# Task 1 - DL - Eyeglasses Segmentation Model

## Solution Description

For this task, the yolov8n-seg from Ultralytics is used.

## Data preparation

### Convert binary masks to polygons

First, we need to prepare the data. Specifically, convert it into the YOLO format, where instead of binary masks, the model accepts .txt files containing polygons of each object.

The problem is that the model considers all pixels within some polygon to be part of an object and that is not true in our case. Most of the time we need to segment eyeglasses frames without lenses. That is why we need to segment lenses separately from eyeglasses and in the end deduct lenses masks from the eyeglasses masks.

Other that that, data conversion is straightforward:
To convert binary mask to yolo-polygons we need:
1. find contours in the mask
2. Convert contours into polygons. External contours append to the 'glasses' class, internal - to the 'lens' class.
3. Write polygons into .txt files

## Rearrange data folder and create dataset.yaml file

Now we need to rearrange the data so yolo can accept it (we can do it manually):

```
data/
├── train/
│   ├── images/
│   │   ├── img1.jpg
│   │   ├── img2.jpg
│   │   └── ...
│   └── labels/
│       ├── img1.txt
│       ├── img2.txt
│       └── ...
├── val/
│   ├── images/
│   │   ├── img1.jpg
│   │   ├── img2.jpg
│   │   └── ...
│   └── labels/
│       ├── img1.txt
│       ├── img2.txt
│       └── ...
```

Then, create a dataset.yaml file:

```
names:
- glasses
- lens
nc: 2
path: /kaggle/input/glasses-segm/data
train: images/train
val: images/val
```

Now we can train our model

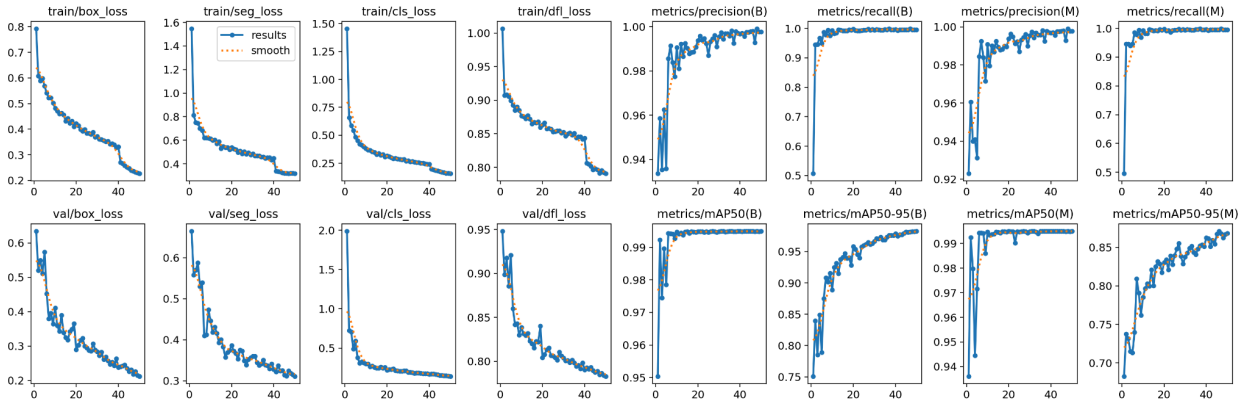
# Model Training

## Training configuration

Model training has been done on kaggle
The model trained for 50 epochs with a batch size 32 (~ 30 min on GPU P100).
Other parameter of the model were default

## Losses and Scores

Here we can see losses and scores during training:



# Validation Batches:


# Overall Result

## CPU VS GPU

On average it takes 140ms for the CPU to inference one image. And 8ms for GPU (GPU P100)

## Actual Performance

Below we can see the actual performance of the model on the test data: