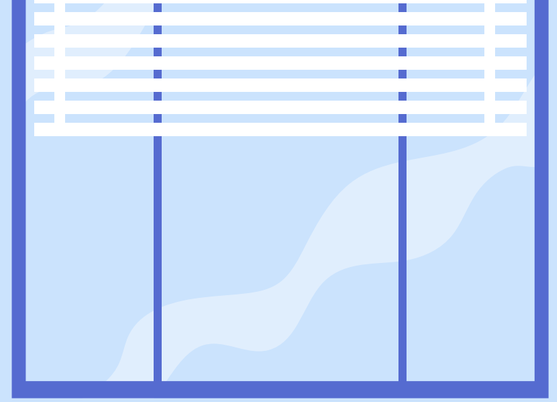
The Vital Extraction Challenge

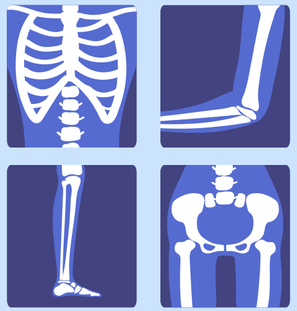
Team No: 41

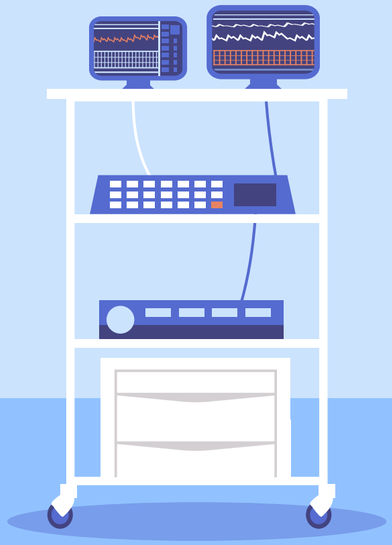
Problem Statement

01 Extract the values of vitals like Heart Rate, SpO2, RR, Systolic Blood Pressure, Diabolic Blood Pressure, and MAP from ICU monitor images

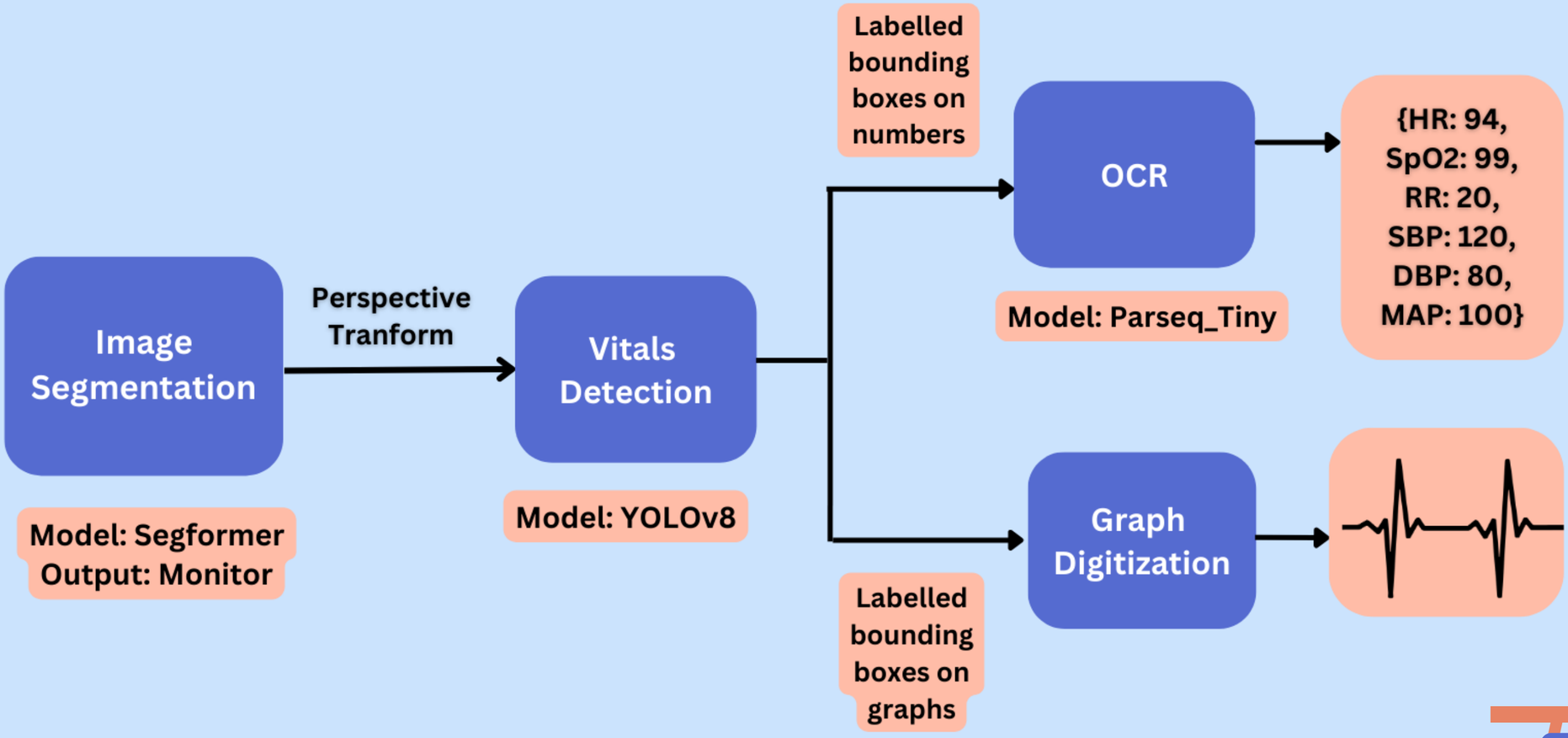
02 Extract the waveforms of Heart rate, SPO2, RR and digitize them





