Dent.py: Training Dental Models from Zero to Hero

Part 1 of 4

# Planning for the Future

# Objectives

- Pipeline Overview
  - Understand the purpose of building a pipeline and identifying its inputs and outputs.
  - Define a pipeline and its modular components.
- Pipeline Components
  - Examine the stages (building blocks) of a pipeline.
- Input and Output Specifications
  - Inputs: Discuss modalities and Dimensionality.
  - Outputs: Define tasks.

# Pipelining AI Tasks

# What is a Pipeline?

**Pipeline** is a structured sequence of processes or steps

- Transforms raw data into meaningful insights or outputs
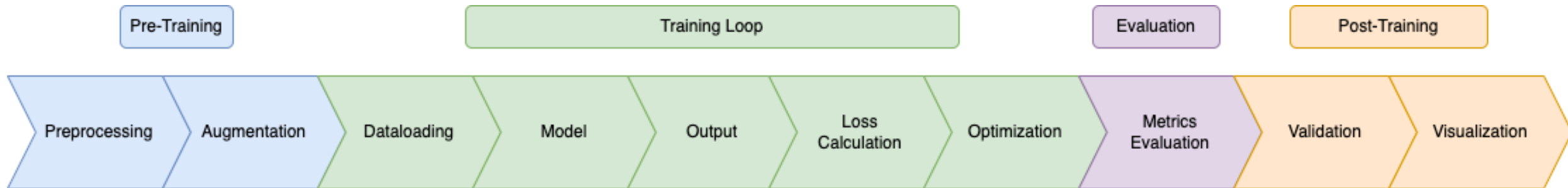
## Multistep

## Modular

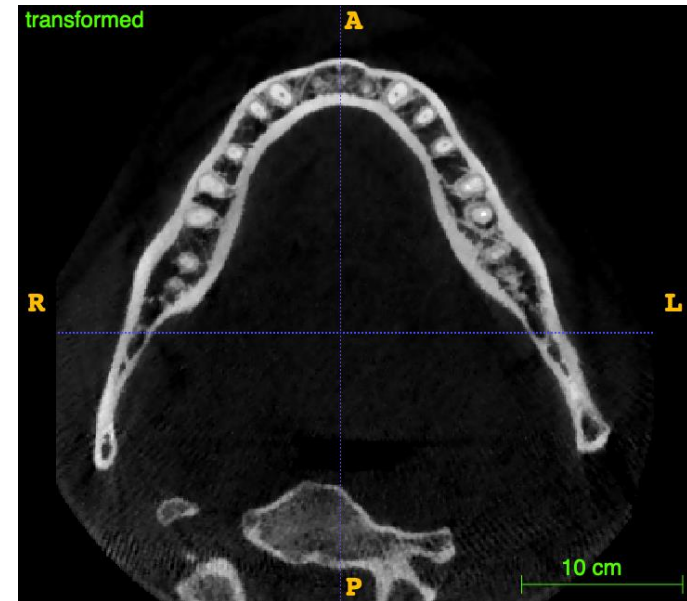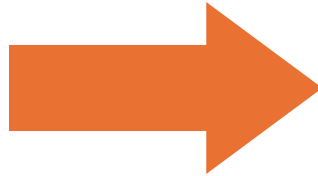- Step can be customized or replaced

## Adaptable

- Different tasks
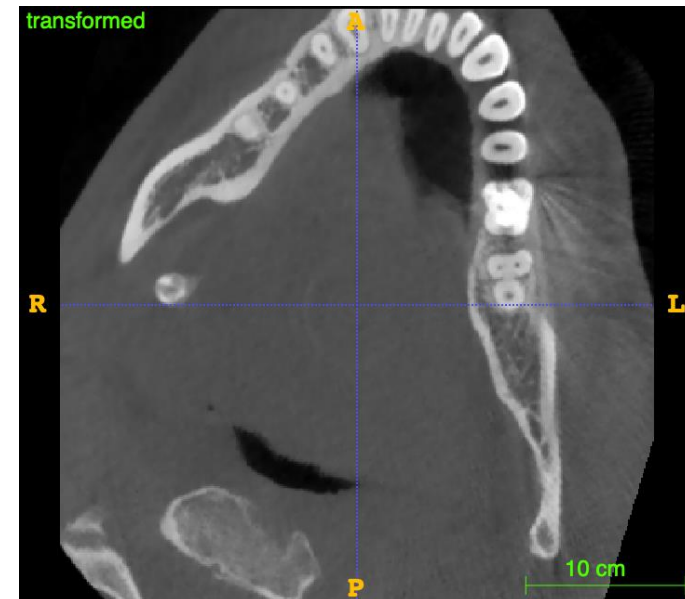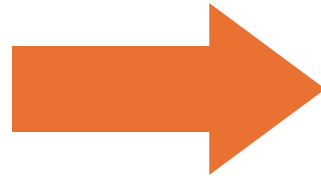- Different data types

