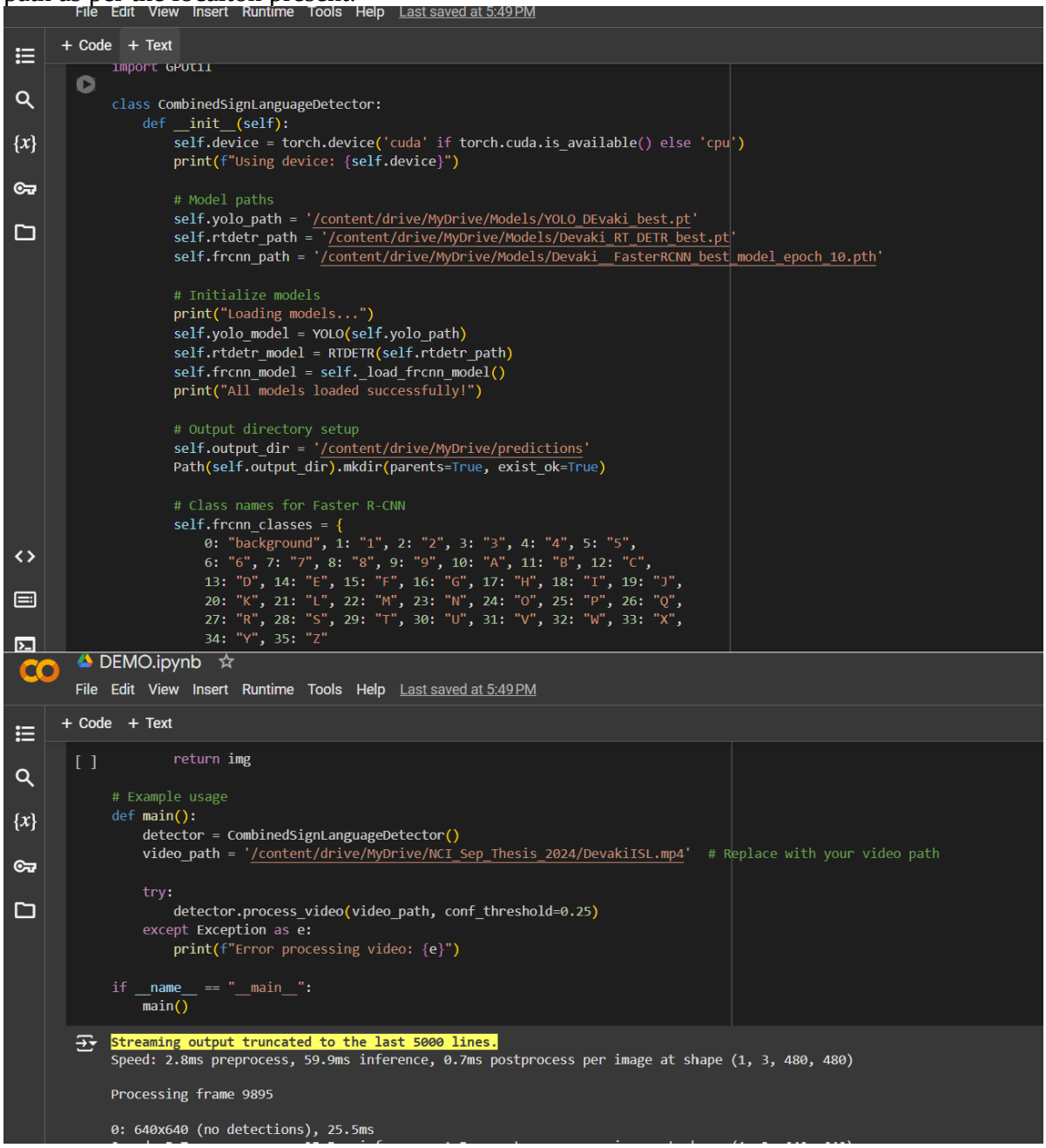
```
DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 5:49 PM
+ Code + Text
            ax.set_title(f'Image {idx}: {image_name}', fontsize=14)
            ax.axis('off')

            # Print detections
            print("Detections:")
            for score, label in zip(scores, labels):
                print(f"{detector.class_names[label.item()]}: {score:.4f}")
            print("-" * 50)

        plt.tight_layout()
        plt.show()

    def main():

        # Paths
        model_path = '/content/drive/MyDrive/Models/Devaki__FasterRCNN_best_model_epoch_10.pth'
        test_folder = '/content/drive/MyDrive/ISL'

        print(f"Test folder: {test_folder}")
        print(f"Number of test images: {len(os.listdir(test_folder))}")

        # Process random images
        process_random_images(test_folder, model_path, num_images=37)

    if __name__ == "__main__":
        main()
```

Video Input

1. Upload your test video to Google Drive
2. Update video path:

video_path = '/content/drive/MyDrive/NCI/DevakiISL.mp4'{sample Path}

Update the Three model paths corresponding to the directories they are available and vidoeo path as per the locaiton present.



```
import GPUtil

class CombinedSignLanguageDetector:
    def __init__(self):
        self.device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
        print(f"Using device: {self.device}")

        # Model paths
        self.yolo_path = '/content/drive/MyDrive/Models/YOLO_DEvaki_best.pt'
        self.rtdetr_path = '/content/drive/MyDrive/Models/Devaki_RT_DETR_best.pt'
        self.frcnn_path = '/content/drive/MyDrive/Models/Devaki__FasterRCNN_best_model_epoch_10.pth'

        # Initialize models
        print("Loading models...")
        self.yolo_model = YOLO(self.yolo_path)
        self.rtdetr_model = RTDETR(self.rtdetr_path)
        self.frcnn_model = self._load_frcnn_model()
        print("All models loaded successfully!")

        # Output directory setup
        self.output_dir = '/content/drive/MyDrive/predictions'
        Path(self.output_dir).mkdir(parents=True, exist_ok=True)

        # Class names for Faster R-CNN
        self.frcnn_classes = {
            0: "background", 1: "1", 2: "2", 3: "3", 4: "4", 5: "5",
            6: "6", 7: "7", 8: "8", 9: "9", 10: "A", 11: "B", 12: "C",
            13: "D", 14: "E", 15: "F", 16: "G", 17: "H", 18: "I", 19: "J",
            20: "K", 21: "L", 22: "M", 23: "N", 24: "O", 25: "P", 26: "Q",
            27: "R", 28: "S", 29: "T", 30: "U", 31: "V", 32: "W", 33: "X",
            34: "Y", 35: "Z"
```

DEMO.ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:49 PM

+ Code + Text

```
[ ]         return img

    # Example usage
    def main():
        detector = CombinedSignLanguageDetector()
        video_path = '/content/drive/MyDrive/NCI_Sep_Thesis_2024/DevakiISL.mp4'  # Replace with your video path

        try:
            detector.process_video(video_path, conf_threshold=0.25)
        except Exception as e:
            print(f"Error processing video: {e}")

    if __name__ == "__main__":
        main()
```

```
Streaming output truncated to the last 5000 lines.
Speed: 2.8ms preprocess, 59.9ms inference, 0.7ms postprocess per image at shape (1, 3, 480, 480)

Processing frame 9895

0: 640x640 (no detections), 25.5ms
```

4.4 Output Directory Setup

4.4.1 Create directories for results:

/content/drive/MyDrive/predictions/

- **Step 4: Running Inference**
- **For Images**
  - Run RT-DETR inference:
    - Execute RT-DETR section
    - Expected output: Visualizations with bounding boxes and class labels
  - Run YOLOv8 inference:
    - Execute YOLOv8 section

        - Expected output: Grid visualization of detections
    - Run Faster R-CNN inference:
        - Execute Faster R-CNN section
        - Expected output: Detection visualizations with confidence scores
- **For Video**
    - Run combined video inference:
        - Execute video inference section
        - Expected outputs:
            - Individual model predictions
            - Combined visualization grid
            - Performance metrics

4.5 Verification

- **Check Output Files**
    - Image results:
        - Look for detection visualizations
        - Verify bounding boxes and labels
        - Check confidence scores
    - Video results:
        - Verify four output videos:
            - YOLO predictions
            - RT-DETR predictions
            - Faster R-CNN predictions
            - Combined grid view
- **Verify Performance**
- Check console output for:
    - Processing time
    - FPS (Frames Per Second)
    - GPU usage metrics
    - Memory consumption