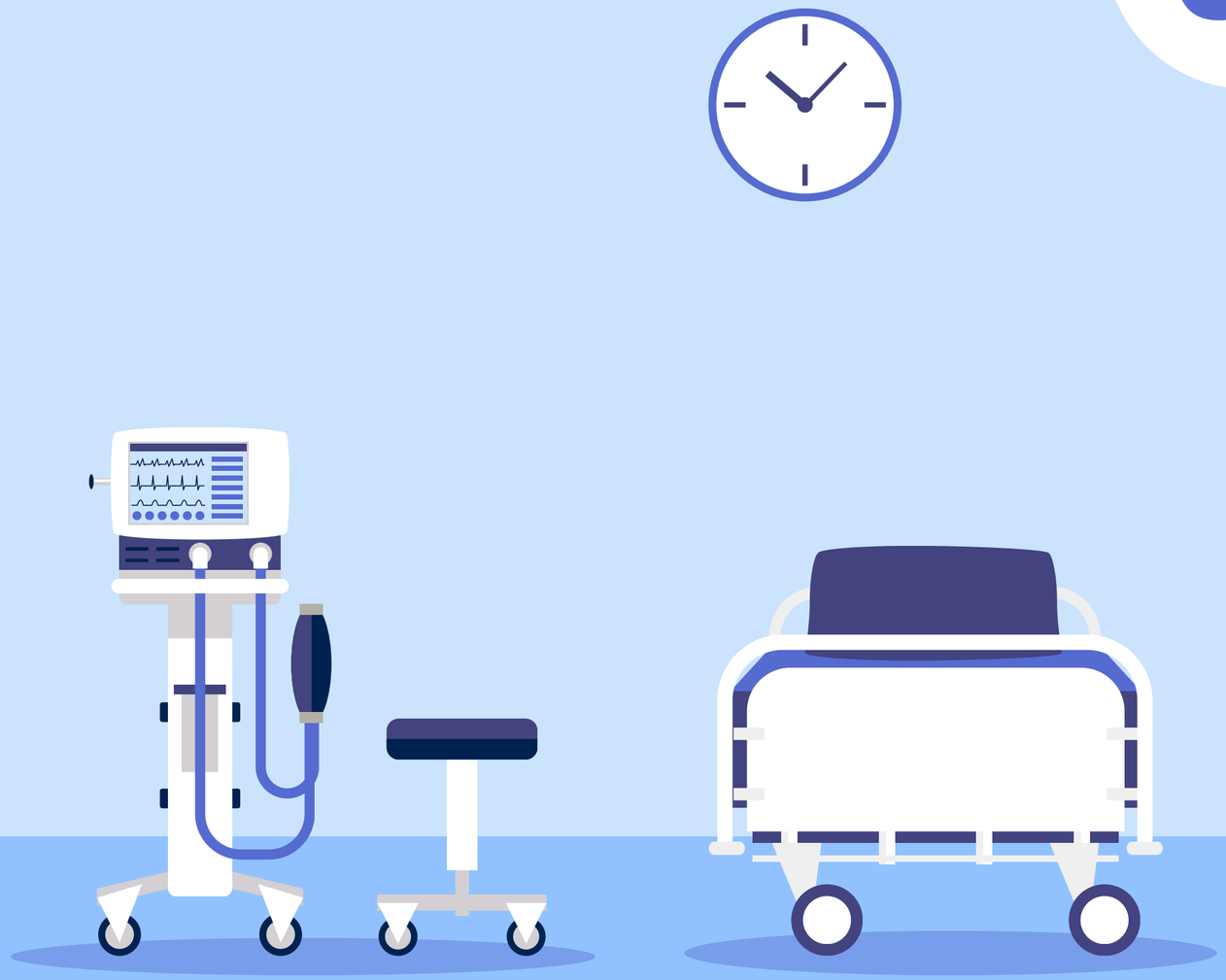


Overview of Pipeline



01

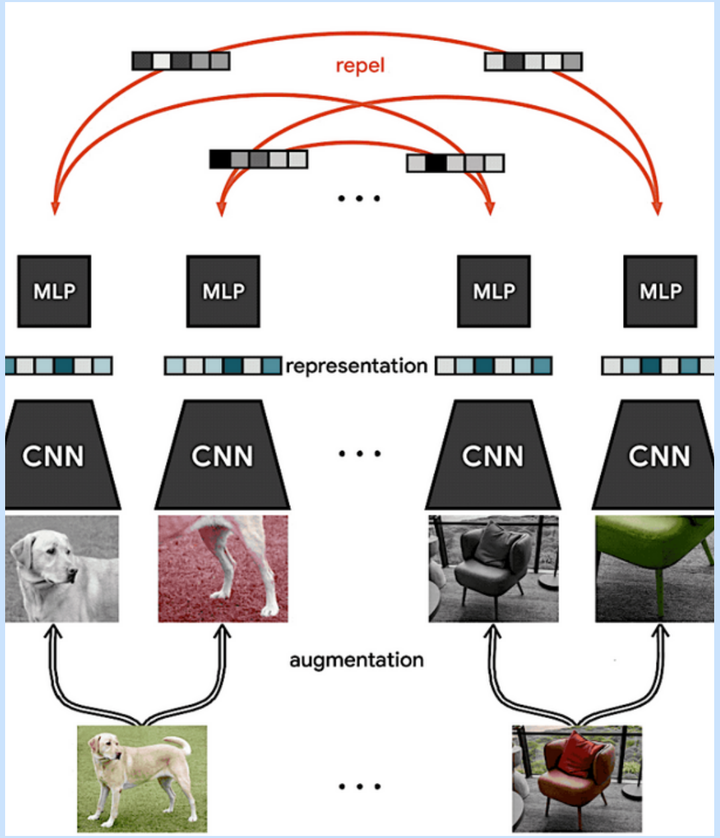
Monitor Screen Extraction



Step 1: Monitor Screen Extraction

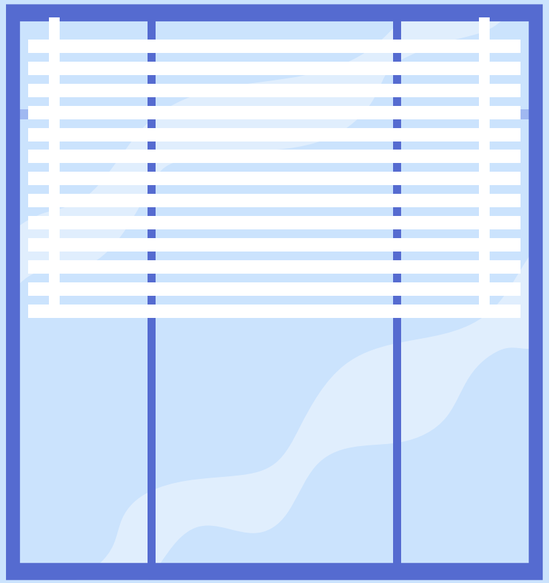
ResNet with modified output layer

- ResNet backbone + MLP head with 8 output neurons predicting the 4 corners
- We freeze the backbone with weights from unsupervised training of SimCLR and finetune the MLP head
- Good performance on training data, unsatisfactory on unlabelled data



Unsupervised SimCLR training on (unlabelled + labelled) and fine-tuning on labelled

Regression vs Classification



We wish to maximise the mutual information between the ground truth images and labels.

