Hostel Id : 84

Semi Supervised Weed Detection Challenge

Problem Statement

Our task is to develop a weed detection model using semi-supervised learning techniques.

We are provided with :

- **A small labeled dataset of agricultural field images containing sesame crops and weeds (200 images).**
- **A larger unlabeled dataset of similar images of sesame crops and weeds (1000 images).**

The objective is to train a model/s capable of accurately identifying and localizing weeds within these images, demonstrating a clear improvement in performance by effectively utilizing the unlabeled data.

Metric for evaluation: 0.5 * (F1-Score) + 0.5 * (mAP@[.5:.95])

Initial Approaches

1.) SimCLR Implementation

Initially, we leveraged SimCLR (Simple Framework for Contrastive Learning of Visual Representations) for its robustness in self-supervised learning. SimCLR's methodology involves:

- Random augmentation of unlabeled images to create two correlated views
- Passing these views through a shared encoder
- Utilizing a contrastive loss function to:
  - →Maximize similarity between views of the same image
  - → Minimize similarity with other images in the batch

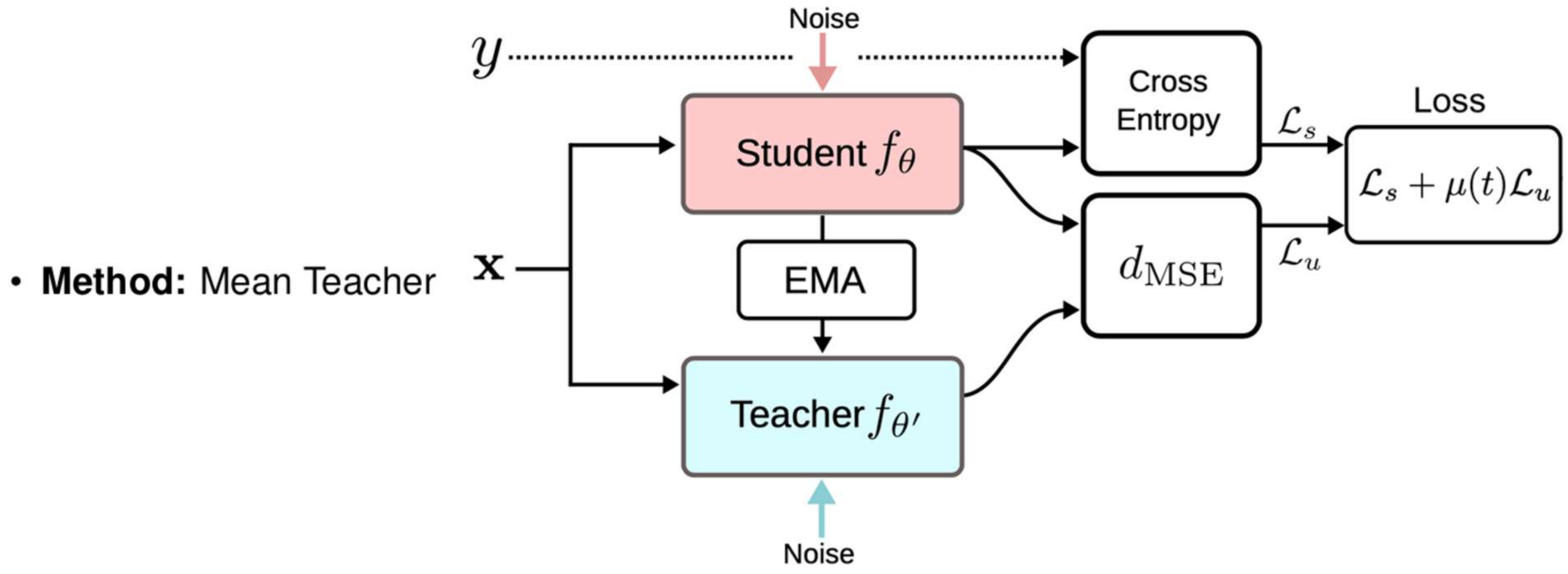
This process enables learning meaningful features from unlabeled data, which can then be fine-tuned with limited labeled data4. However, with only 200 labeled and 1000 unlabeled images, we lacked diverse negative samples.

2.) Mean Teacher with YOLOv8s

We then transitioned to a Mean Teacher approach using YOLOv8s, anticipating improved performance6. This method involves:

- Generating pseudo-labels using a teacher model
- Performing consistency checks between student and teacher predictions

Despite its potential, this approach significantly increased inference time due to the pseudo-label generation and consistency checking processes.



Proposed Solution

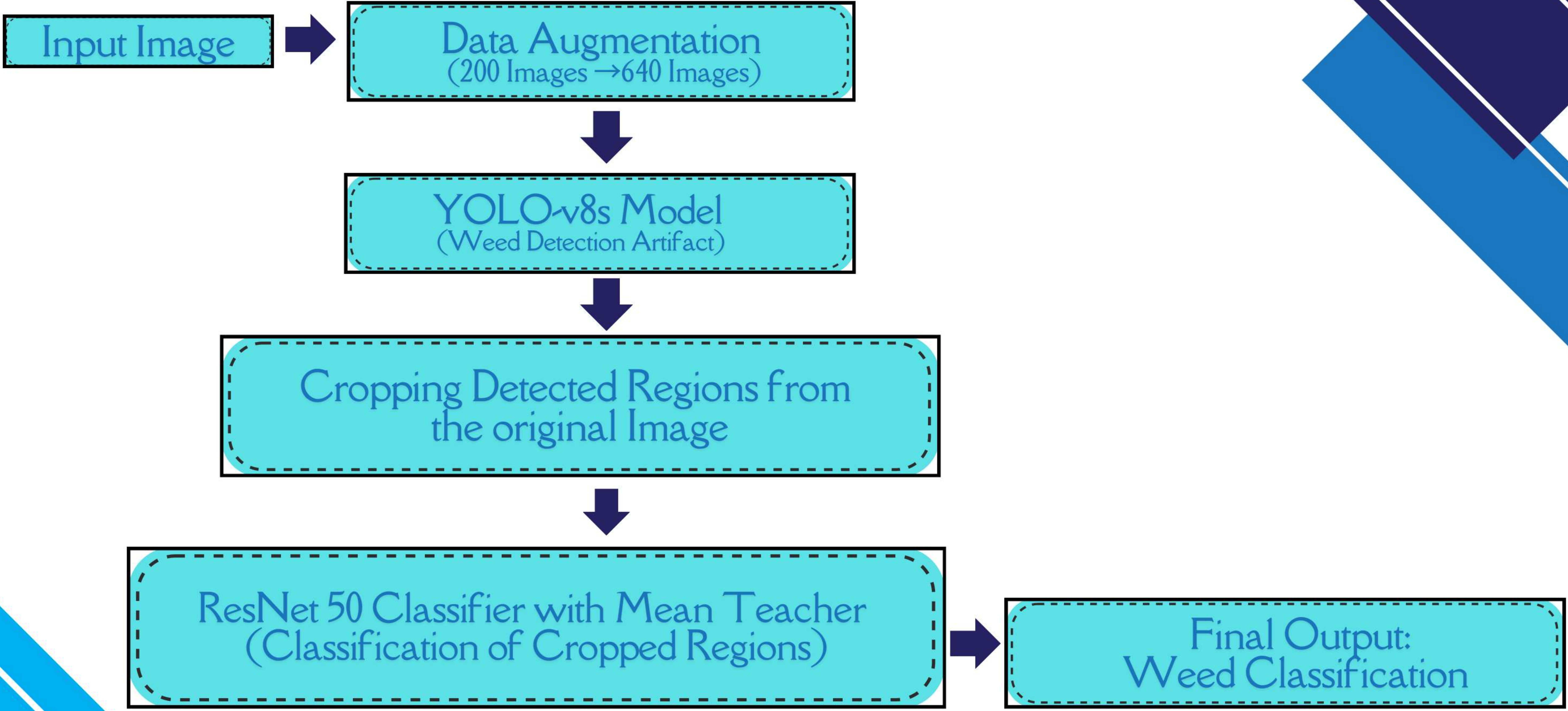
- Integrated Pipeline: Combines robust object detection with refined classification into a seamless system.

