# Final Project — AI Dental Diagnosis Model

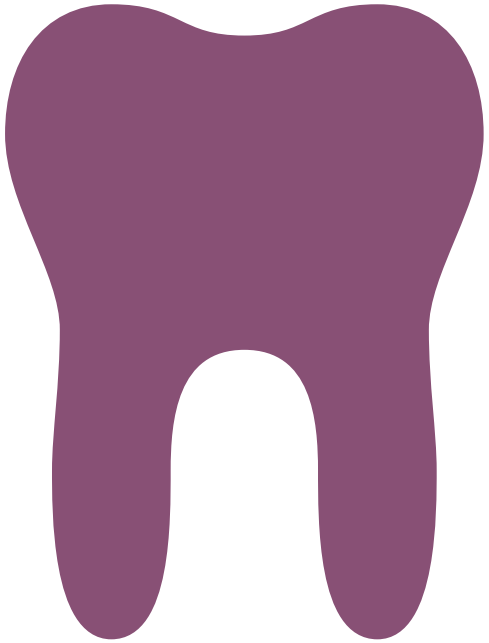
Artificial Intelligence

P25-LIS3082-2

Dr. Alejandra Hernández Sánchez

Emiliano Ruiz Plancarte

177478

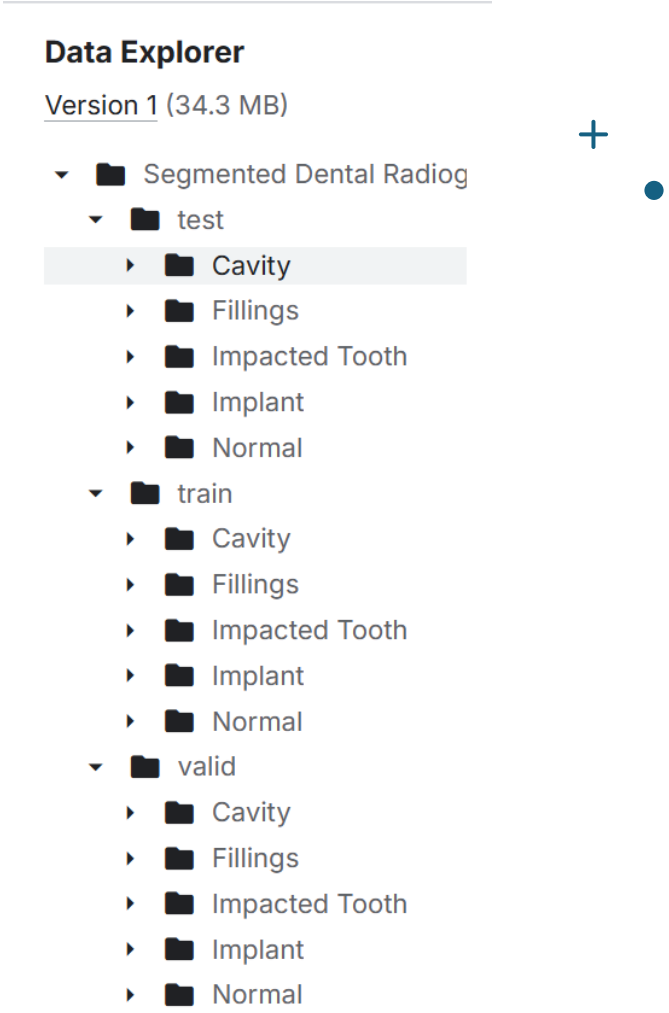


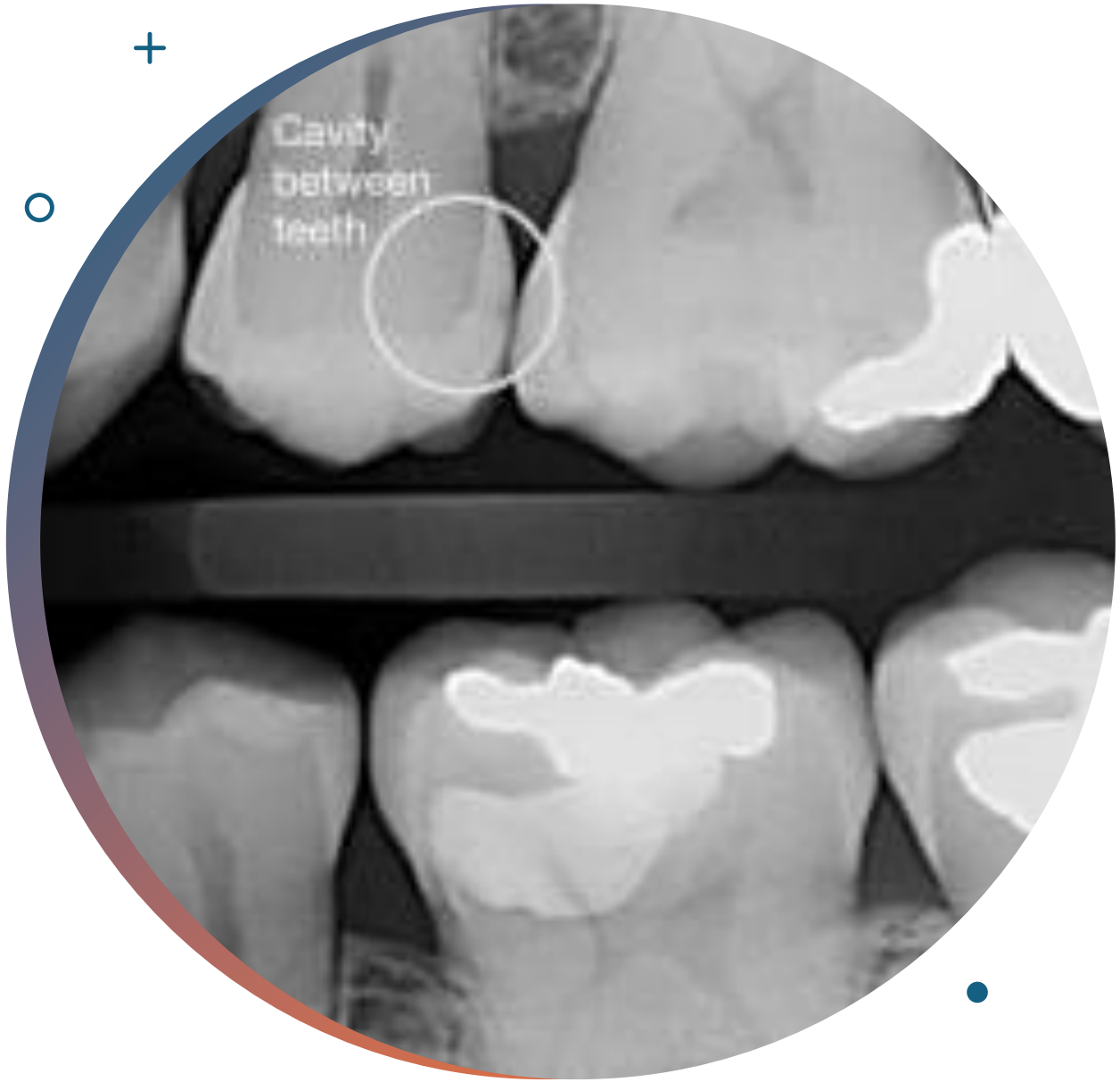
# INTRODUCTION

- Apply AI algorithms and techniques to solve a real-world problema in dentistry through image-based diagnosis.
- Original focus: detecting cavities
- Expanded goal: more conditions like
  - Dentall fillings
  - Impacted teeth
- Purpouse:
  - Accurate early diagnosis
  - Better patient outcomes

# METHODOLOGY

- Python (Google Colab) codes.
- Large dataset (already sorted and labeled).
- MobileNetV2 model — pre-trained for image processing.
- Plotting metrics for better understanding of the model.






# Experimental Results

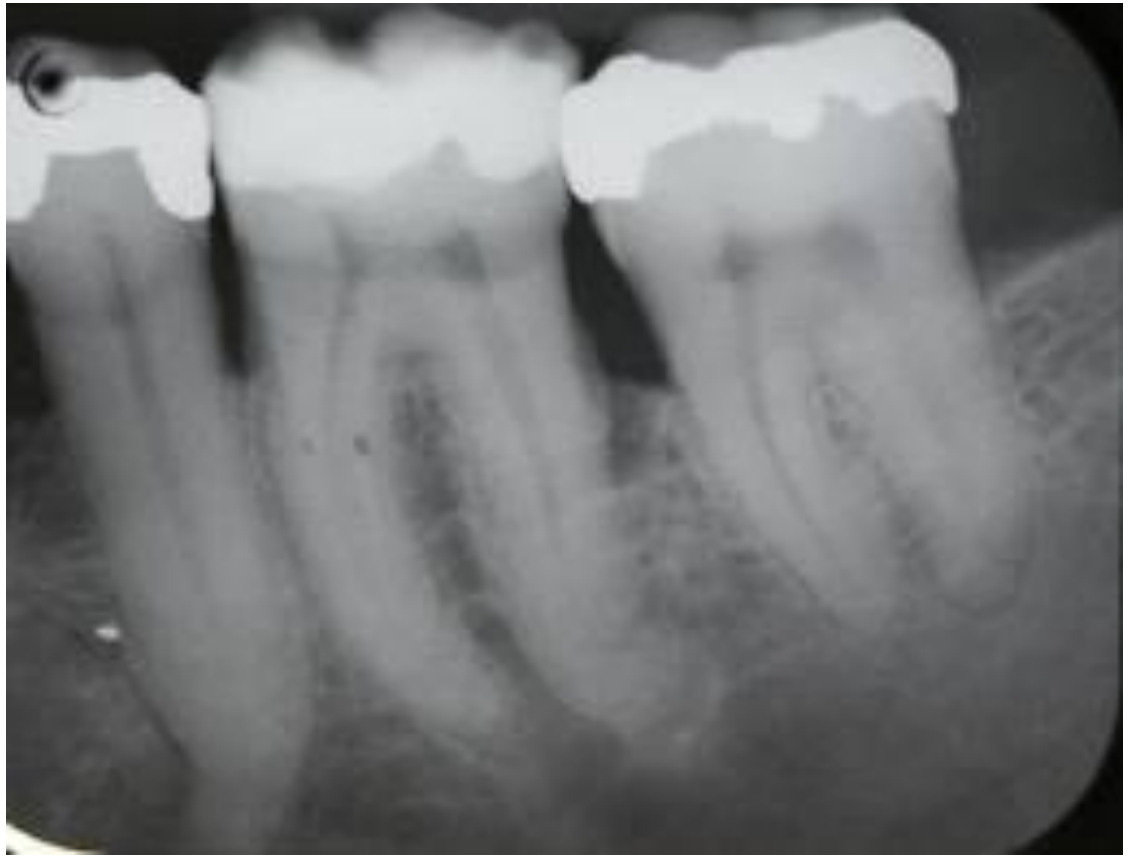
## What helped

- All 3 main folders contained uniform labeled data:
  - Cavity, Fillings, Impacted Tooth, Implant, Normal
- Consistency made initial setup and label handling easier.

# Experimental Results

## Early Changes

- Poor performance on the Cavity class:
  - Low accuracy & F1-score
- Initial model coudn’t generalize well for minority classses (before MobileNetV2).