# Pipeline Stages

# Pre-Training

- **Preprocessing**
  - Transform raw data into a clean, structured format suitable for model input
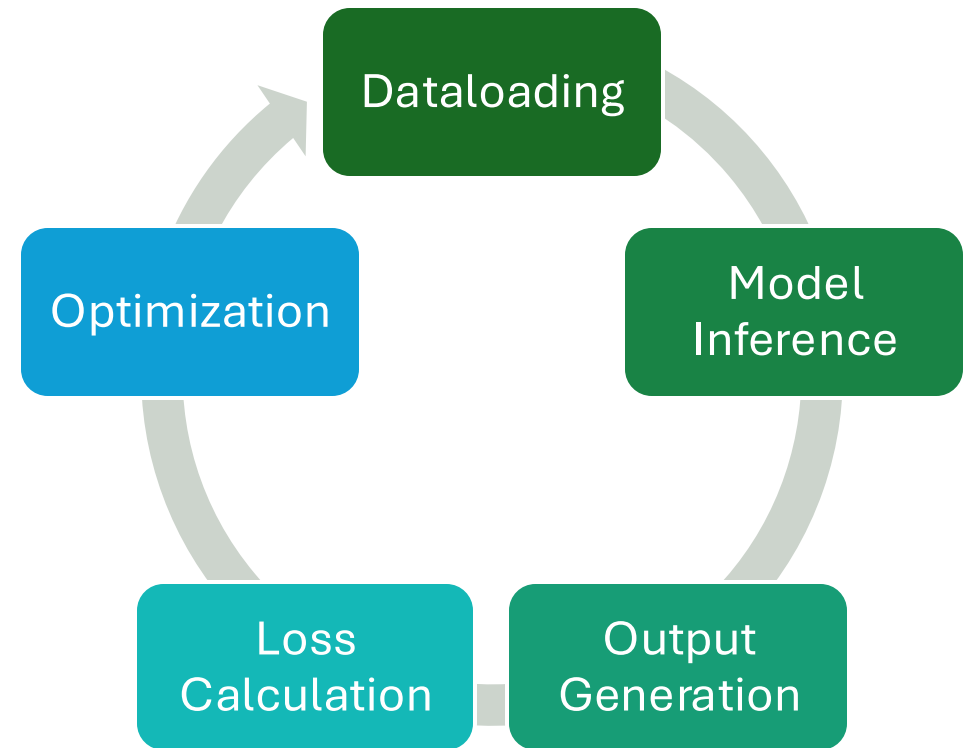
# Pre-Training

- **Augmentation**
  - Enhance data by applying transformations to create diverse, robust samples for training
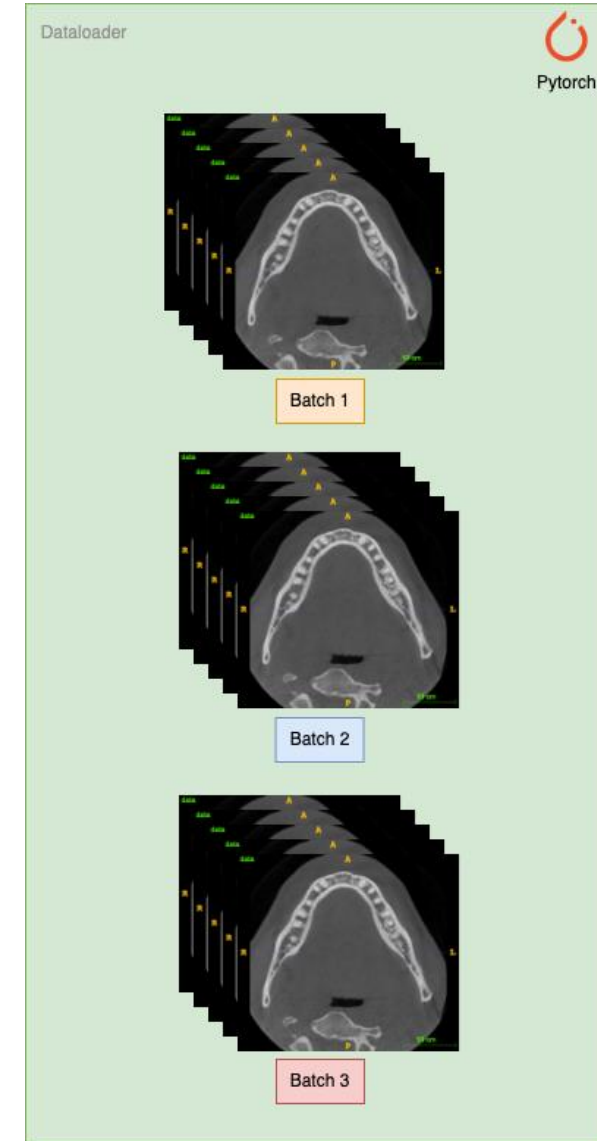
# Training Loop