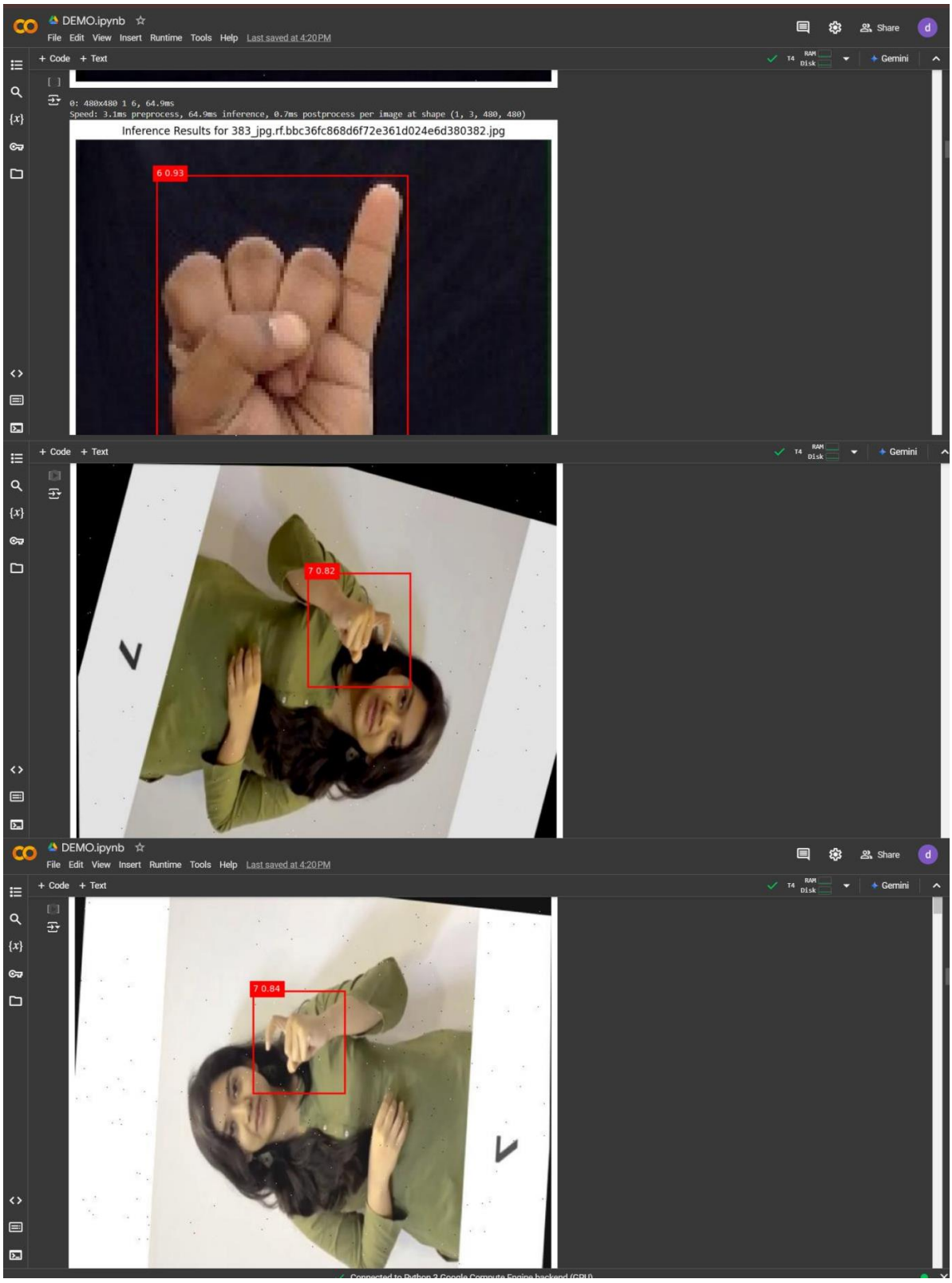
4.6 Troubleshooting

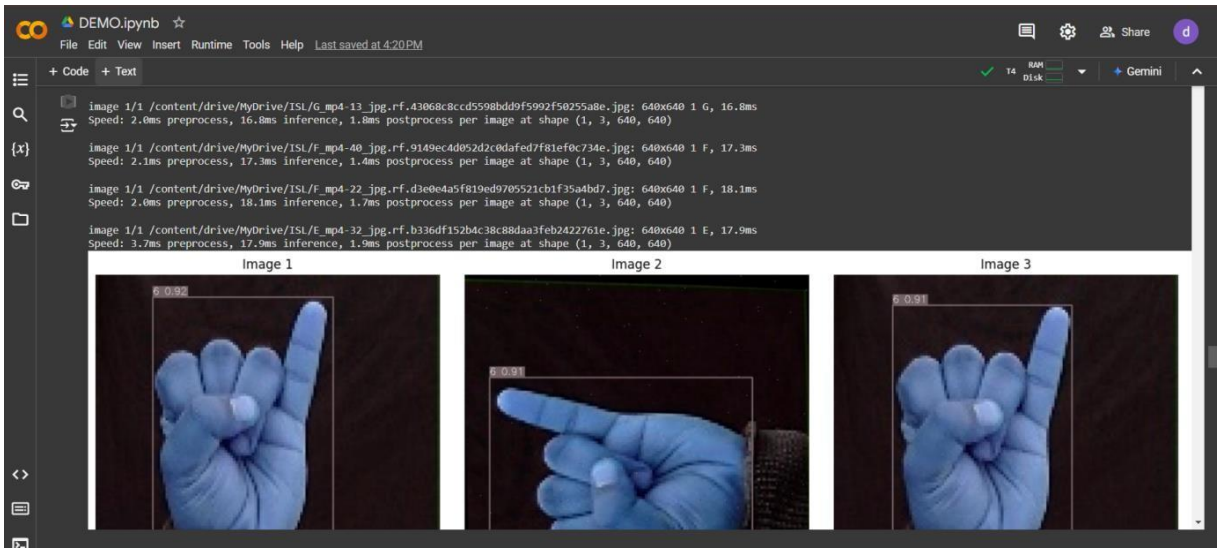
Common Issues:

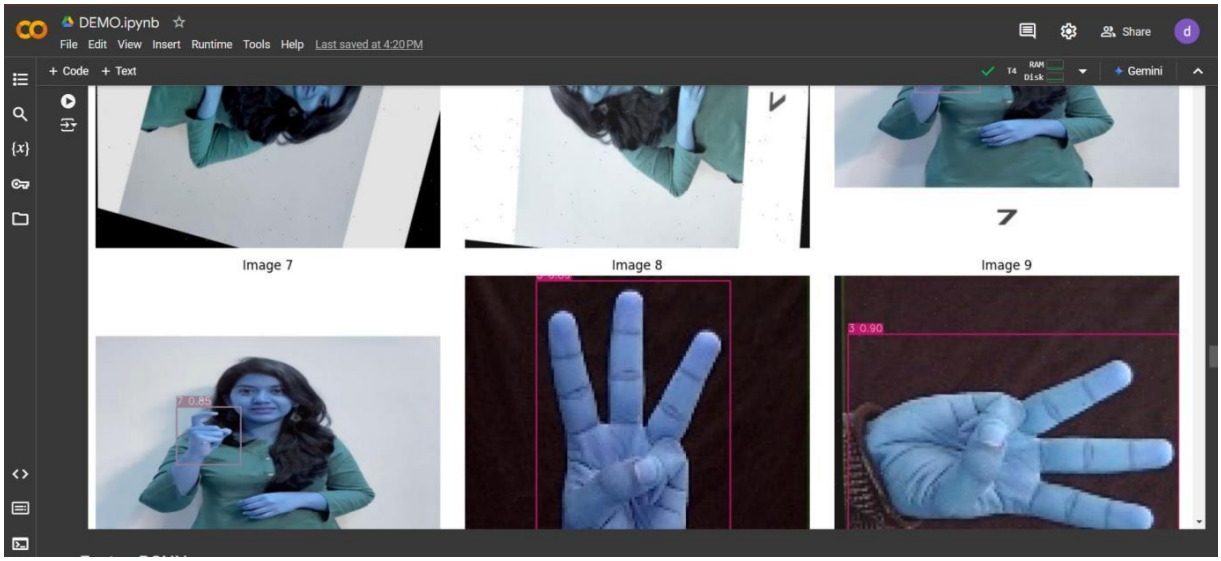
1. File not found errors:
    - Verify model paths
    - Check image/video paths
    - Confirm directory structure
2. Memory issues:
    - Reduce batch size
    - Process fewer images
    - Clear runtime memory
3. GPU errors:
    - Verify GPU availability
    - Monitor GPU memory
    - Restart runtime if needed