# Experimental Results

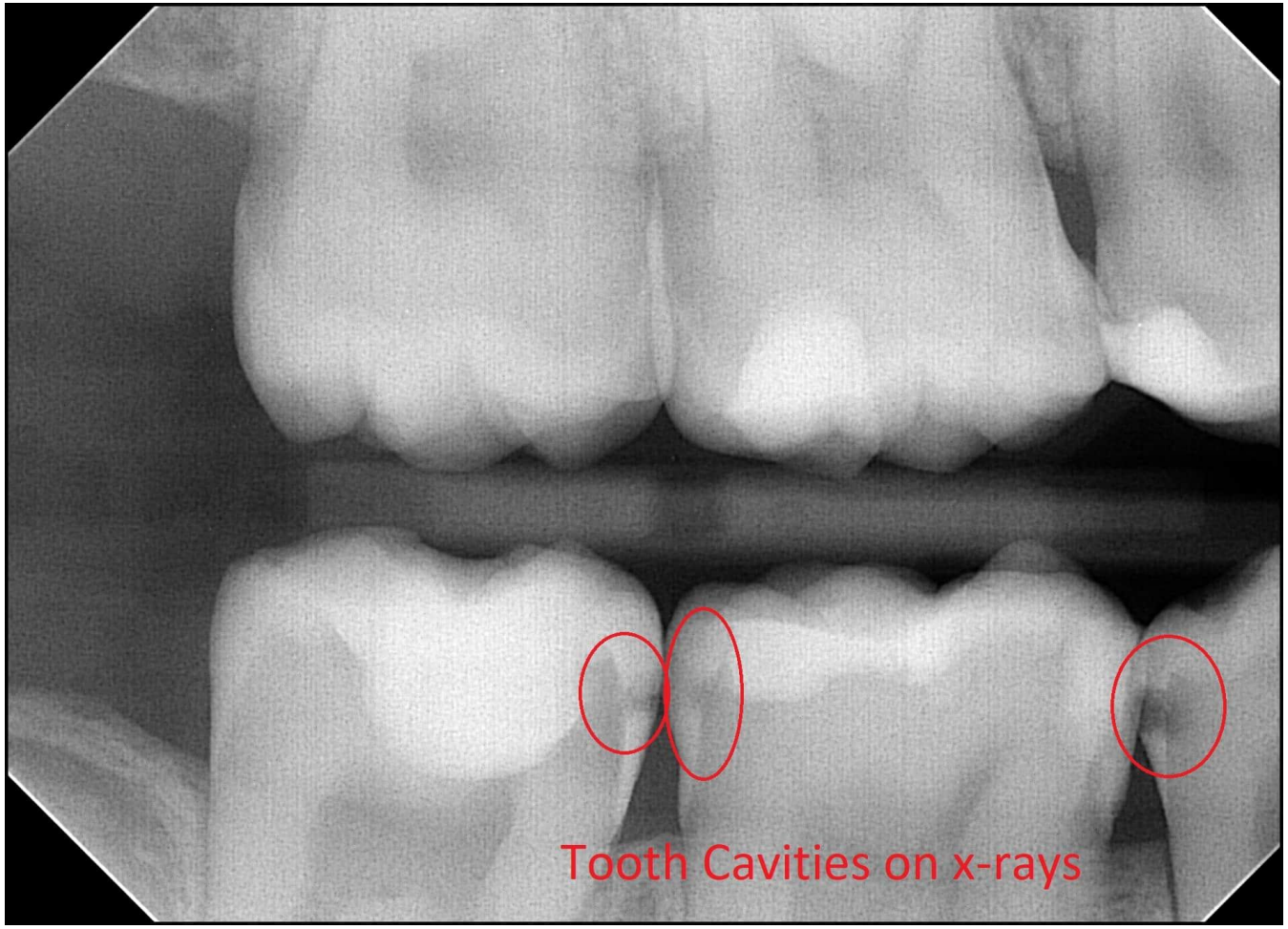
## Solutions Attempted

- Applied data augmentation
- Switched to MobileNetV2
- Faced long runtime (~6 hrs)
  - Large dataset (~25, 000 images)
  - High-resolution inputs
  - Too many training epochs

# Experimental Results

## Final adjustments

- Reduced image size.
- Lowered number of epochs.
- Significantly improved training efficiency.



# Experimental Results

## Code snippets

- Image size (reduction)
- Training

```
#### REST OF THE CODE ####

# Image size (reduced for faster performance)
img_height, img_width = 128, 128
batch_size = 32

#### REST OF THE CODE ####

#Train
epochs = 15
history = model.fit(train_generator,
                    validation_data = val_generator,
                    epochs = epochs,
                    class_weight = class_weights)}

#### REST OF THE CODE ####
```

# Experimental Results

## Code snippets

```
#### REST OF THE CODE ####

# Compute class weights

class_indices = train_generator.class_indices

inv_class_indices = {v: k for k, v in
  class_indices.items()}

labels = train_generator.classes

class_weights_arr = compute_class_weight(class_weight =
  'balanced', classes = np.unique(labels), y = labels)

class_weights = dict(enumerate(class_weights_arr))

print("Class Weights:", class_weights)
```

# Experimental Results

## Code snippets

- Focal los
- Compilation with focal loss

```
# Define focal loss
def focal_loss(gamma = 2., alpha = 0.25):
    def focal_loss_fixed(y_true, y_pred):
        y_pred = tf.clip_by_value(y_pred, 1e-8, 1.0 - 1e-8)
        cross_entropy = -y_true * tf.math.log(y_pred)
        weight = alpha * tf.pow(1 - y_pred, gamma)
        return tf.reduce_mean(tf.reduce_sum(weight *
  cross_entropy, axis = 1))
    return focal_loss_fixed


#### MORE FUNCTIONS OF THE CODE ####


# Compile model with focal loss
model.compile(optimizer = 'adam',
              loss = focal_loss(gamma = 2., alpha = 0.25),
              metrics = ['accuracy’])
#### REST OF THE CODE ####
```

# Experimental Results

## Code snippets

- Report (CSV file).

```
#### REST OF THE CODE ####


# Report


report = classification_report(true_classes, pred_classes,
  target_names = class_labels, output_dict = True)

report_df = pd.DataFrame(report).transpose()

report_df.to_csv("dental_classification_report_mobilenet.csv")


print("Classification report saved as
  'dental_classification_report_mobilenet.csv'")

#### END OF THE CODE ####
```

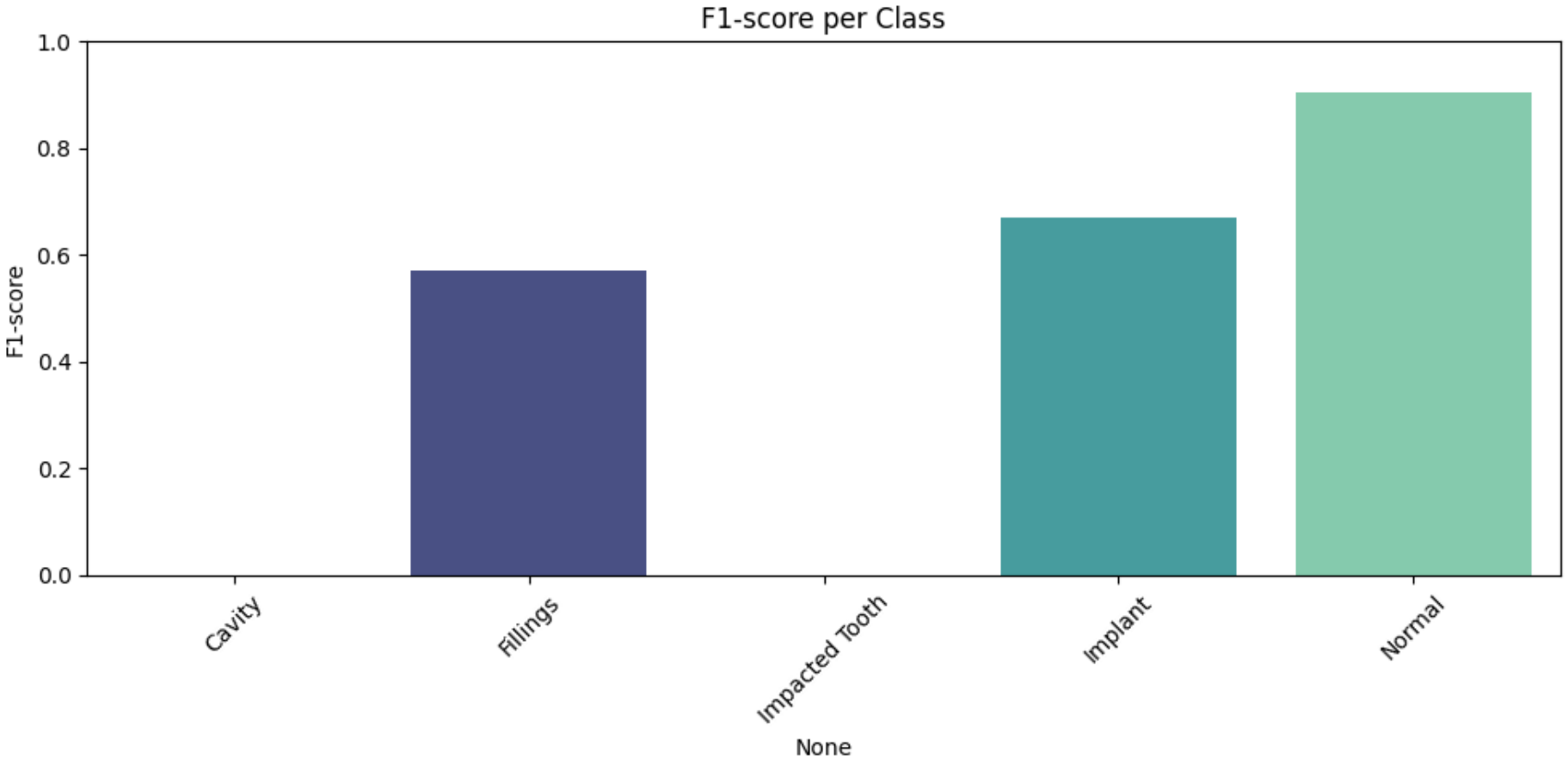
# Experimental Results

## Table of metrics

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Cavity | 0.0000 | 0.0000 | 0.0000 | 22 |
| Fillings | 0.8046 | 0.4444 | 0.5726 | 315 |
| Impacted Tooth | 0.0000 | 0.0000 | 0.0000 | 32 |
| Implant | 0.8194 | 0.5673 | 0.6705 | 104 |
| Normal | 0.8338 | 0.9863 | 0.9037 | 1241 |
| Accuracy | | 0.8302 | | |
| Macro avg | 0.4916 | 0.3996 | 0.4293 | 1714 |
| Weighted avg | 0.8013 | 0.8302 | 0.8002 | 1714 |

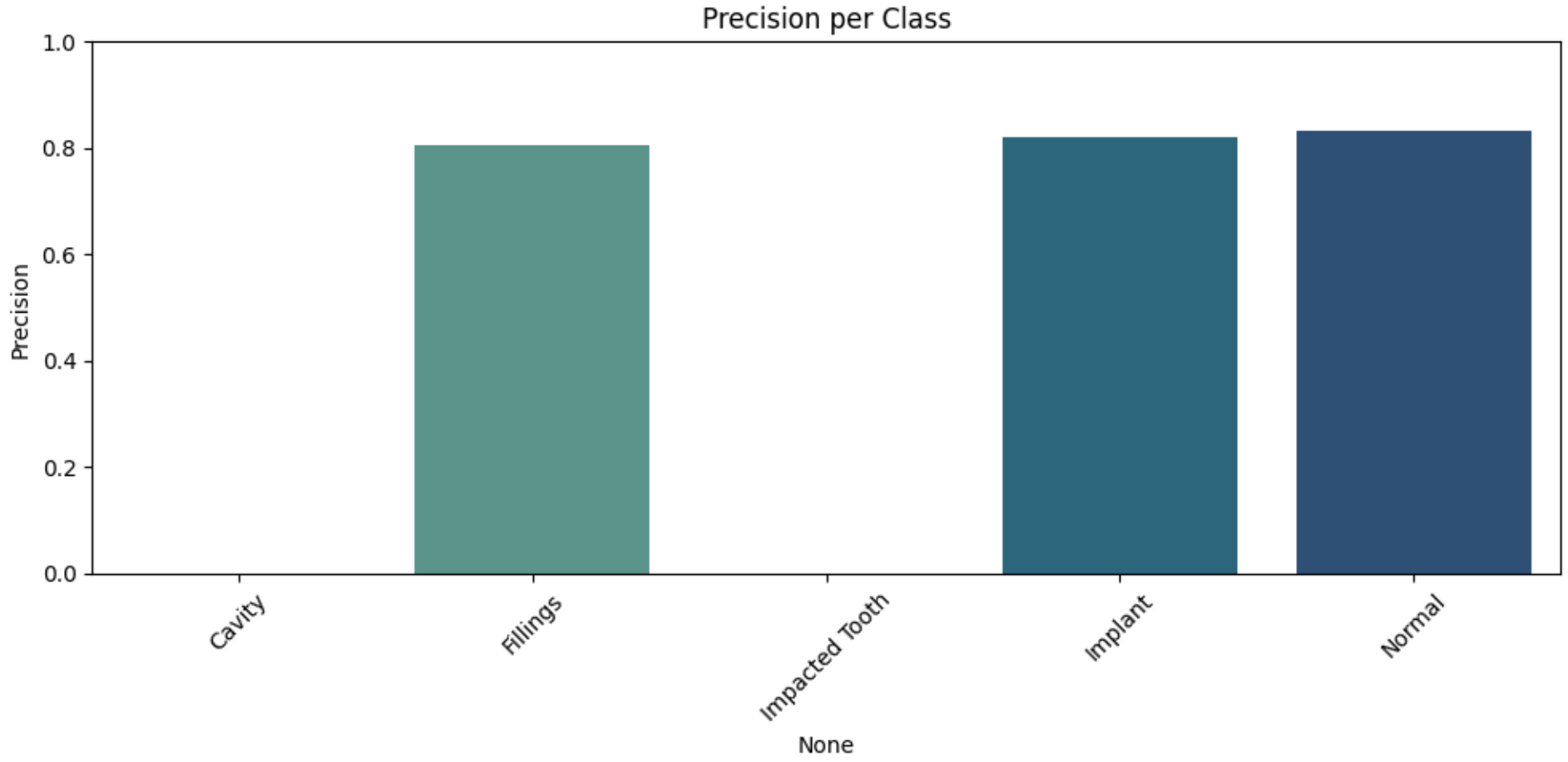
# Experimental Results

## F1-Score



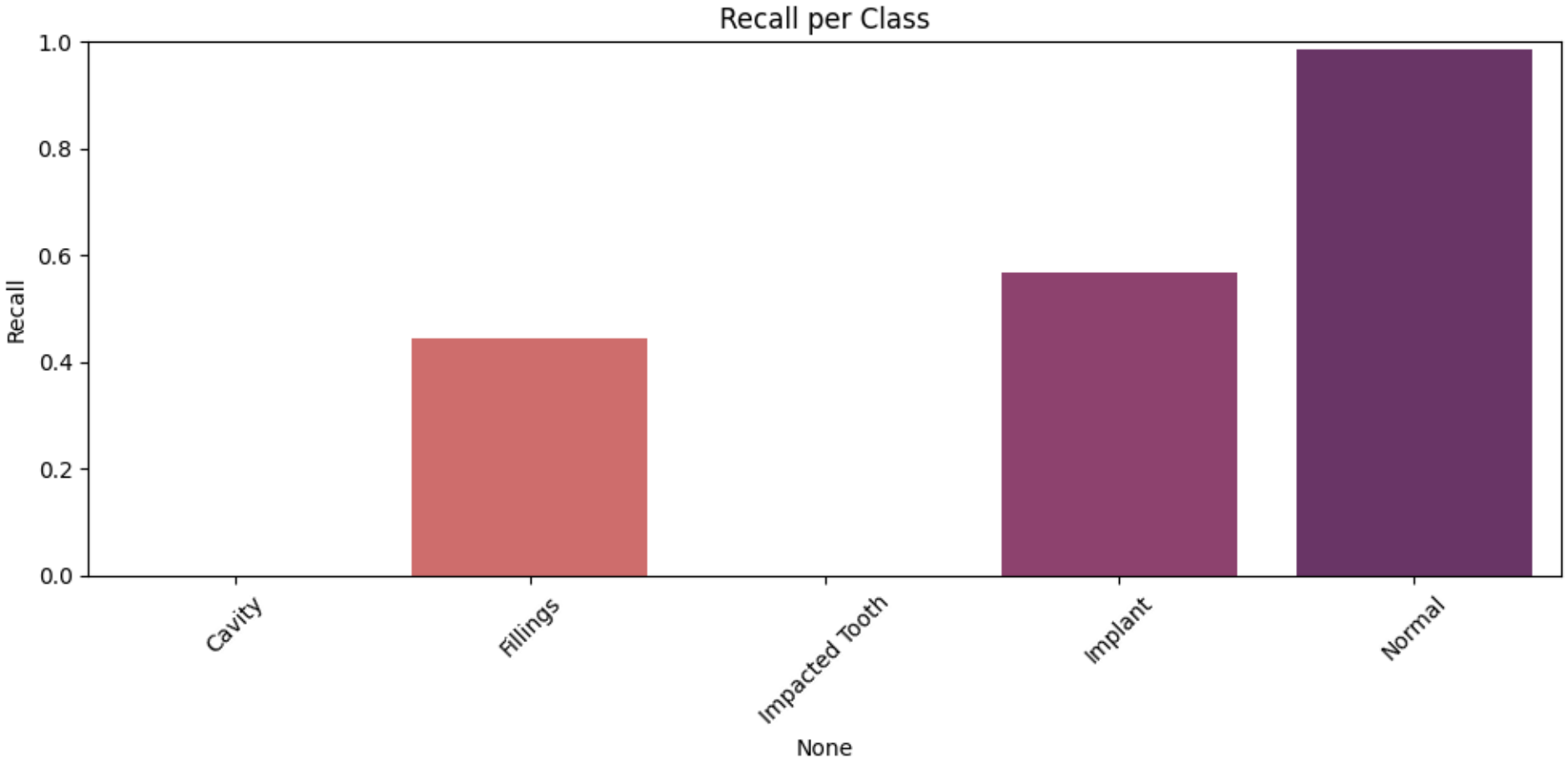
# Experimental Results

## Precision per class



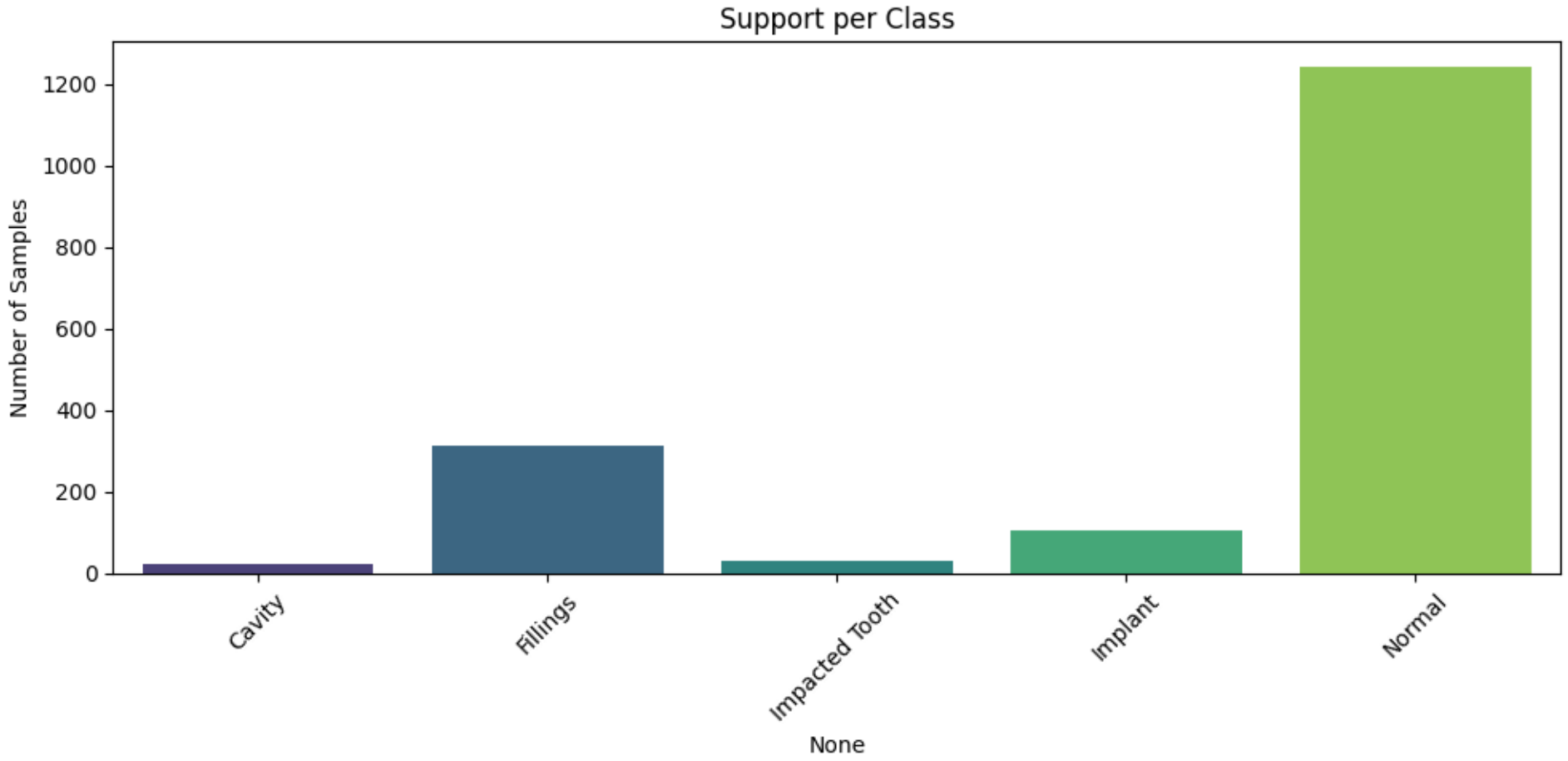
# Experimental Results

## Recall per class



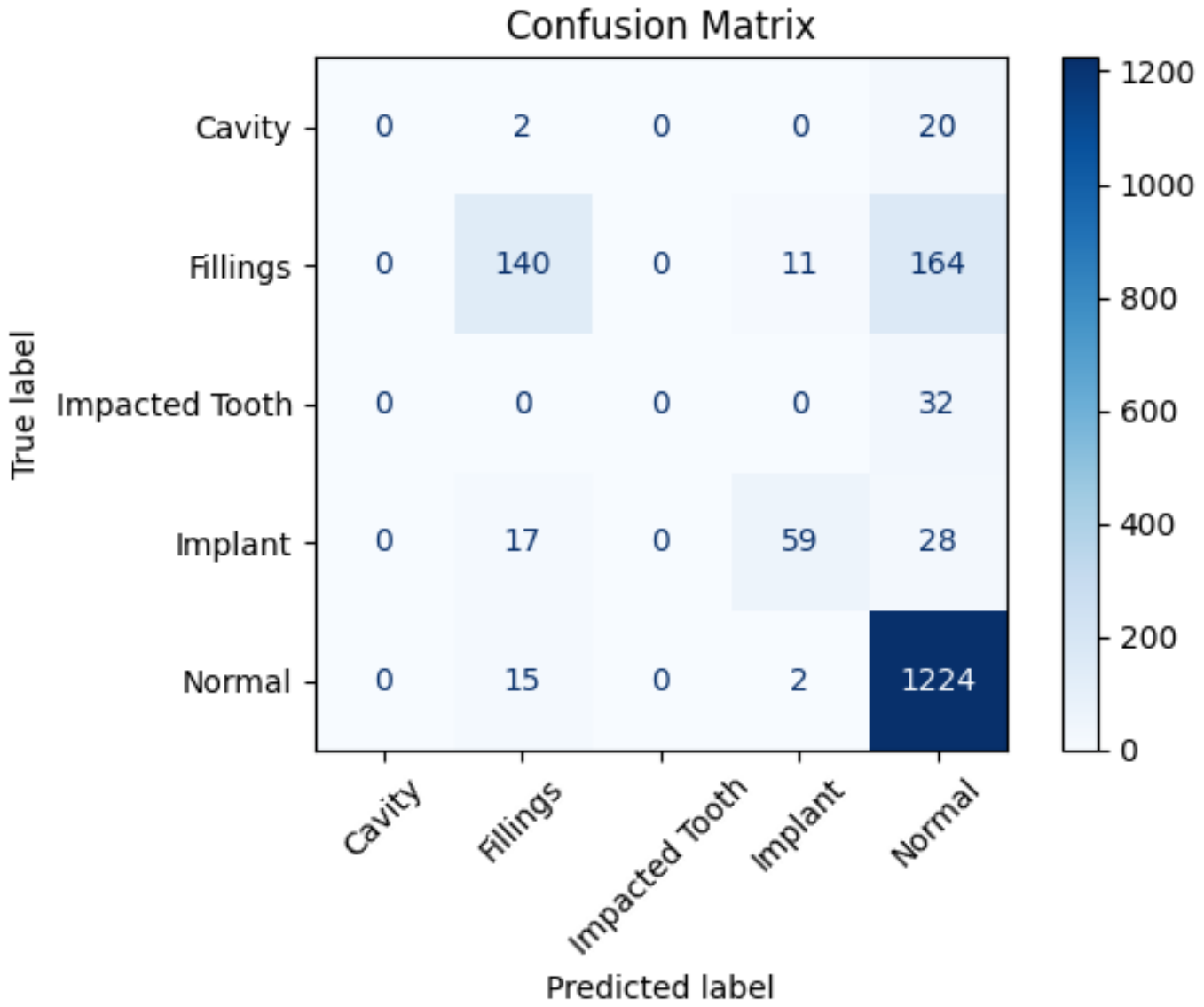
# Experimental Results

## Support per class



# Experimental Results

## Confusion Matrix



# CONCLUSIONS

- Project impact
  - Applied key AI concepts and techniques from the course.
- Challenges & learning
  - Strengthened understanding of: model selection, data preprocessing, performance tuning.
- Future potential
  - Approach is scalable to other areas of healthcare (non-invasive diagnostics and AI-assited medical tolos)