Cloudphysician

VITAL EXTRACTION CHALLENGE

Inter IIT Tech Meet 11.0

PRESENTED BY
Team 55

Cloudphysician

Model Overview

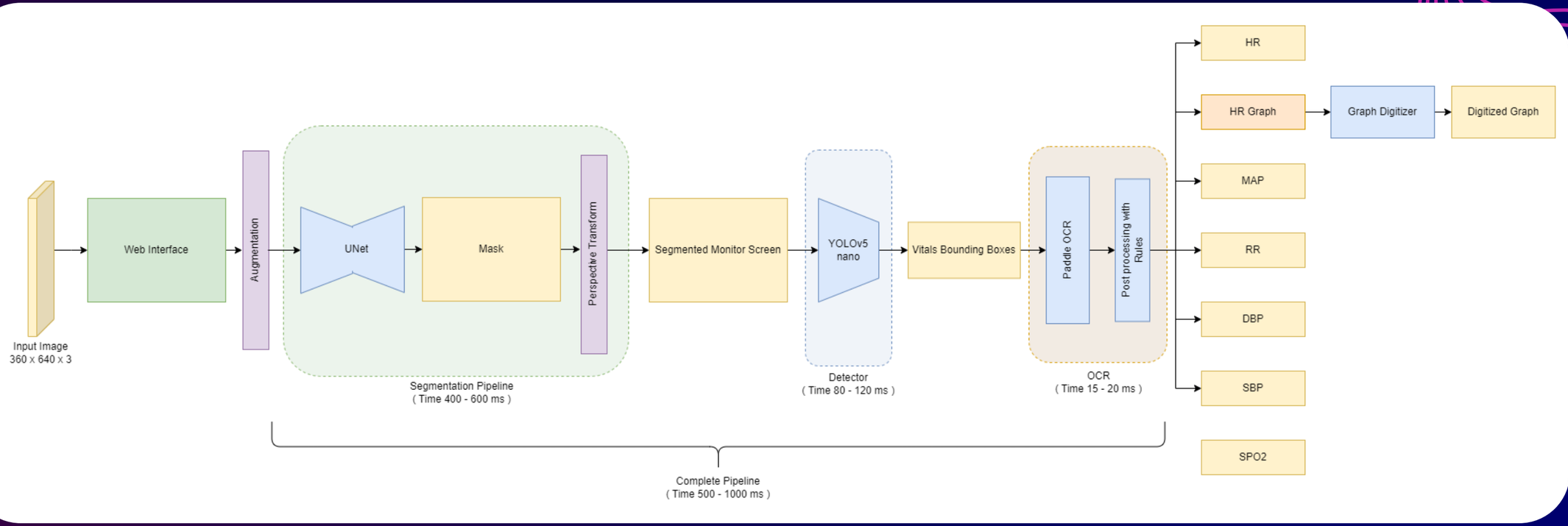
Our proposed model has 3 main steps:

- **Step 1:** Segmentation
- **Step 2:** Detection
- **Step 3:** Character Recognition.

