Final Report : *Teeth abnormality classification in Panoramic x-rays*

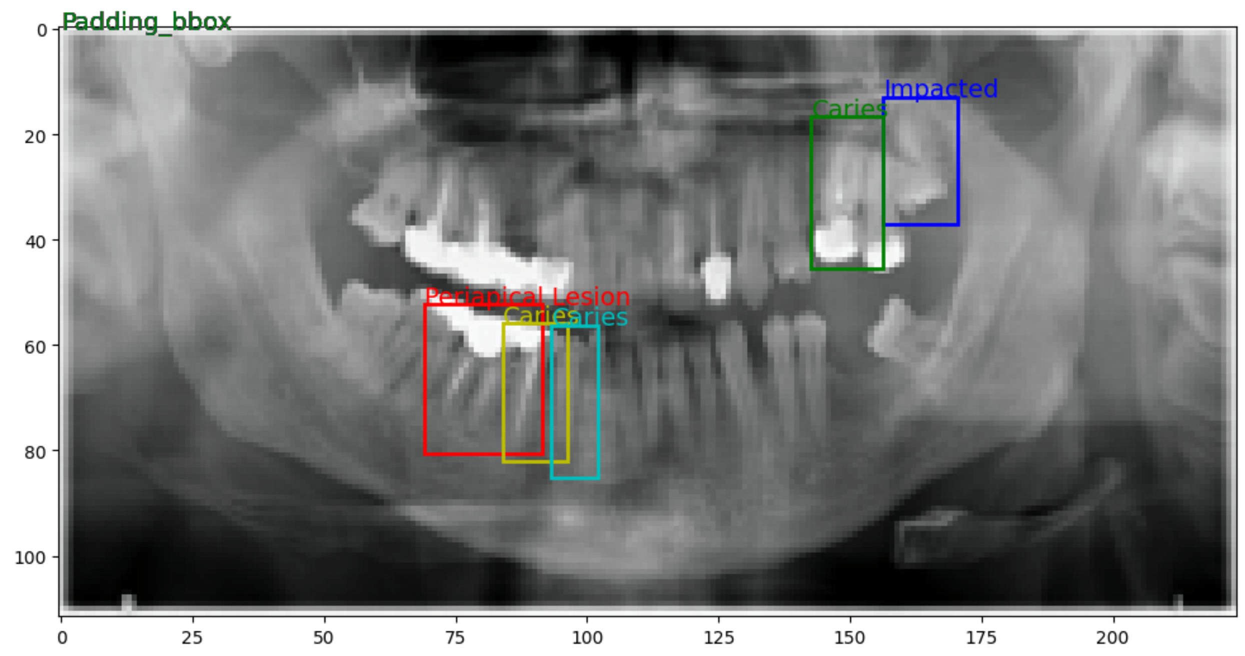
In the following project, we are seeking to **detect** and **classify** abnormal teeth given an x-ray scan.

1. Dataset and preprocessing

The dataset for teeth abnormality classification comprises x-ray images and corresponding annotation files. The relevant annotations for this task pertain to the quadrant-enumeration-disease category. Each image is annotated with all detected diseases. Abnormal teeth in each image are marked with bounding boxes (detection task) and disease labels (classification task). The data folder will be processed to generate the following PyTorch training dataset:

Image_Idx --> [all_bbox], [all_labels]

This dataset is divided into training (80%) and validation set (20%). The validation set is used to evaluate the model's performance. All images are resized to 640*480 for uniformity. The following picture is a dataset sample.



As we can see all abnormal teeth are labeled and are located by bboxes. Furthermore a padding_box [0,0,0,0] is added to all instances to match raw data requirements when building the dataset.

2. Desired evaluation Metrics

The evaluation metrics will include accuracy, precision, and recall.

- **Accuracy**: Accuracy will be measured on the entire validation dataset. It represents the number of correct classifications divided by the total number of classifications in the x-rays. This will simply give the amount of detected diseases.

- **Sample Recall**: Sample recall will be measured for a specific x-ray scan. For a given disease, disease recall will first measure how many cases are correctly classified among all actual instances of the disease in the scan. The sample recall is then the average of all disease recalls within that scan.