- **Iterative**
  - Consists of multiple iterations called epochs
- Epoch consists of multiple activities
  1. Dataloading
  2. Model Inference
  3. Output Generation
  4. Loss calculation
  5. Optimization

# Training Loop

- **Dataloading**
  - Efficiently load and batch data to feed into the model during training and evaluation.

# Training Loop

## Model Inferencing

Use the trained model to make predictions on new or unseen data.
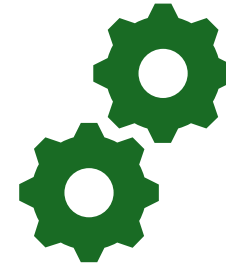
## Output Generation

Generate predictions or classifications as the final model result.

# Training Loop

## Loss Calculation

Measure the difference between predicted and true values to guide model improvements.
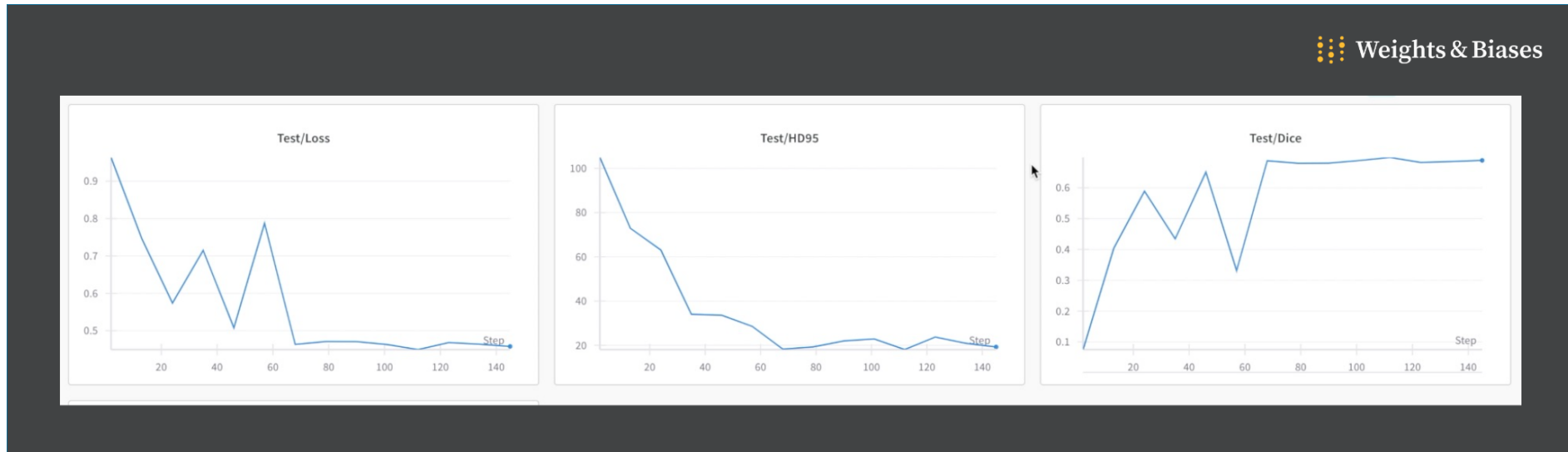
## Optimization

Adjust model parameters to minimize the loss and improve accuracy.

# Evaluation

- **Metrics Evaluation**
  - Assess model performance using quantitative metrics to ensure it meets the objectives.