$$I(Y,X) = H(Y) - H(Y|X)$$

To maximise $I(Y,X)$ -

- Maximise $H(Y)$, i.e. entropy of the predicted labels
- Minimise $H(Y|X)$, i.e. entropy of predicted labels conditioned on the input image

- The regression loss minimise $H(Y|X)$
- CE loss simultaneously minimises $H(Y|X)$ while maximising $H(Y)$

Theoretically, turning this regression to a classification problem with cross-entropy loss should lead to more robust predictions

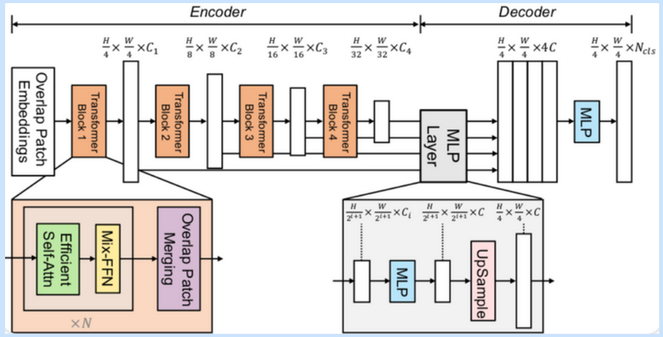
Probabilistic Heatmaps

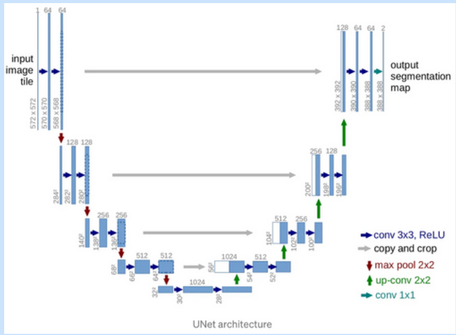
Model:

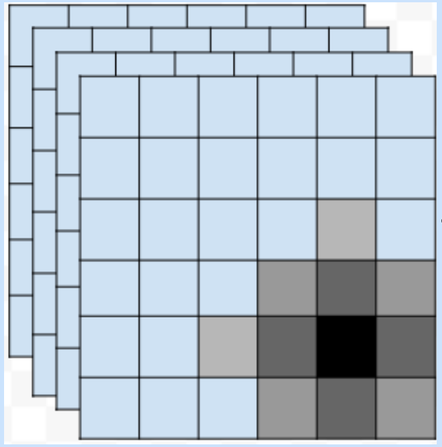
Input: 3 channel image
Output: 4 channel heat map (one for each corner)