- **Detection with YOLO-v8s** :
  - Fine-tuned on an augmented dataset expanded from 200 to 640 images using diverse augmentation techniques.
  - Scans input images to accurately localize potential weed zones.
  - Region Cropping: Detected weed areas are cropped from the original image for further analysis using our classifier model.

- **Classification with ResNet-50** :
  - Utilizes a ResNet-50 classifier enhanced by the Mean Teacher Method.
  - Mean Teacher Approach:
    - Student Model: Learns from both labeled and unlabeled data using cross-entropy loss.
    - Teacher Model: Acts as an exponential moving average of the student to provide stable target predictions.

Outcome : The dual-stage system leverages augmented data and advanced learning strategies to deliver high accuracy and robust weed artifact classification.

Implemented Pipeline



Data Augmentation

```python
# Data augmentation
def adjust_annotations(annotations, crop_x, crop_y, crop_w, crop_h, orig_w, orig_h):
    """Adjust bounding box annotations after cropping."""
    new_annotations = []
    for line in annotations:
        class_id, x_center, y_center, width, height = map(float, line.strip().split())
        x_abs, y_abs = x_center * orig_w, y_center * orig_h
        width, height = width * orig_w, height * orig_h

        x1, y1 = x_abs - width / 2, y_abs - height / 2
        x2, y2 = x_abs + width / 2, y_abs + height / 2

        if x1 >= crop_x and y1 >= crop_y and x2 <= crop_x + crop_w and y2 <= crop_y + crop_h:
            x1_new, y1_new = x1 - crop_x, y1 - crop_y
            x2_new, y2_new = x2 - crop_x, y2 - crop_y
            x_center_new = (x1_new + x2_new) / 2 / crop_w
            y_center_new = (y1_new + y2_new) / 2 / crop_h
            width_new = (x2_new - x1_new) / crop_w
            height_new = (y2_new - y1_new) / crop_h
            new_annotations.append(f"{class_id} {x_center_new:.6f} {y_center_new:.6f} {width_new:.6f} {height_new:.6f}")
    return new_annotations

def random_crop(image, annotations, crop_size=(224, 224)):
    """Randomly crop image and adjust annotations."""
    h, w, _ = image.shape
    crop_h, crop_w = crop_size
    if crop_h > h or crop_w > w:
        return image, annotations
    x_start = random.randint(0, w - crop_w)
    y_start = random.randint(0, h - crop_h)
    cropped_image = image[y_start:y_start + crop_h, x_start:x_start + crop_w]
    new_annotations = adjust_annotations(annotations, x_start, y_start, crop_w, crop_h, w, h)
    return cropped_image, new_annotations

def invert_colors(image):
    """Invert image colors."""
    return cv2.bitwise_not(image)

def apply_gaussian_blur(image, kernel_size=(5, 5)):
    """Apply Gaussian blur to image."""
    return cv2.GaussianBlur(image, kernel_size, 0)
```