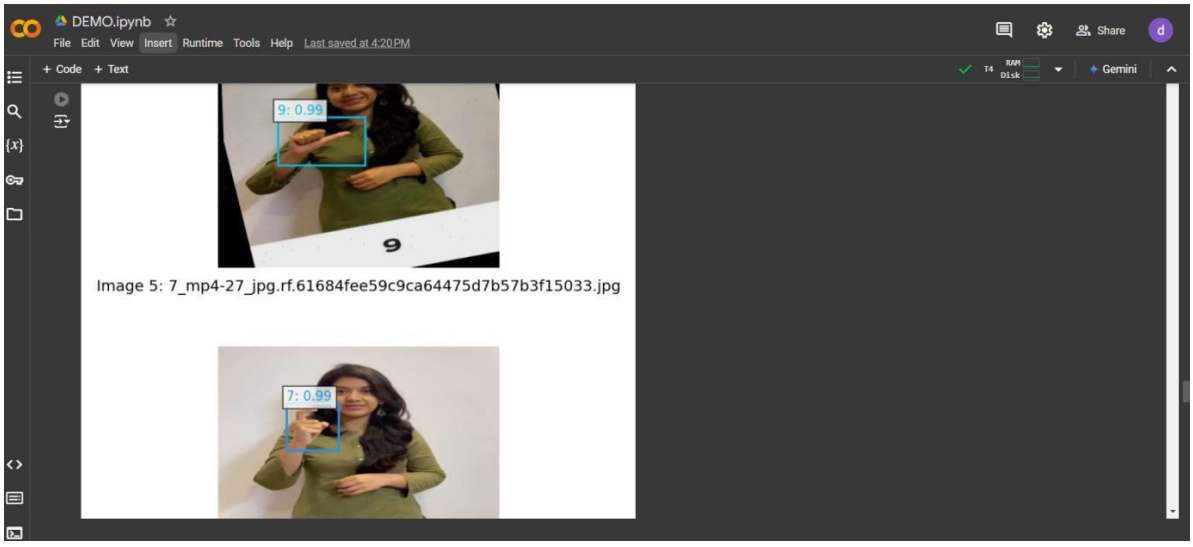
Follow these steps sequentially to ensure proper execution of the ISL recognition system.

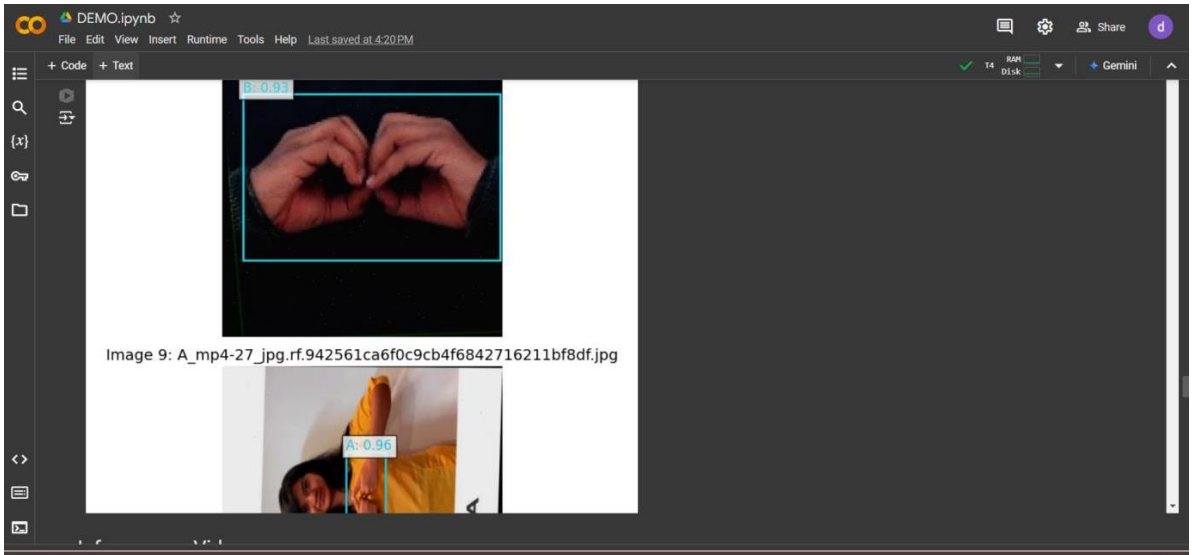
5 Results :

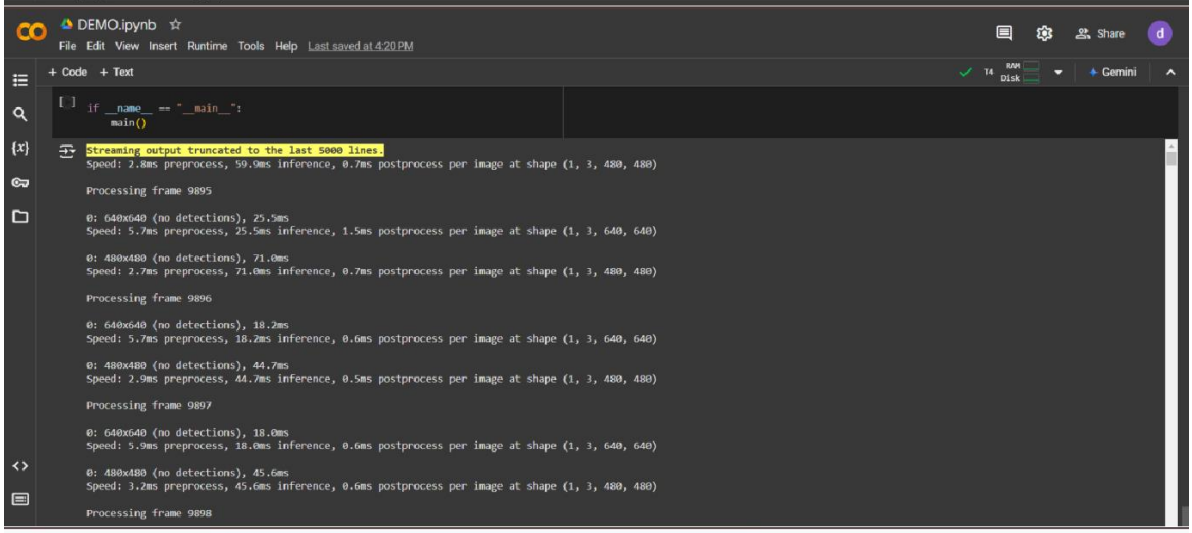

DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
0: 480x480 1 6, 64.9ms
Speed: 3.1ms preprocess, 64.9ms inference, 0.7ms postprocess per image at shape (1, 3, 480, 480)
Inference Results for 383_jpg.rf.bbc36fc868d6f72e361d024e6d380382.jpg
6 0.93
7 0.82
DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
7 0.84

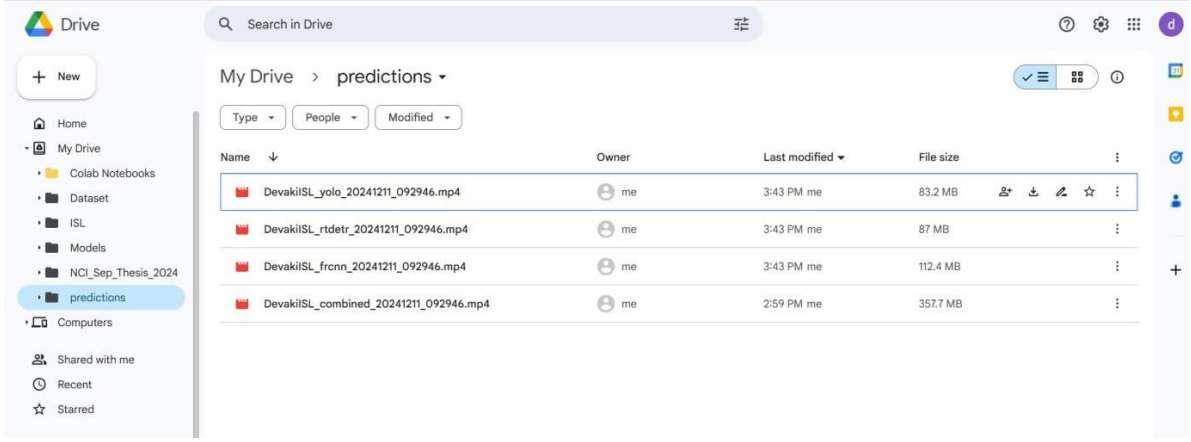

DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
image 1/1 /content/drive/MyDrive/ISL/G_mp4-13_jpg.rf.43068c8ccd5598bdd9f5992f50255a8e.jpg: 640x640 1 G, 16.8ms
Speed: 2.0ms preprocess, 16.8ms inference, 1.8ms postprocess per image at shape (1, 3, 640, 640)
image 1/1 /content/drive/MyDrive/ISL/F_mp4-40_jpg.rf.9149ec4d052d2c0dafed7f81ef0c734e.jpg: 640x640 1 F, 17.3ms
Speed: 2.1ms preprocess, 17.3ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 640)
image 1/1 /content/drive/MyDrive/ISL/F_mp4-22_jpg.rf.d3e0e4a5f819ed9705521cb1f35a4bd7.jpg: 640x640 1 F, 18.1ms
Speed: 2.0ms preprocess, 18.1ms inference, 1.7ms postprocess per image at shape (1, 3, 640, 640)
image 1/1 /content/drive/MyDrive/ISL/E_mp4-32_jpg.rf.b336df152b4c38c88daa3feb2422761e.jpg: 640x640 1 E, 17.9ms
Speed: 3.7ms preprocess, 17.9ms inference, 1.9ms postprocess per image at shape (1, 3, 640, 640)
Image 1
Image 2
Image 3

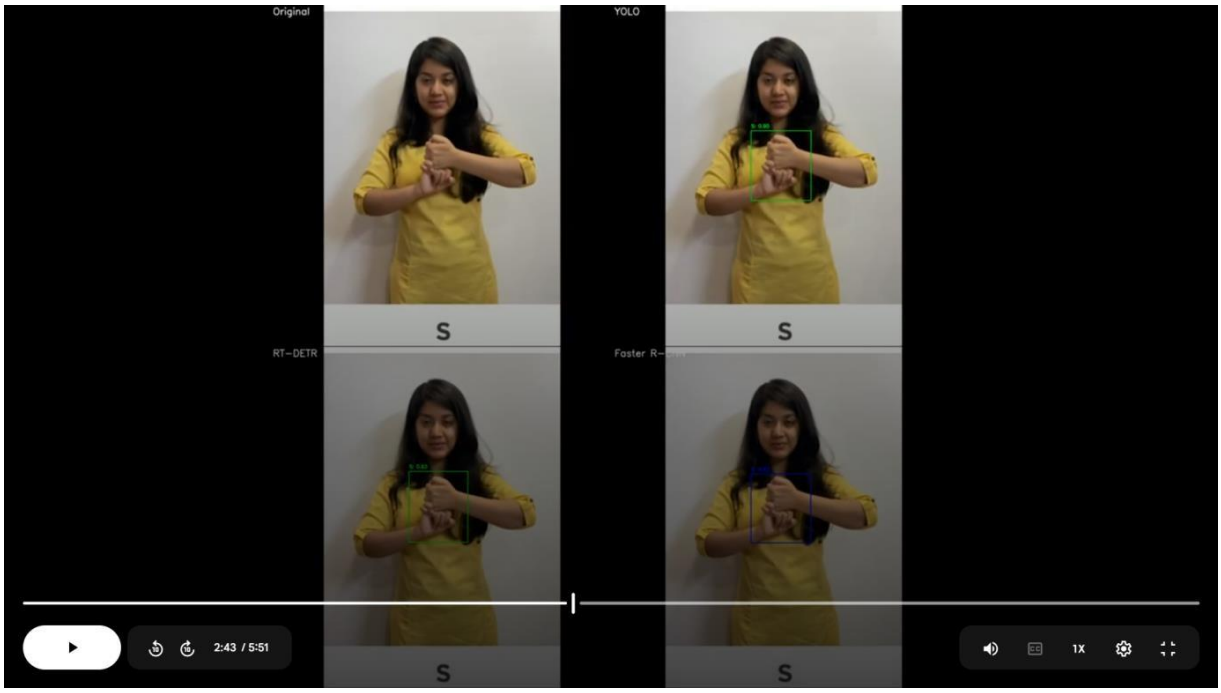

DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
Image 7
Image 8
Image 9


DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
9: 0.99
Image 5: 7_mp4-27_jpg.rf.61684fee59c9ca64475d7b57b3f15033.jpg
7: 0.99


DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
B: 0.93
Image 9: A_mp4-27_jpg.rf.942561ca6f0c9cb4f6842716211bf8df.jpg
A: 0.96

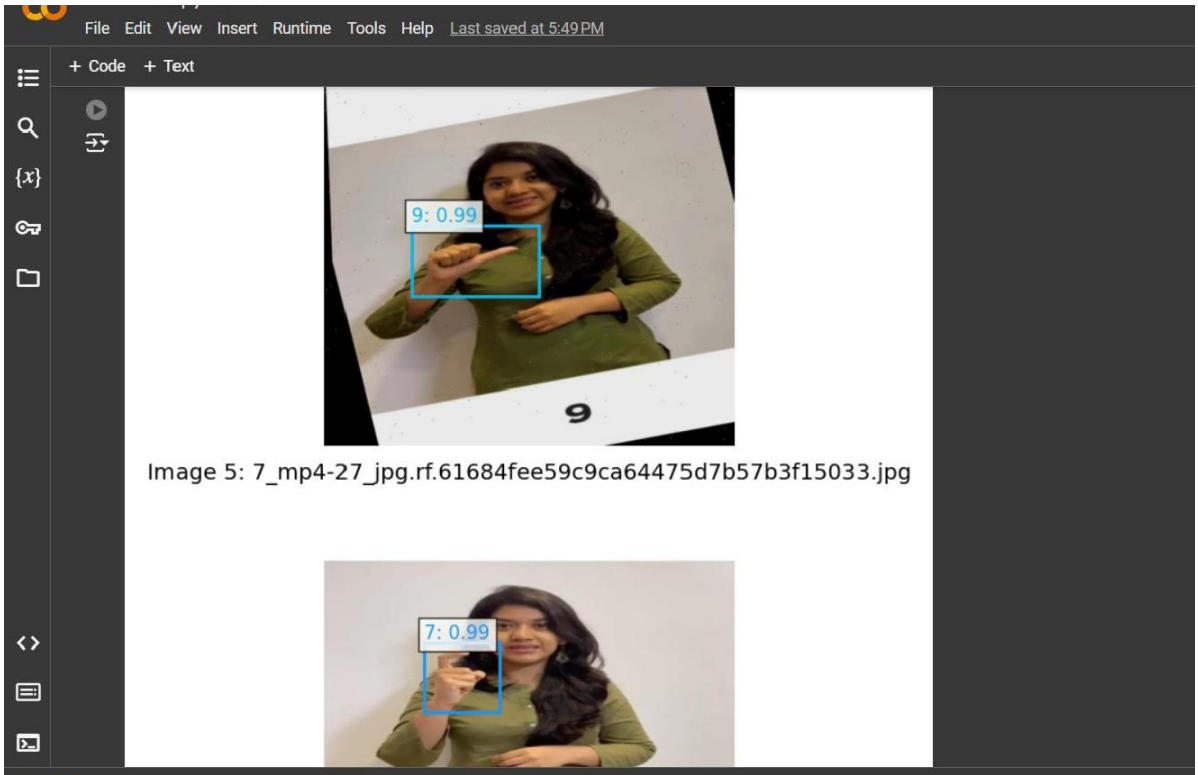

DEMO.ipynb
File Edit View Insert Runtime Tools Help Last saved at 4:20 PM
Share
+ Code + Text
Gemini
if __name__ == "__main__":
main()
Streaming output truncated to the last 5000 lines.
Speed: 2.8ms preprocess, 59.9ms inference, 0.7ms postprocess per image at shape (1, 3, 480, 480)
Processing frame 9895
0: 640x640 (no detections), 25.5ms
Speed: 5.7ms preprocess, 25.5ms inference, 1.5ms postprocess per image at shape (1, 3, 640, 640)
0: 480x480 (no detections), 71.0ms
Speed: 2.7ms preprocess, 71.0ms inference, 0.7ms postprocess per image at shape (1, 3, 480, 480)
Processing frame 9896
0: 640x640 (no detections), 18.2ms
Speed: 5.7ms preprocess, 18.2ms inference, 0.6ms postprocess per image at shape (1, 3, 640, 640)
0: 480x480 (no detections), 44.7ms
Speed: 2.9ms preprocess, 44.7ms inference, 0.5ms postprocess per image at shape (1, 3, 480, 480)
Processing frame 9897
0: 640x640 (no detections), 18.0ms
Speed: 5.9ms preprocess, 18.0ms inference, 0.6ms postprocess per image at shape (1, 3, 640, 640)
0: 480x480 (no detections), 45.6ms
Speed: 3.2ms preprocess, 45.6ms inference, 0.6ms postprocess per image at shape (1, 3, 480, 480)
Processing frame 9898