UNet has heavy encoder and decoder -> high inference time

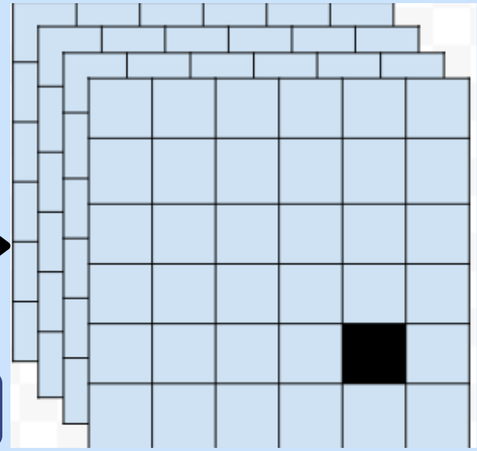
Segformer has a light decoder, better accuracy and faster inference







Cross-Entropy Loss

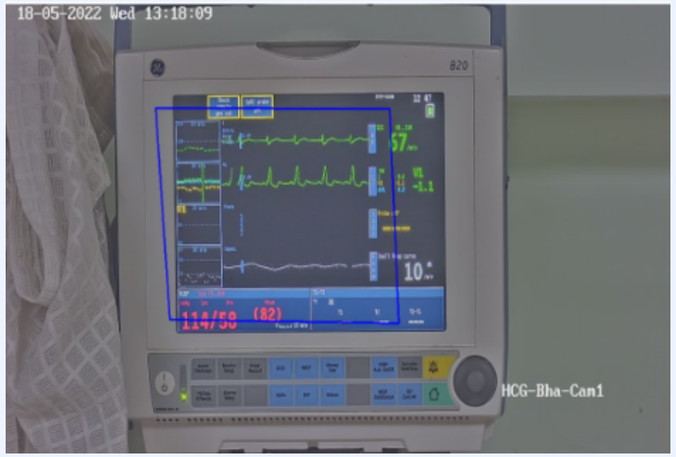


Predicted Probability Maps

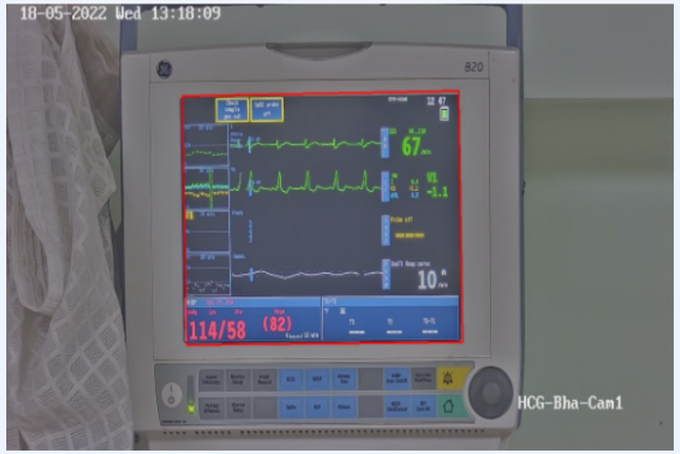
Ground truth

Performance Comparison for Segmentation

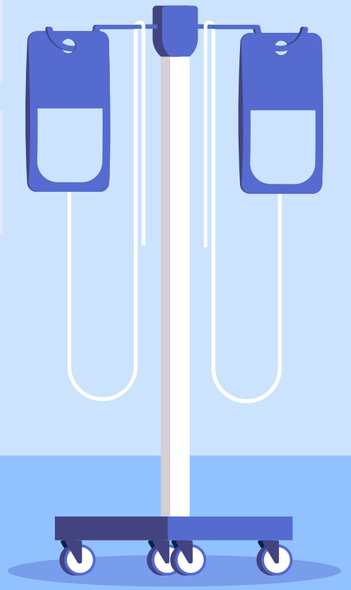
Resnet result



UNet/SF result



Model	Time	Validation Score
Segformer	0.7 seconds	4.235 pixels
UNet	3 seconds	6.516 pixels



02

Monitor Classification & Vital Detection

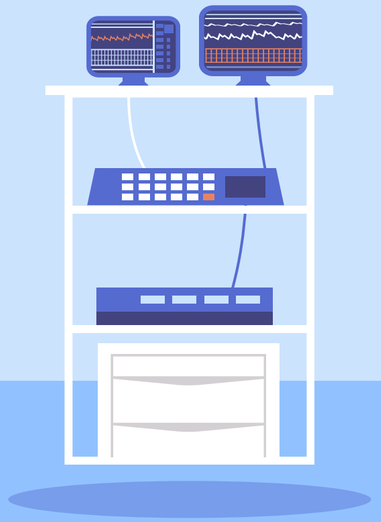
Step 2: Monitor Classification and Vital Detection

Initial approach:
Classification model on ResNet 50 while YOLO handles each of the 4 layouts separately

Modified approach:
Vital Detection only using Yolov8 (Bypassing the Classification)

Learnings:

1. Initial approach -> makes pipeline unnecessarily large, having to load 4 different models
2. YOLO -> capable of handling classes present in the dataset without an explicit classification ResNet model