Inference time (on CPU): 0.6 - 1.0 s

Cloudphysician

Proposed Model Pipeline



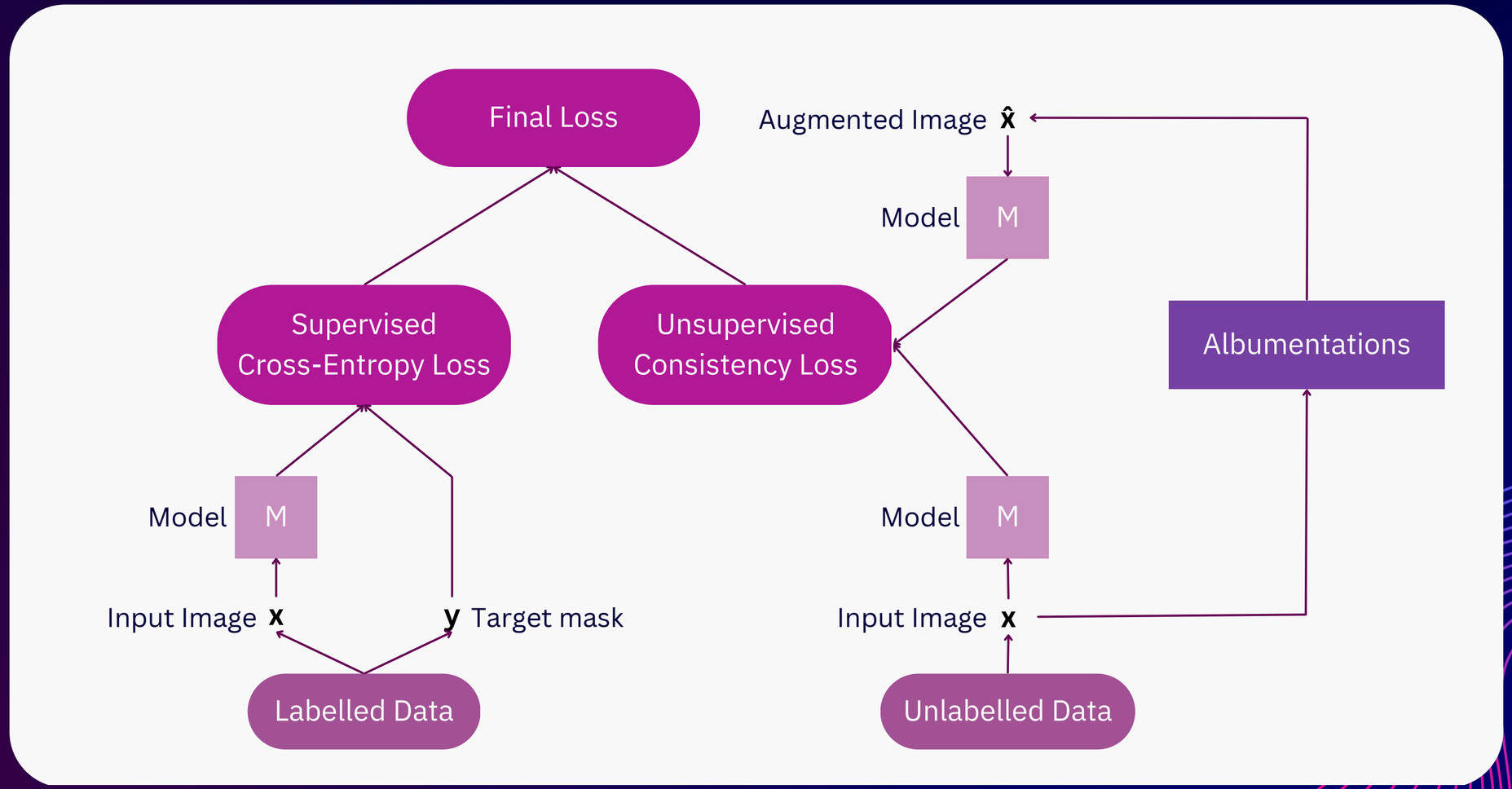
Cloudphysician

Step 1: Segmentation

- Segmentation based **U-Net architecture** (initialized with **IMAGENET weights**)
- **Semi-Supervised(SS) Semantic Segmentation Approach.**
- Novel additions: **LR scheduler**, **Augmentations variations** (randAugment - a type of novel augmentation technique)