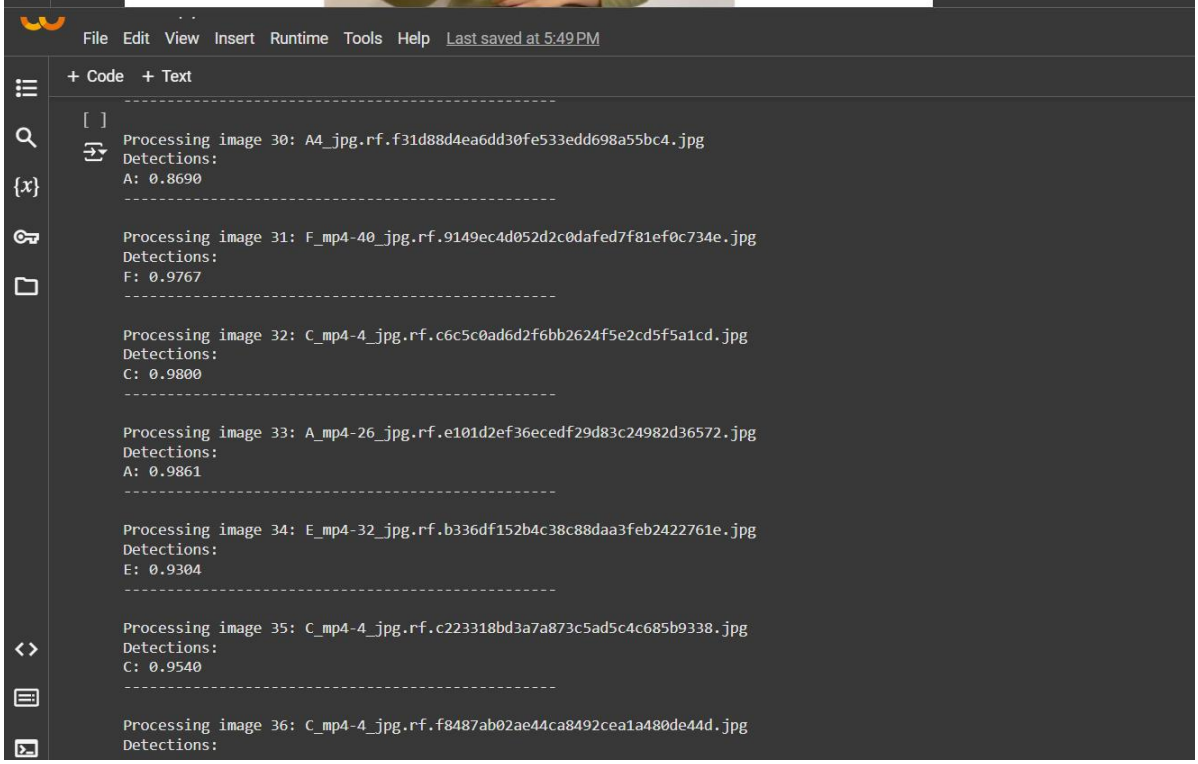

Drive
Search in Drive
New
My Drive > predictions
Type People Modified
Home
My Drive
Colab Notebooks
Dataset
ISL
Models
NCI_Sep_Thesis_2024
predictions
Computers
Shared with me
Recent
Starred
Name Owner Last modified File size
DevakiISL_yolo_20241211_092946.mp4 me 3:43 PM me 83.2 MB
DevakiISL_rtdetr_20241211_092946.mp4 me 3:43 PM me 87 MB
DevakiISL_frcnn_20241211_092946.mp4 me 3:43 PM me 112.4 MB
DevakiISL_combined_20241211_092946.mp4 me 2:59 PM me 357.7 MB