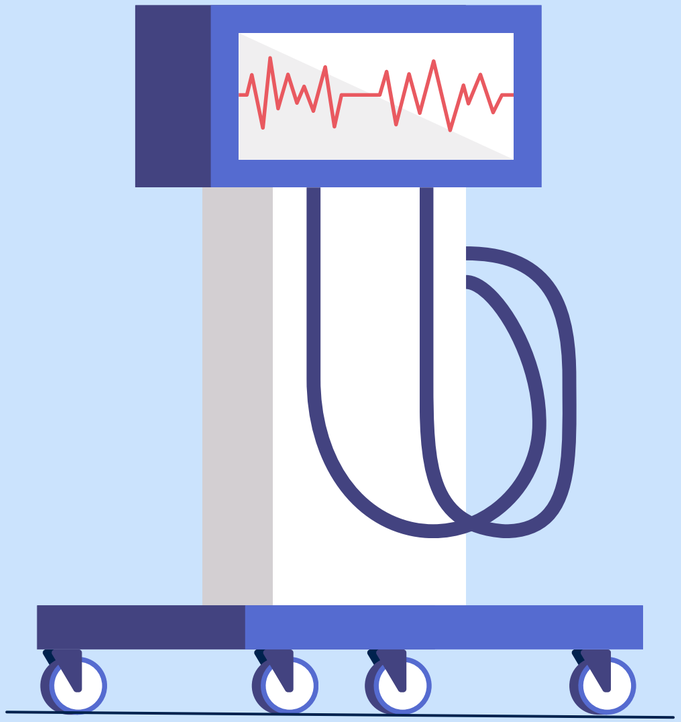
While testing Yolov8...

Drawbacks of the classification dataset for our approach:

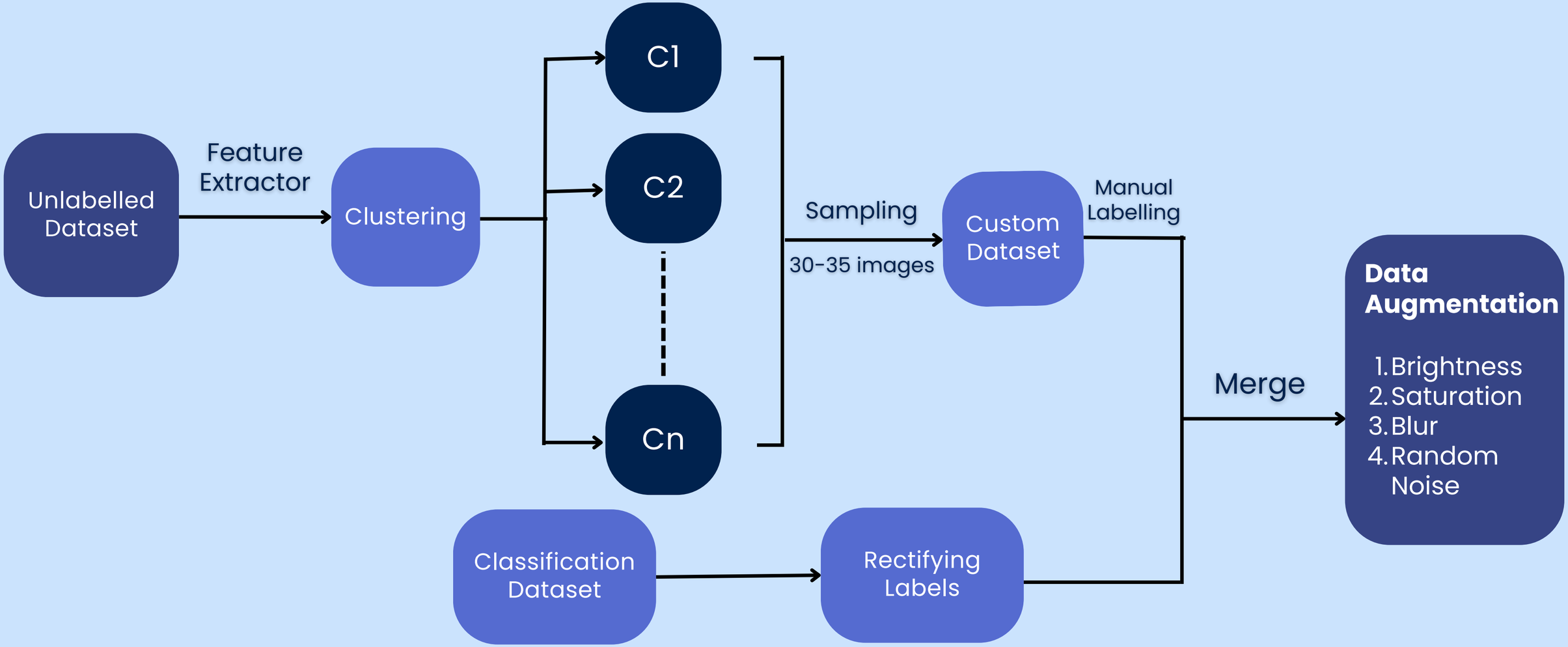
01 Limited orientation of vitals on monitor layouts in the dataset, not representing the complete distribution in the unlabelled dataset