Cloudphysician

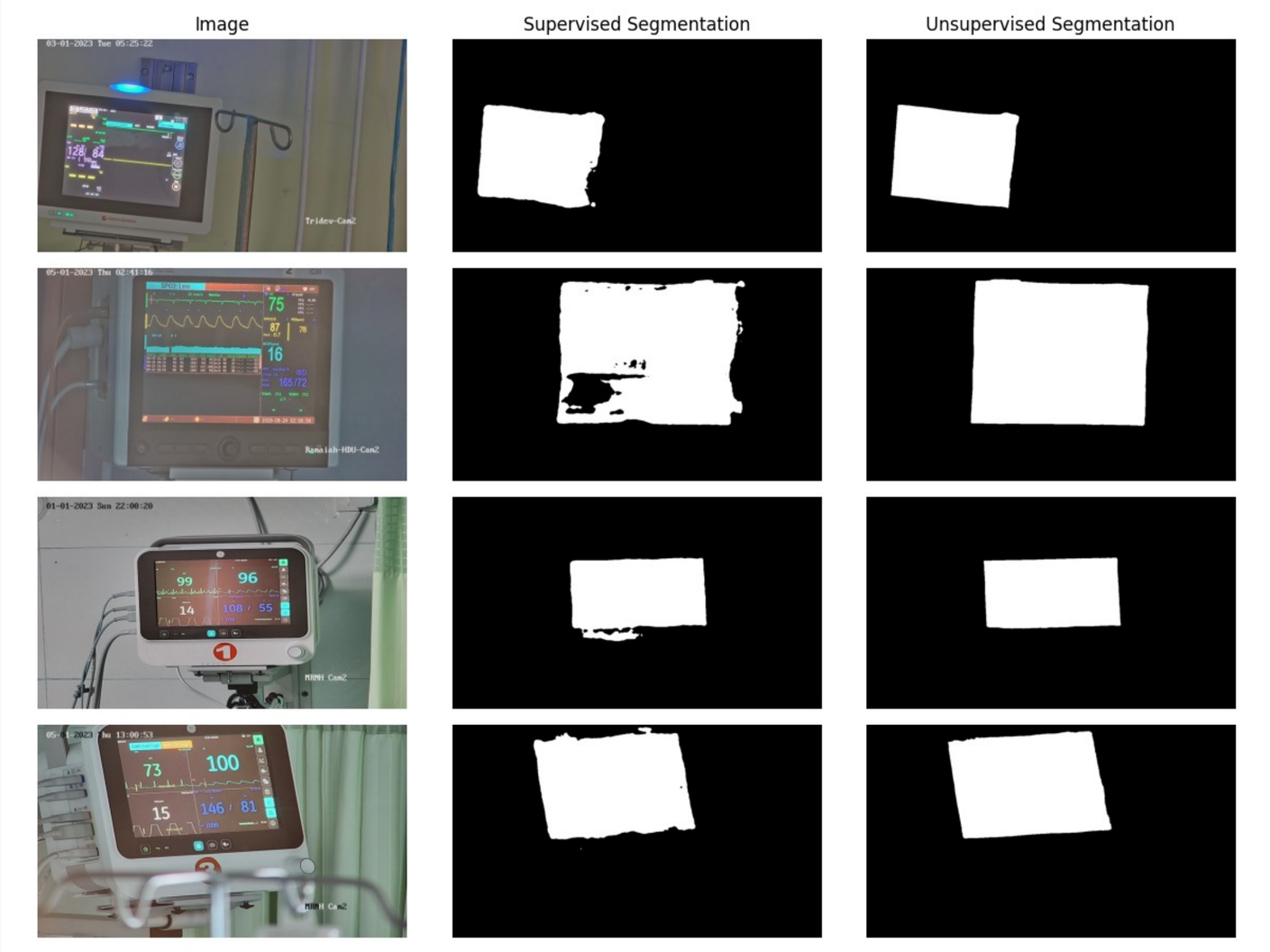
Semi-Supervised Segmentation



Cloudphysician

Why choose Semi-Supervised Segmentation model?

- monitor positions are not always rectangular
- detection model will not give good accuracy
- pixel level segmentation will perform better.