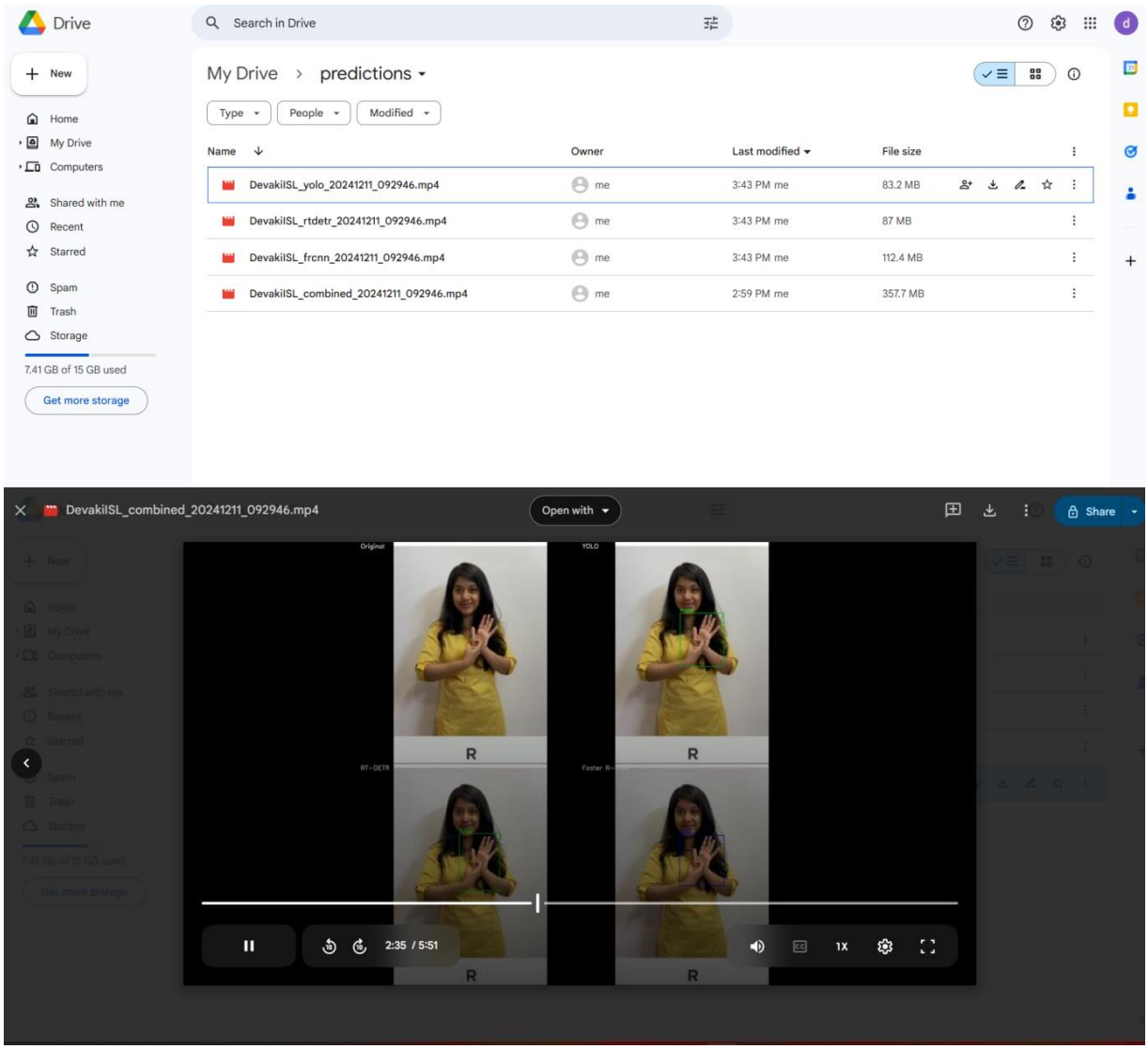



```
    detector = CombinedSignLanguageDetector()
    video_path = '/content/drive/MyDrive/NCI_Sep_Thesis_2024/DevakiISL.mp4'  # Replace with your video path

    try:
        detector.process_video(video_path, conf_threshold=0.25)
    except Exception as e:
        print(f"Error processing video: {e}")

if __name__ == "__main__":
    main()
```

```
Streaming output truncated to the last 5000 lines.
Speed: 2.8ms preprocess, 59.9ms inference, 0.7ms postprocess per image at shape (1, 3, 480, 480)

Processing frame 9895

0: 640x640 (no detections), 25.5ms
Speed: 5.7ms preprocess, 25.5ms inference, 1.5ms postprocess per image at shape (1, 3, 640, 640)

0: 480x480 (no detections), 71.0ms
Speed: 2.7ms preprocess, 71.0ms inference, 0.7ms postprocess per image at shape (1, 3, 480, 480)

Processing frame 9896

0: 640x640 (no detections), 18.2ms
Speed: 5.7ms preprocess, 18.2ms inference, 0.6ms postprocess per image at shape (1, 3, 640, 640)

0: 480x480 (no detections), 44.7ms
Speed: 2.9ms preprocess, 44.7ms inference, 0.5ms postprocess per image at shape (1, 3, 480, 480)

Processing frame 9897
```


File Edit View Insert Runtime Tools Help Last saved at 5:49 PM
+ Code + Text
9: 0.99
9
Image 5: 7_mp4-27_jpg.rf.61684fee59c9ca64475d7b57b3f15033.jpg
7: 0.99


File Edit View Insert Runtime Tools Help Last saved at 5:49 PM
+ Code + Text
[ ]
Processing image 30: A4_jpg.rf.f31d88d4ea6dd30fe533edd698a55bc4.jpg
Detections:
A: 0.8690
Processing image 31: F_mp4-40_jpg.rf.9149ec4d052d2c0dafed7f81ef0c734e.jpg
Detections:
F: 0.9767
Processing image 32: C_mp4-4_jpg.rf.c6c5c0ad6d2f6bb2624f5e2cd5f5a1cd.jpg
Detections:
C: 0.9800
Processing image 33: A_mp4-26_jpg.rf.e101d2ef36ecedf29d83c24982d36572.jpg
Detections:
A: 0.9861
Processing image 34: E_mp4-32_jpg.rf.b336df152b4c38c88daa3feb2422761e.jpg
Detections:
E: 0.9304
Processing image 35: C_mp4-4_jpg.rf.c223318bd3a7a873c5ad5c4c685b9338.jpg
Detections:
C: 0.9540
Processing image 36: C_mp4-4_jpg.rf.f8487ab02ae44ca8492cea1a480de44d.jpg
Detections:



References

Google Colab. (n.d.). https://colab.research.google.com/

RT-DETR: Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision. arXiv:2409.08475. Retrieved from arXiv

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Retrieved from paperswithcode.com

Google Colab for Training Custom Datasets with YOLOv8:Training Custom Datasets with Ultralytics YOLOv8 in Google Colab. Retrieved from ultralytics.com

Inference on Videos with YOLOv8:Roboflow. (n.d.). How to Run Video Inference with YOLOv8. Retrieved from roboflow.com

Inference on Videos with RT-DETR:Gradio. (n.d.). Streaming Object Detection from Video. Retrieved from gradio.app

Inference on Videos with Faster R-CNN:Stack Overflow. (2021). How to test custom Faster RCNN model(using Detectron 2 and PyTorch) on video. Retrieved from stackoverflow.com