02 Incomplete or no labeling of certain vitals (artifacts or noise) in the dataset.

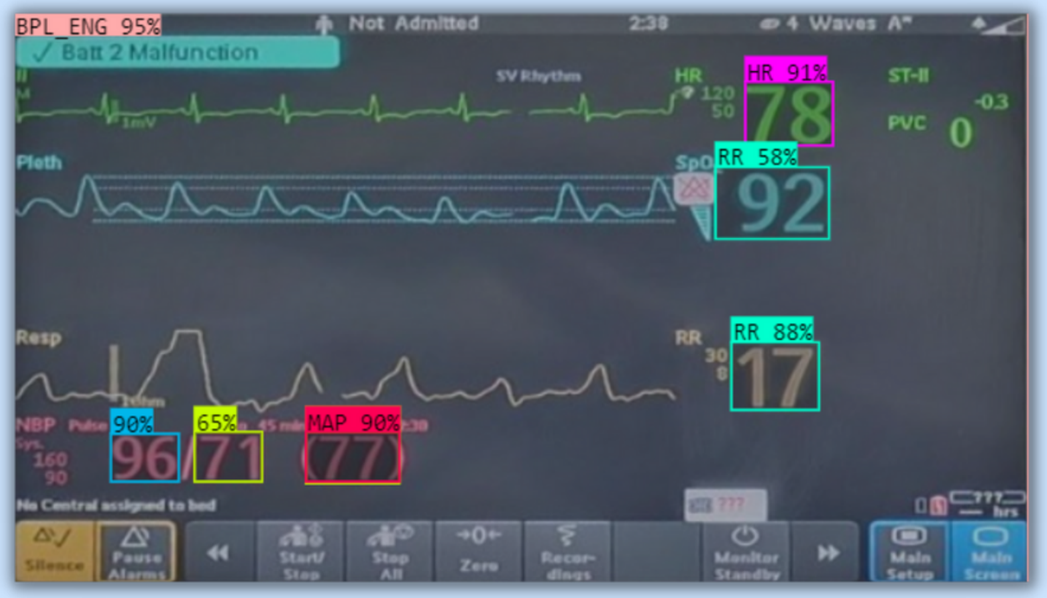
Result:
Poor performance on unlabelled dataset



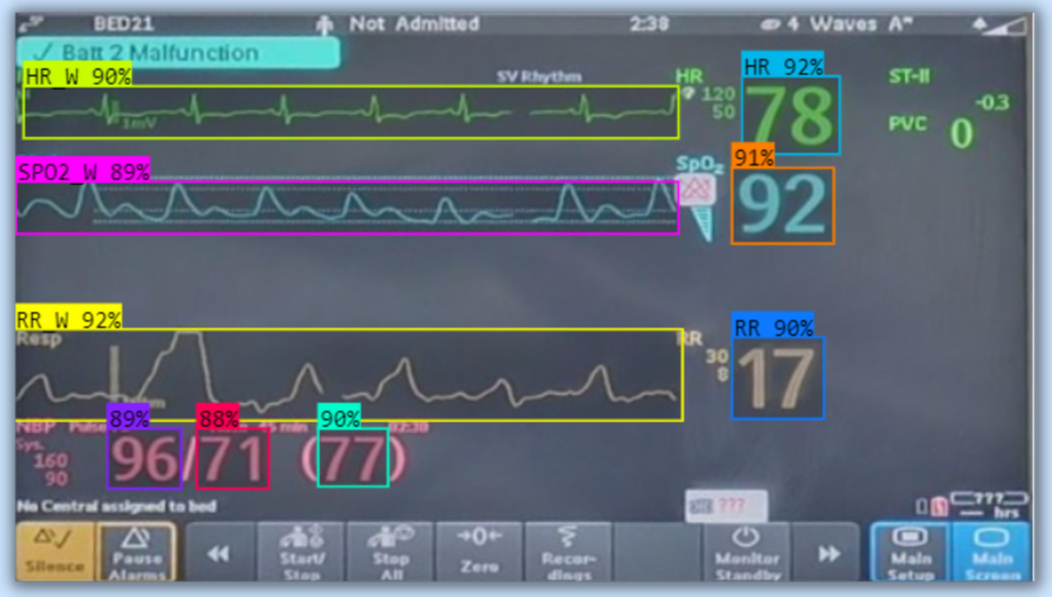
Dataset Generation



Improvements with custom dataset



Trained on classification and test on unlabelled

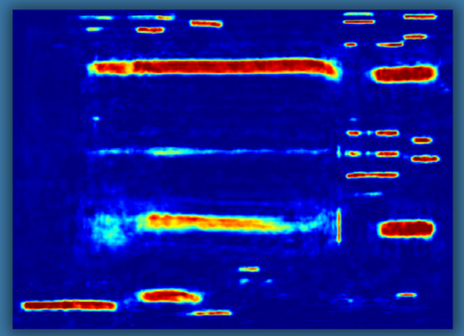


Trained on our new dataset and test on unlabelled

We use the YoloV8 trained on our custom dataset as the final model for the vital detection step

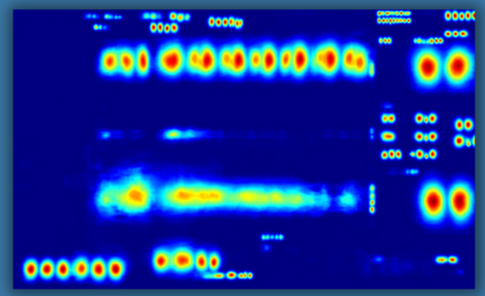
CRAFT for ROI Heatmaps

Input (Monitor Screen)