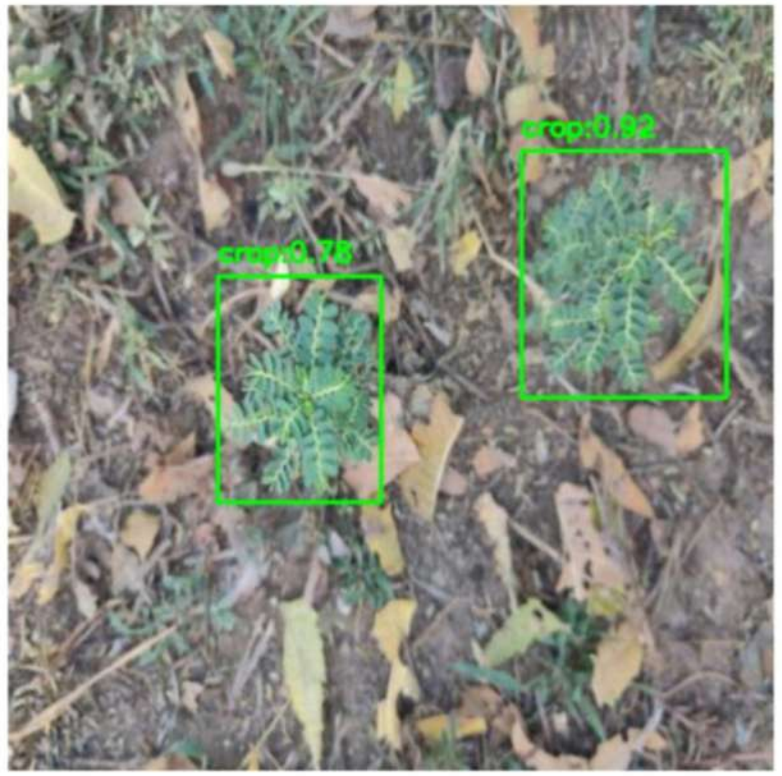
Image 1

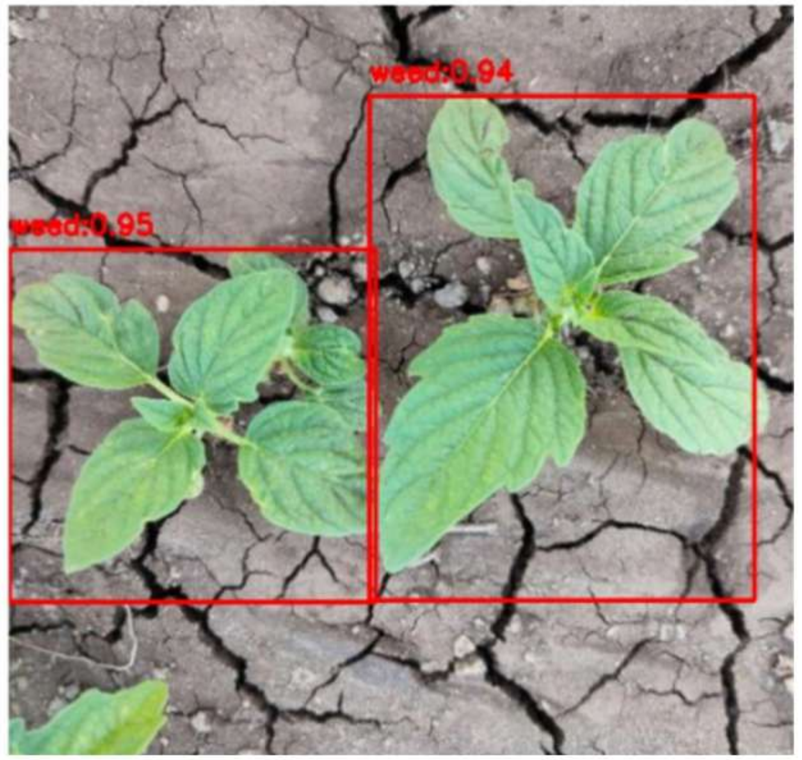
Image 2

Image 3

Sample Results of our Pipeline

Green Box = Crop
Red Box = Weed





Challenges

- **Limited Labeled Data**: As per the problem statement, our main task was to train a model with a dataset that was not only limited but had only a minimal portion labeled. To overcome this, we leveraged the seamless potential of semi-supervised training.

- **Computational Complexity**: Mean Teacher & YOLO-v8 integration increased training/inference time. The YOLO model even when we were freezing it and fine-tuning it took almost 20 min for one epoch. And under MTM architecture we have to train at least 100 such epochs. So we have to move to ResNet-50 model.

- **Accuracy Challenge**: Without Mean Teacher, baseline model ResNet-50 achieved 74% accuracy. Using Mean Teacher method, augmentation, and 100 epochs, an F1-Score of 1 was obtained.

- **Pipeline Integration**: After overcoming all the previous challenges, we successfully created a pipeline with seamless data flow, ensuring no loss of information.

Conclusions

- **Pipeline Architecture** :

  → Two-stage pipeline optimizes both labeled and unlabeled data
  → Stage 1 : Smart integration of unlabeled data for classification
  → Stage 2 : Advanced model refinement techniques

- **Technical Implementation** :

  → Mean Teacher Method with ResNet-50
  → Teacher model guides student model
  → Improved accuracy and reduced loss
  → Improved pretrained yolo-v8s using Resnet50 classifier results

- **Key Benefits** :

  → Maximizes available data utility
  → Reduces overfitting
  → Flexible framework for various domains

Thank you