Cloudphysician

Segmentation Pipeline

Input Image

Augmentation

U-Net

Segmentation Mask

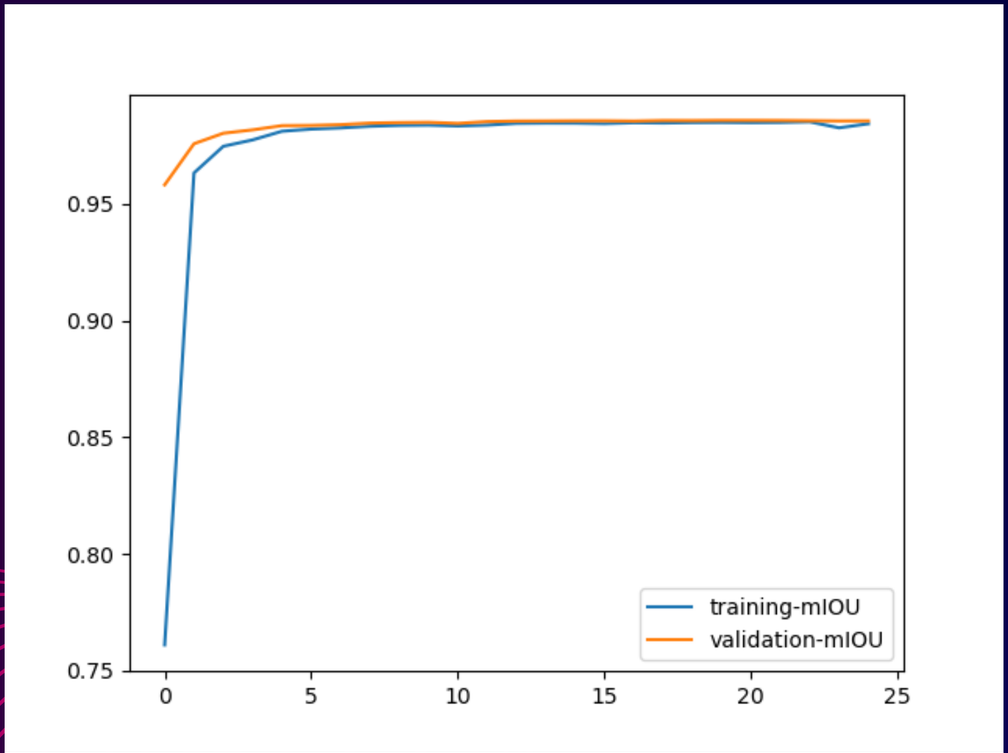
Perspective Transformation

Segmented and Perspective Transformed Image

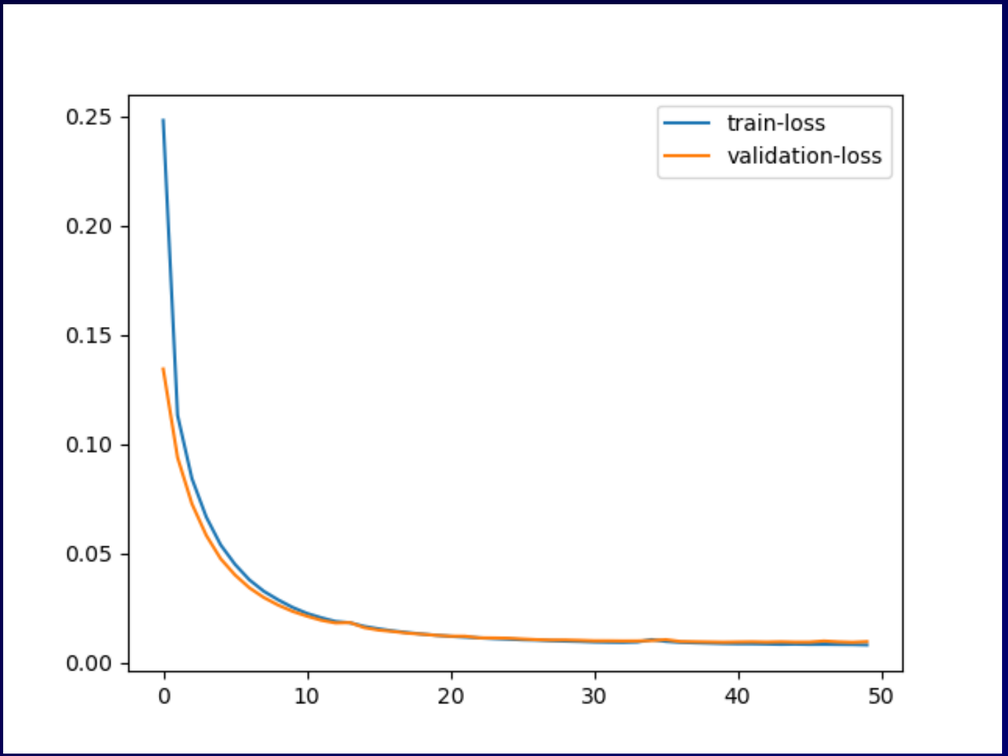
Inference time (on CPU): 0.4 - 0.6 s

Cloudphysician

Metrics



Plot for mIOU socre

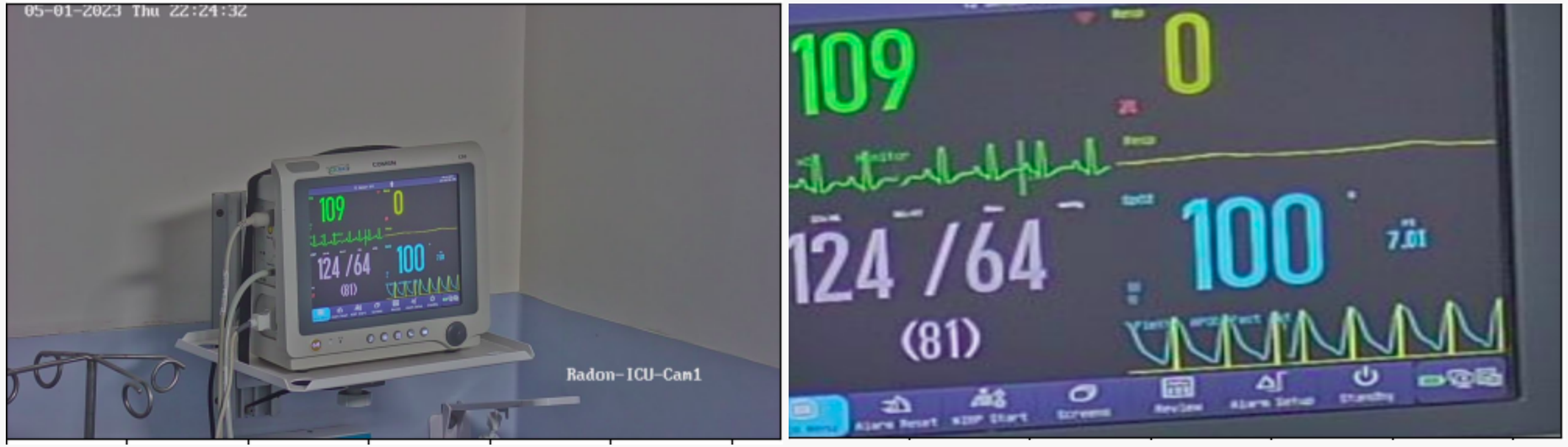


Plot for loss

Cloudphysician

Perspective Transformation

- **Convex hull** and **Contours methods** for finding mask corners.
- **OpenCV perspective transform** for wrapped perception.



Left - Original Image, **Right** - Segmented and Perspective transformed image

Cloudphysician

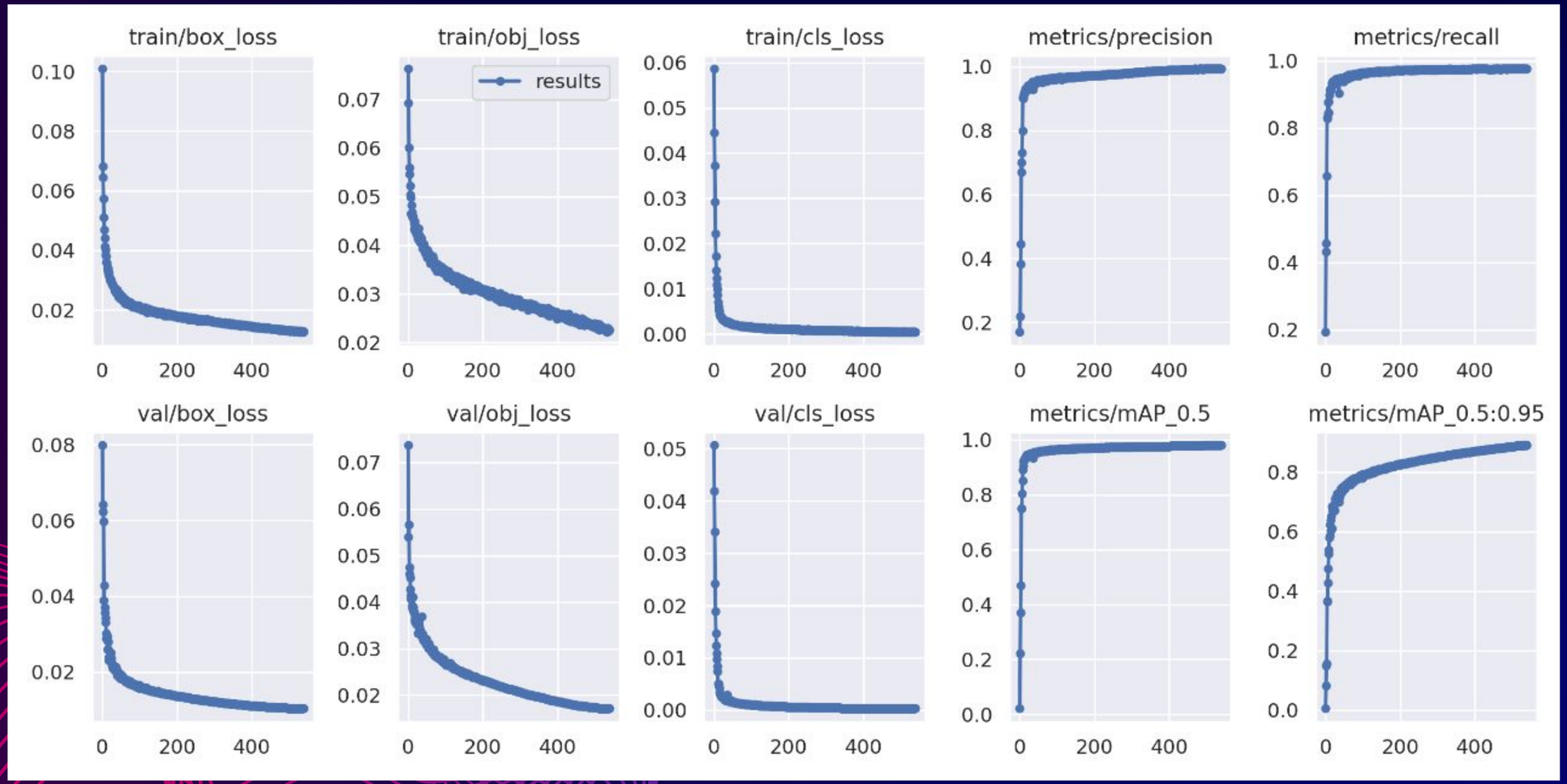
Step 2: Detection

- **YOLOv5nano** model - fast and lightweight
- Preprocessing and normalization done by the model itself
- **Inference time** (on CPU): 0.05 - 0.1 s

Top - Perspective Transformed Image
Bottom - Yolo Extracted Vitals and Graph

Cloudphysician

Metrics



Cloudphysician

Step 3: Character Recognition

- **PaddleOCR**, an open source OCR model
- Performed **OCR recognition** for vitals from the bounding boxes extracted from YOLO.
- Done **post-processing** on the predicted results based on monitor layout info.
- **Inference time** (on CPU): 0.15 - 0.20 s

Cloudphysician

PaddleOCR Framework

ODM OEM

ODM OEM

Image → Text Detection (db_mv3_slim, 1.4M) → Detection Boxes Rectify (dir_cls_mv3_slim, 0.5M) → Text Recognition (crnn_mv3_slim, 1.6M) → Output

1. Light Backbone
2. Light Head
3. Remove SE
4. Cosine Learning Rate Decay
5. Learning Rate Warm-up
6. FPGM Pruner

1. Light Backbone
2. Data Augmentation
3. Input Resolution
4. PACT Quantization

1. Light Backbone
2. Data Augmentation
3. Cosine Learning Rate Decay
4. Feature Map Resolution
5. Regularization Parameters
6. Learning Rate Warm-up
7. Light Head
8. Pre-trained Model
9. PACT Quantization