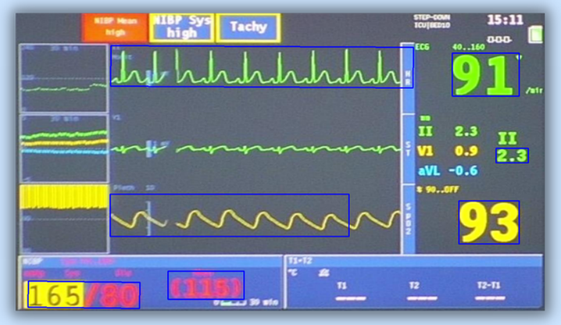


Link Score Heatmap

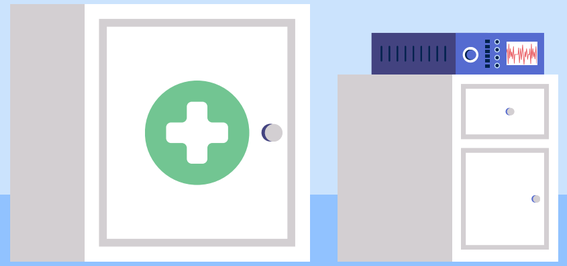
CRAFT



Text score Heatmap



Predicted Bounding Boxes



Improving YoloV8 using CRAFT

- **Synthesized image = Input image*Text_score_heatmap**
- **The training involved 6 classes of HR, RR, SPO2, SBP, DBP, MAP**
- **Incremental improvement in the YoloV8 performance + 0.7s overhead in inference --> decided to skip this enhancement**

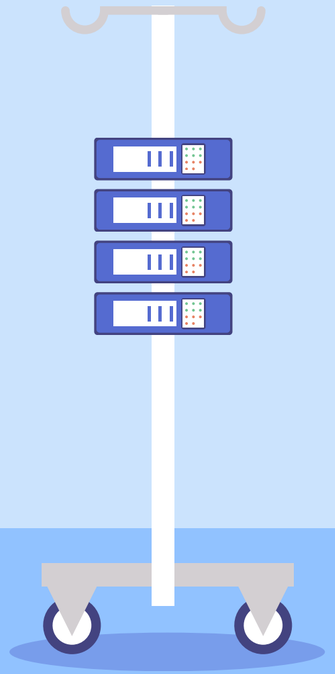
YoloV8 Training Performance

Dataset	mAP-50
Normal Dataset	**98.80%**
Synthesized images (Using Craft's output)	**99.00%**

03

Optical Character Recognition

Model	Runtime	Cons
TesseractOCR	> 1.5 seconds	Requires pre-processing, Poor performance
Paddle OCR	0.4- 0.5 seconds	Requires pre-processing, Longer runtime
ABINet	> 1.5 seconds	No pre-processing, Longer runtime
Parseq Tiny	~ 0.3 seconds	None



Final OCR Model

Parseq: Scene Text Recognition (STR) model, pre-trained on multiple real and synthetic datasets

Advantages:

- Runtime < 0.15s
- Doesn't require pre-processing

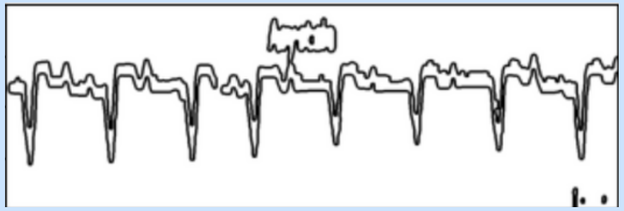
Prediction:	94	72	60	56

04

Graph Digitization

Pre-processing

Waveform Image



Contour Graph

Median Filter

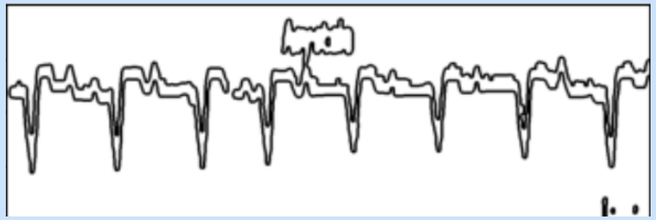
Green/ Channel Selection

Otsu Binarization

Canny Edge Detection

Contour Detection

Algorithm



Contour Graph

Choose a starting point from left most strip

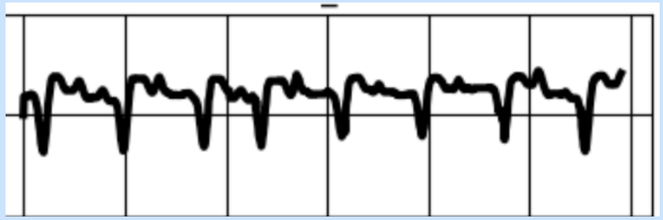
Find contours occurrences in the next strip

Repeat for all strips

Mean closest to the previous column mean is selected

Calculate the means between all such neighboring contours

Represent the contour thickness using its middle pixel



Digitized Graph

Novelty

Posing the quadrilateral bounding box problem as a 4-channel classification problem

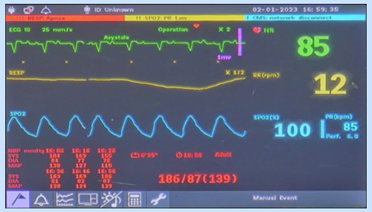
Utilising CRAFT to highlight regions of interest(ROI), to further distill information for YoloV8

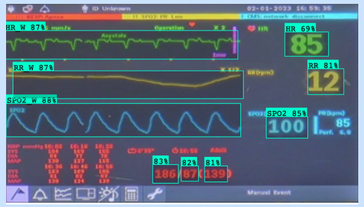
Designed a novel algorithm for graph digitization under the presence of artifacts

Average Inference Time: 1.8s !!

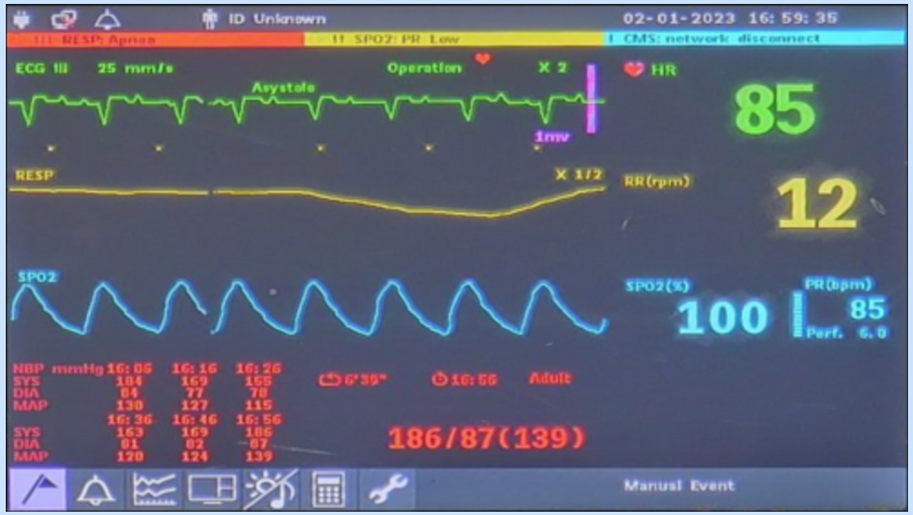
Pipeline in Action

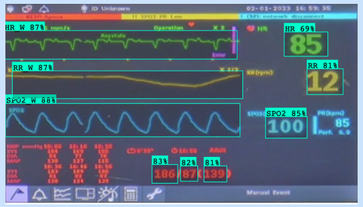
Original Image





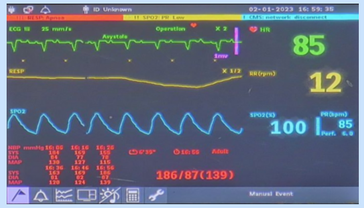
Pipeline in Action

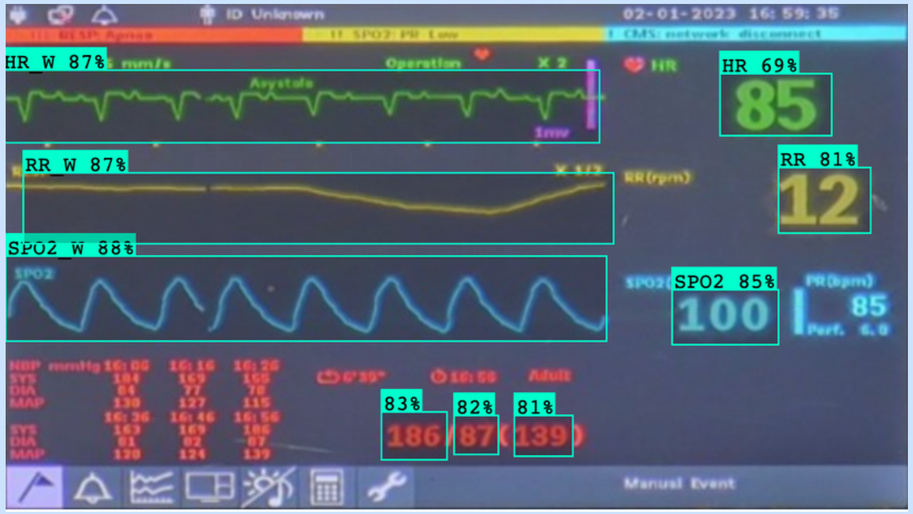




Segmented Image

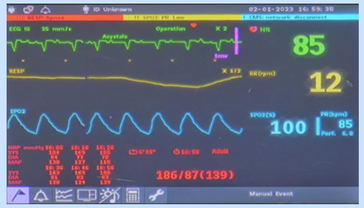
Pipeline in Action

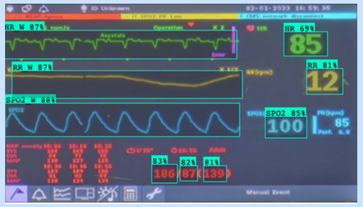


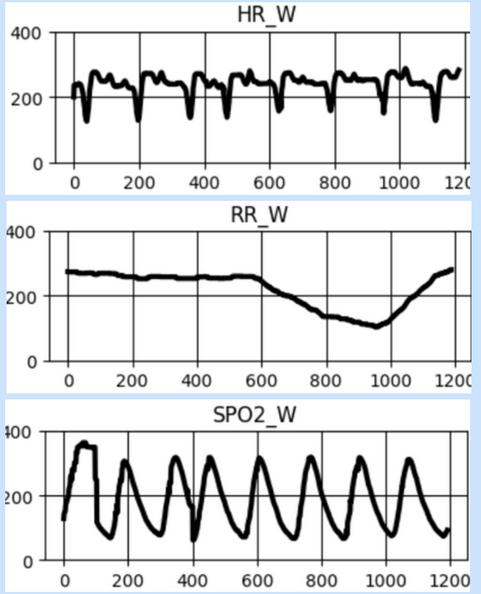


Vitals Detected

Pipeline in Action







HR:85
RR:12
SpO2:100
SBP:186
DBP:87
MAP:139

Extracted Vitals and Waveforms

Thank You!