- **Average Recall**: Average recall is measured over the entire validation dataset. It is the average of all sample recalls.

- **Sample Precision**: Sample precision will first be measured at the disease level and then averaged over all detected diseases. It indicates, among all predictions of a specific disease, how many correspond to the exact same disease.

- **Average Precision**: Average precision is measured over the entire validation dataset. It is the average of all sample precisions.

The accuracy and the average precions and recall will be used to evaluated the fianal performance of the model on the validation set.

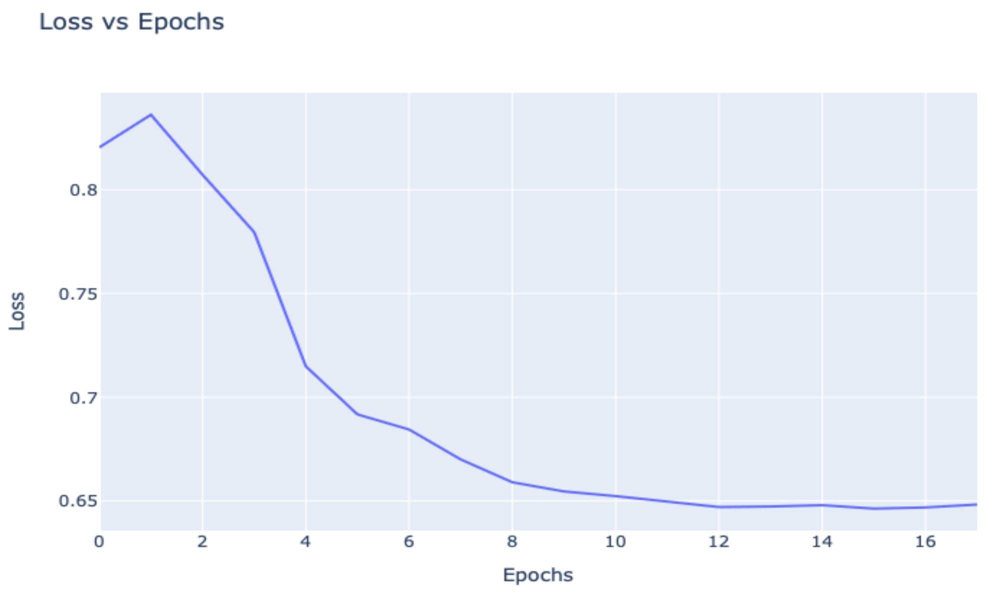
3. Model training

The selected model for the classification task will be the faster-RCNN. It is initialized with default pretrained weigths and is fine-tuned using the training dataset. Before feeding training data into the model, padding bboxes and labels are filter out to avoid biases.

The model performance is optimized using the combined classification and detection losses.