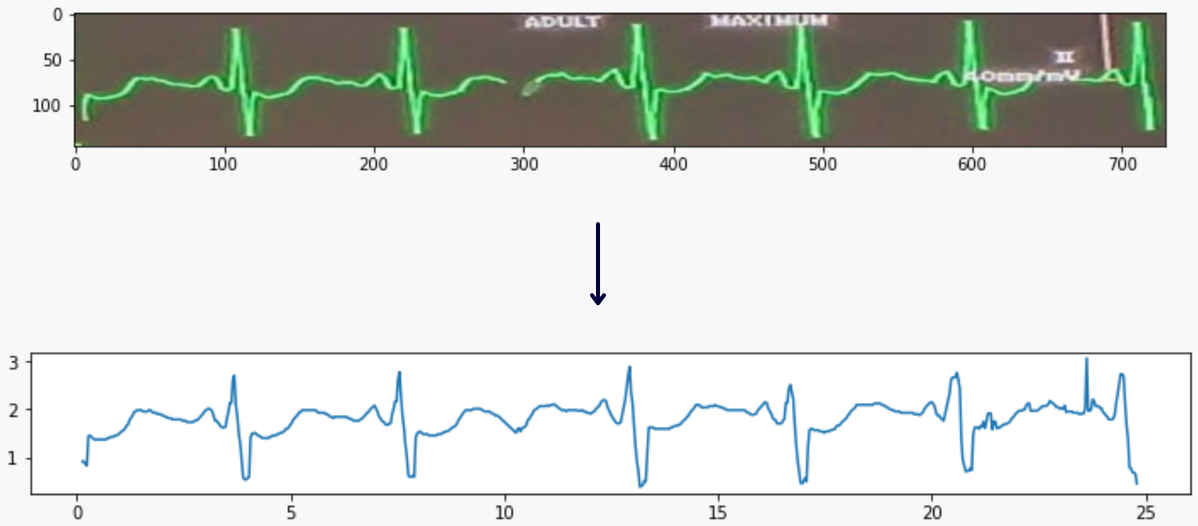
Source: Arxiv

Cloudphysician

Graph Digitizing

- Python implementation of **WebPlotDigitizer**.

- WebPlotDigitizer - web based tool to extract numerical data from plot images or maps.

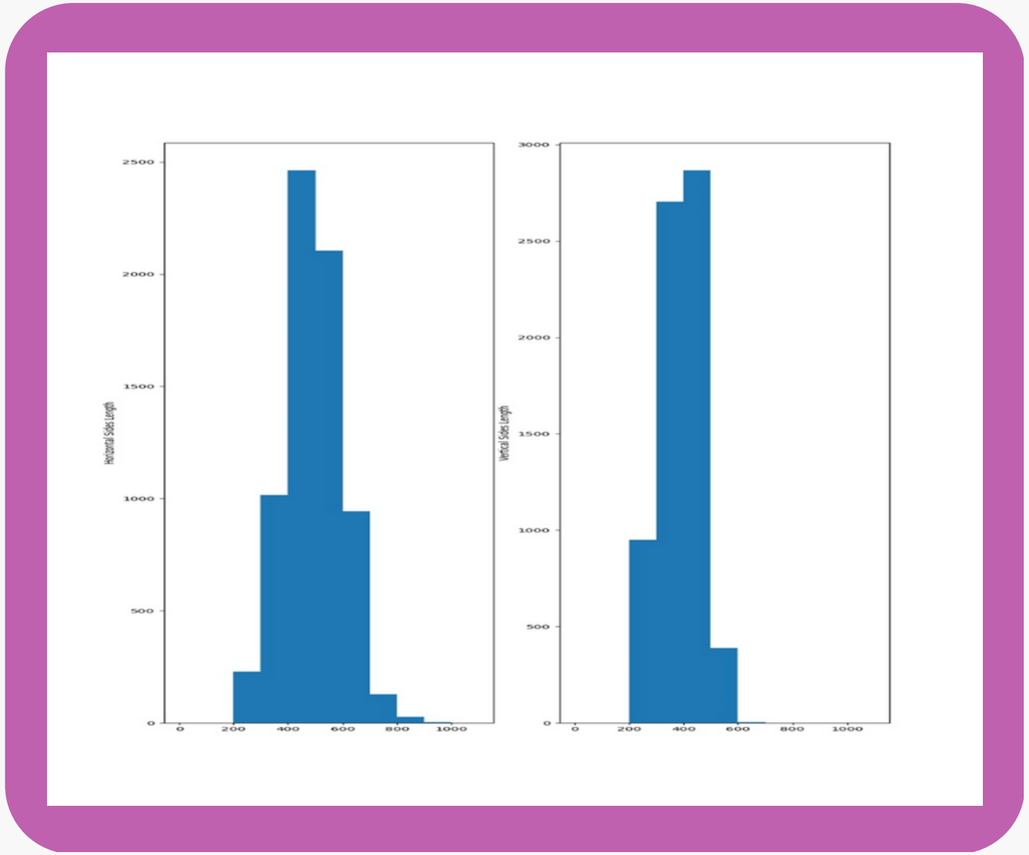


Cloudphysician

Novelty

In terms of Speed

- **Multiprocessing** during OCR recognition boosted speed by **2.5X.**
- **YOLOv5nano** for direct extraction of vitals very fast on CPU
- **Carefully chosen input size** - plotted histogram of both dimensions of extracted masks and took the mean of histogram as the optimal dimension for YOLO.



Cloudphysician

Novelty

In terms of Accuracy

- **Post-processing** on the predicted results based on monitor layout info.
- **Segmentation instead of detection**
- **Semi-supervised** based segmentation approach to leverage unlabelled data.

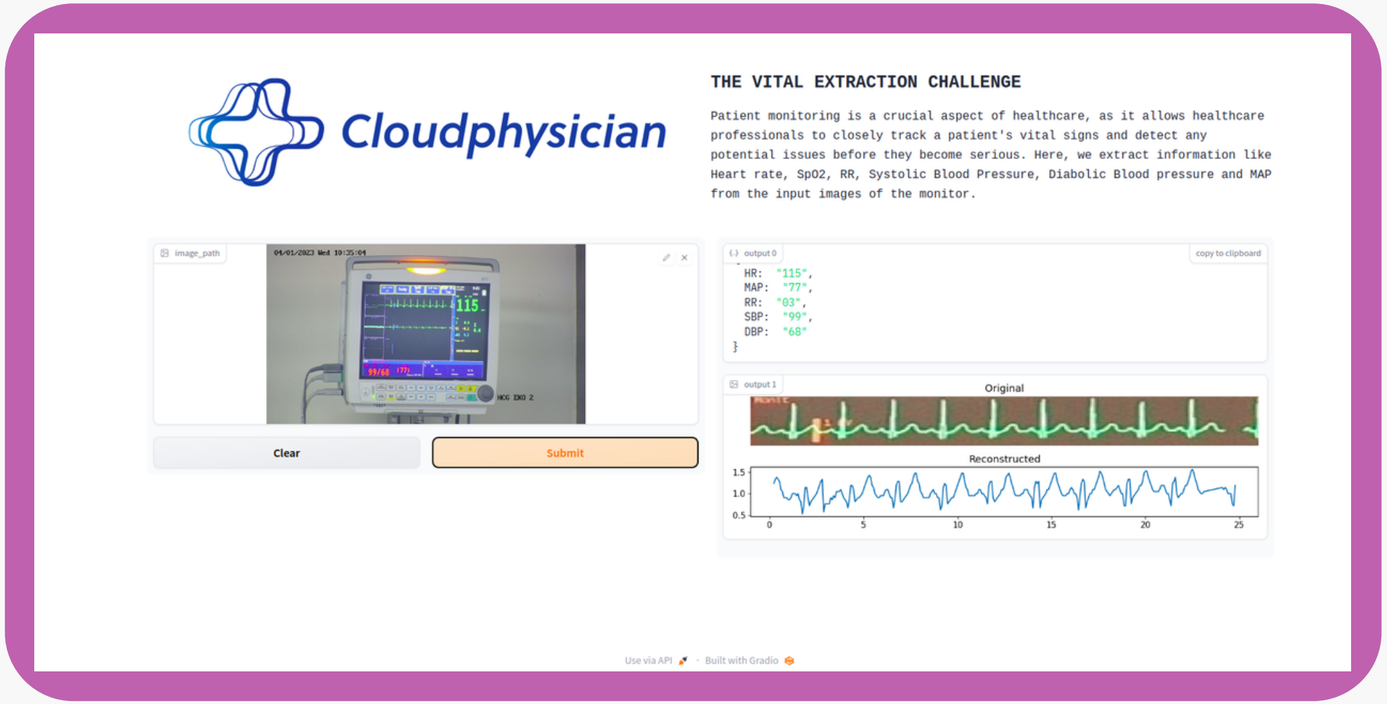
Cloudphysician

Novelty

In terms of Visualization

- **Gradio visualisation** to show results - a library which runs in Colab itself.

- Provides a localhost to drop images and get results.



Thank You