The trainning is done using the following defined methods

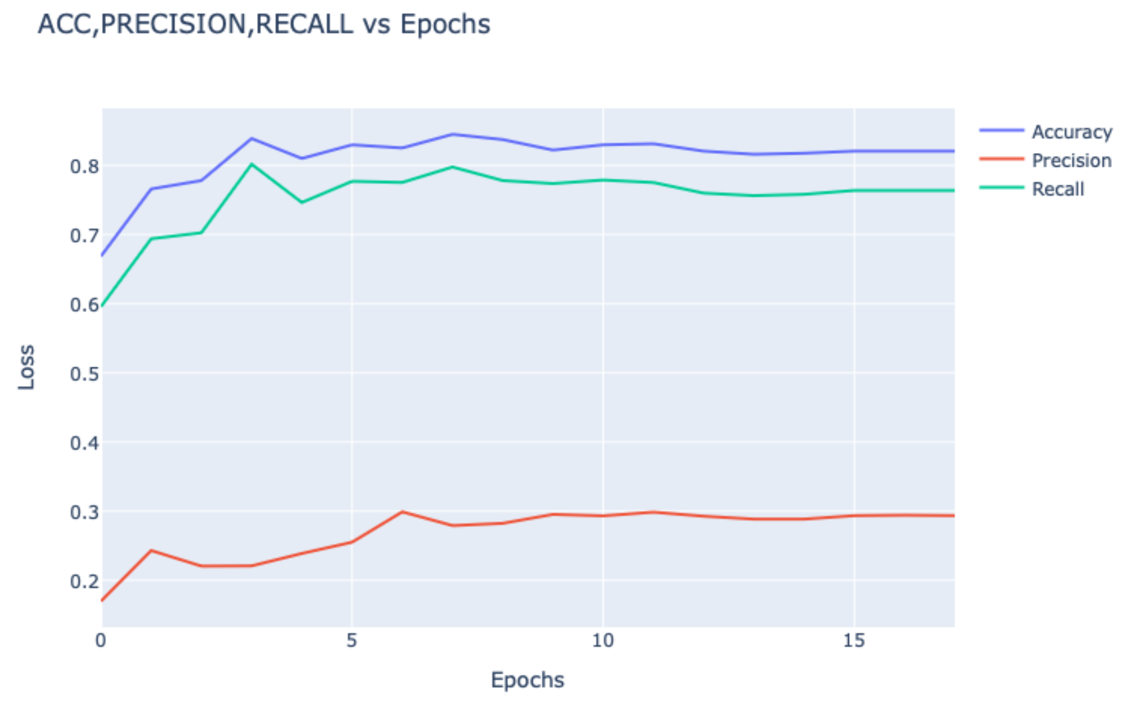
- **Averager** : Python class to average loss over epochs iterations
- **training_loop** : Train a model on a train_set and validate on a val_set. Return trainning losses over iteration and ACCURACY, AVG_PRECISION, AVG_RECALL over iteration on the validation set.
- **plot_training_results** : Takes as training_loop outputs as input and plot them
- **evaluation** : Compute the ACCURACY, AVG_PRECISION, AVG_RECALL on a given set
- **compute_metrics** : Compute the ACC, AVG_PRECISION, AVG_RECALL on a given sample prediction This function is used in evalution as helper.
- **process_bbox** : Convert back bbox in the desired format after model output
- **apply_nms** : As the model output a lot of overlapping bboxes, this function used a thershold to only select non-overlapping bboxes, using non maximum suppression. This help to make prediction clear and avoid duplicated predictions.



The training process over 17 epochs, show that the training loss is improved, going from 0.83 on the pretrained weight to 0.65 after the optimization of the model weight. After various experimations, stochastic gradient descent with the following parameters result to the best optimizer :

- learning rate : 0.005
- momuntum : 0.9
- weigth_decay : 0.0005

In addition, over training epochs, the model has been evaluated on the validation data set. Resulting to the following graph



As we can see, the trainning process improved both metrics. The accuracy has been increased by 12%, the recall by 16% and the precision by 10%.

4. **Model evaluation**

The final metrics on the tranning and validation set are the following :

- **Train (Accuracy) : 0.84**
- **Validation (Accuracy) : 0.82**
- **Train (Avg Precision) : 0.30**
- **Validation (Avg Precision) : 0.29**
- **Train (Avg Recall) : 0.79**
- **Validation (Avg Recall) : 0.76**
- **Train (Avg F1) : 0.44**
- **Validation (Avg F1) : 0.42**

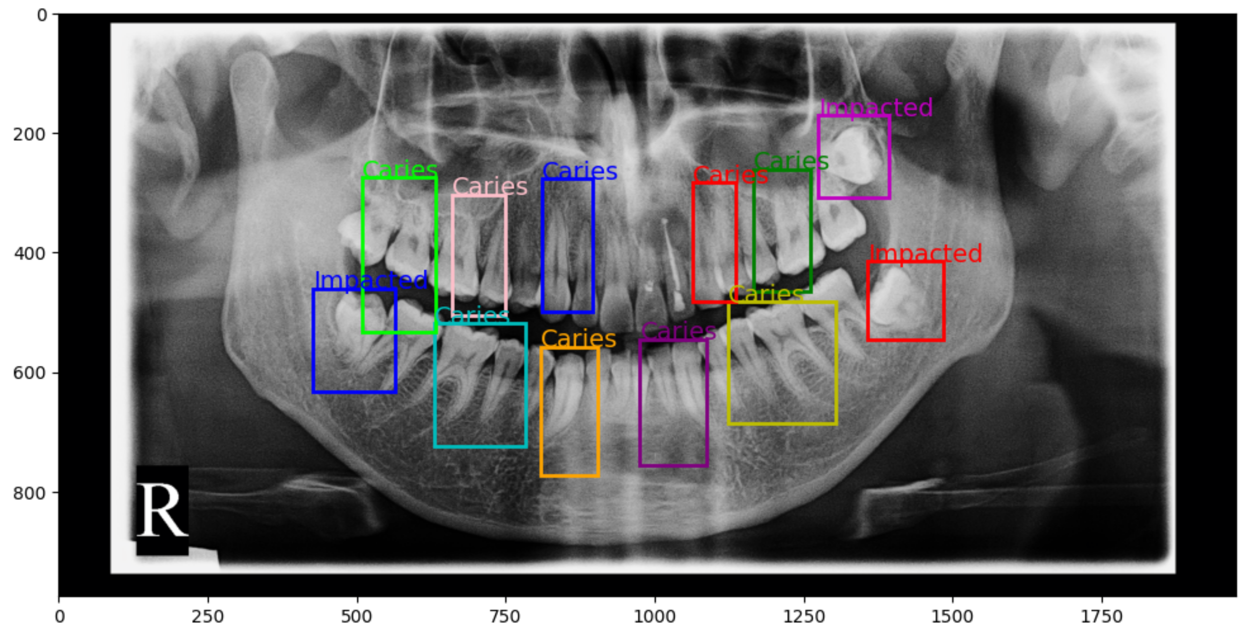
First, we observe that the performance metrics for the train set and validation set are quite similar. This indicates that the model has successfully avoided overfitting.

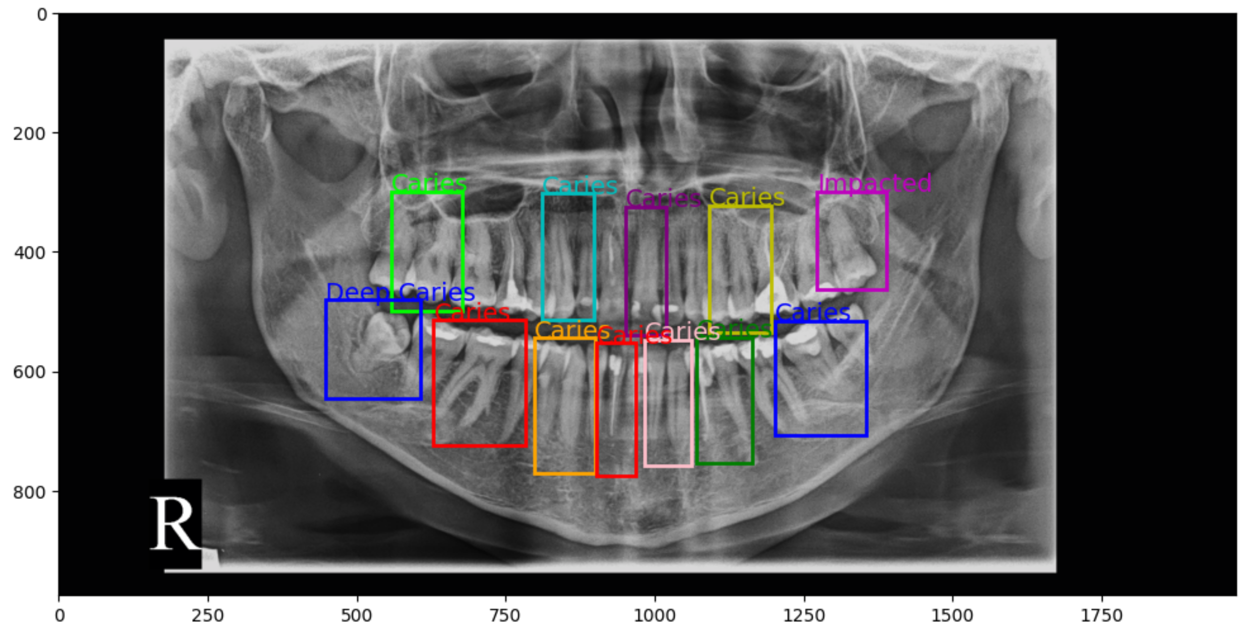
With higher accuracy and recall, the model demonstrates an acceptable capacity to detect and classify all abnormal teeth. However, the lower precision is a downside, as the model might overestimate the number of actual cases of disease to ensure all cases are detected. This results in poor precision.

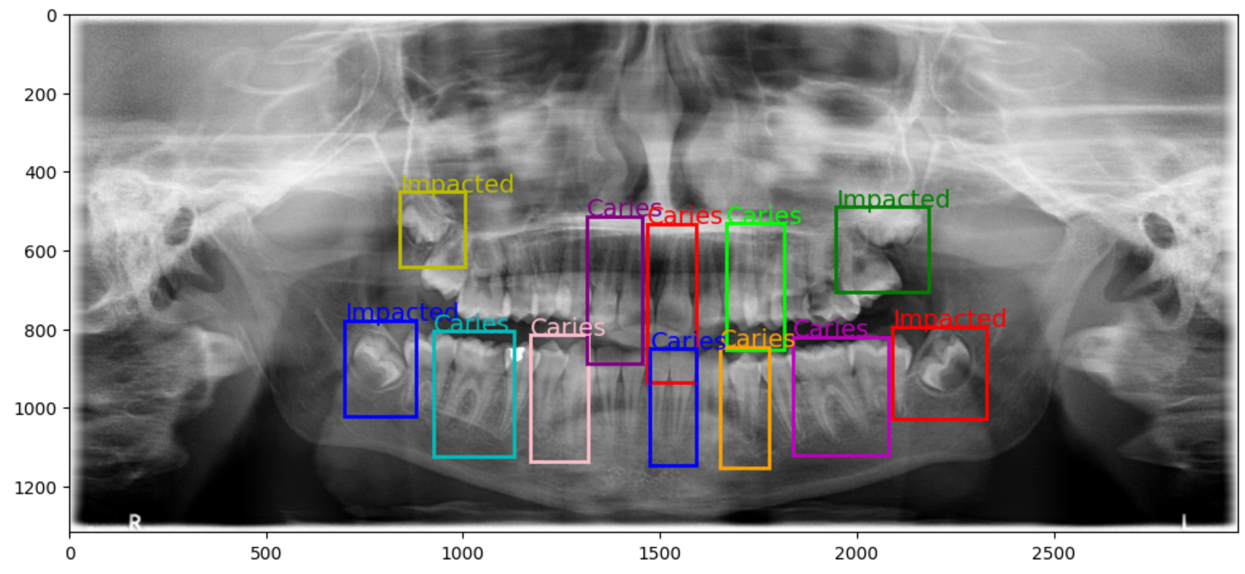
An important point to note is the F1-score of **0.42**, which averages precision and recall on the validation set. During training experiments and model improvements, it is possible to achieve nearly equal precision, recall, and accuracy (around 0.42), or to have higher precision instead of higher recall. Both approaches result in a similar approximate F1-score.

The final model was selected because it is more beneficial to detect all abnormalities, even at the cost of lower precision. Therefore, higher recall was prioritized.

5. Some Prediction on test images







As we can see, the detection model performs pretty correctly as it locates the some real abnormal teeth and correct classifications. But the number of detections is not as confident, as we prioritized recall over precision.

6. Reaching better performance with over model ?

The task performance could be improved by using diffusion-based hierarchical multi-labeled object detection. This new approach leverages a three-level encoder-decoder structure, optimizing the detection and classification tasks sequentially. By transferring weights between levels, this method can achieve substantially better performance than Faster R-CNN, which improves the model using a combined loss function. Implementing this model was challenging due to the lack of available documentation resources.