# Post-Training

**Validation**

Test the model on a separate dataset to check generalization and avoid overfitting.
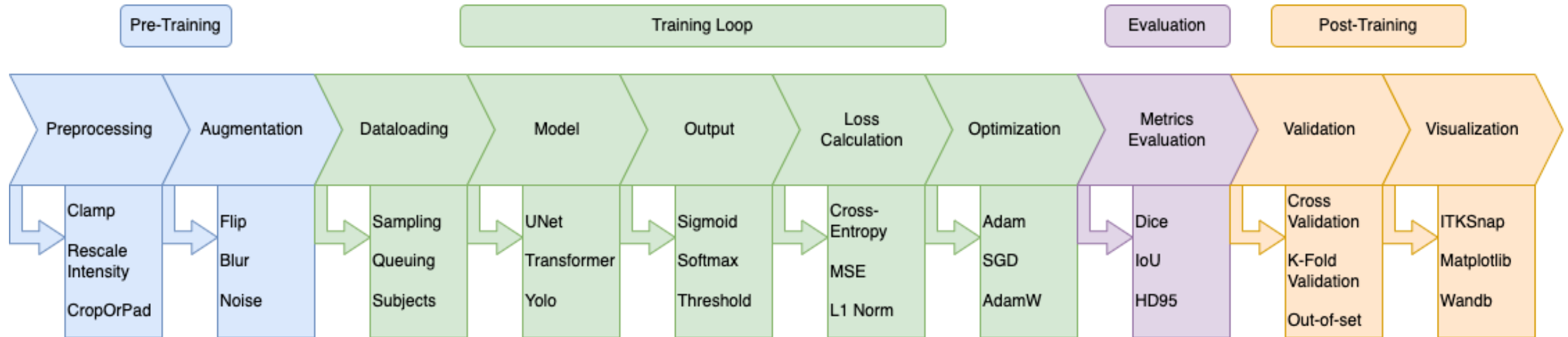
**Visualization**

Present data, model performance, or results in graphical formats to aid interpretation and analysis.

# Pipeline Flexibility

Flexibility allows for interchangeable steps

| Pre-Training | | Training Loop | | | | | Evaluation | Post-Training | |
|---|---|---|---|---|---|---|---|---|---|
| **Preprocessing** | **Augmentation** | **Dataloading** | **Model** | **Output** | **Loss Calculation** | **Optimization** | **Metrics Evaluation** | **Validation** | **Visualization** |
| Clamp | Flip | Sampling | UNet | Sigmoid | Cross-Entropy | Adam | Dice | Cross Validation | ITKSnap |
| Rescale Intensity | Blur | Queuing | Transformer | Softmax | MSE | SGD | IoU | K-Fold Validation | Matplotlib |
| CropOrPad | Noise | Subjects | Yolo | Threshold | L1 Norm | AdamW | HD95 | Out-of-set | Wandb |

Knowing your I/Os

# I/Os

- I/O refers to **inputs** and **outputs**
- Understanding your pipeline's inputs and outputs is crucial for:
  - Model performance and architecture
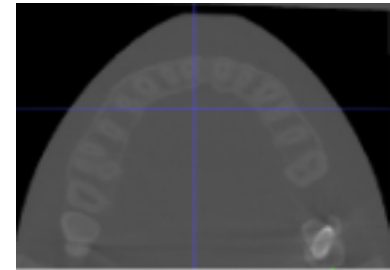  - Task accuracy
  - Evaluation decisions

# Input: Modality

- Modality refers to the type of data or sensory input used by a model or system
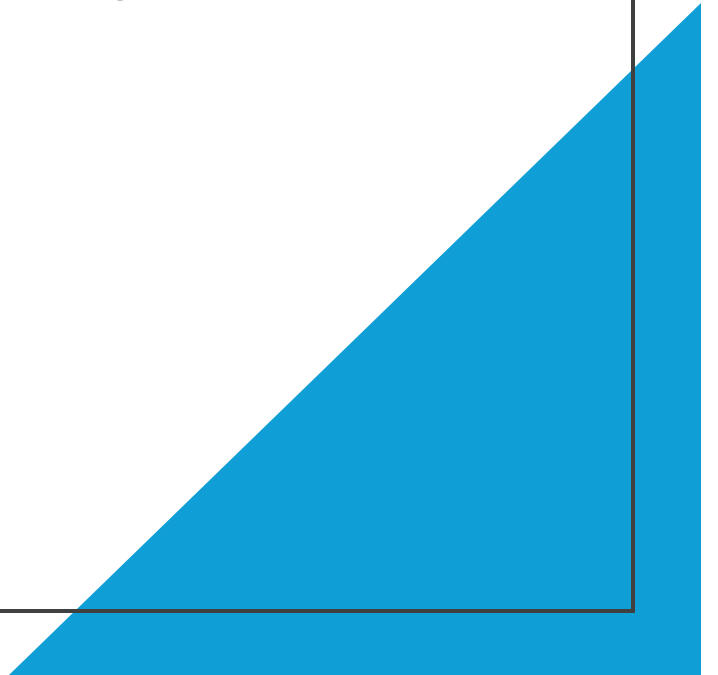
# Input: Modality

- Modality come in the form of
  - Images / Medical Images (CT, CBCT, Panorama)
  - Videos
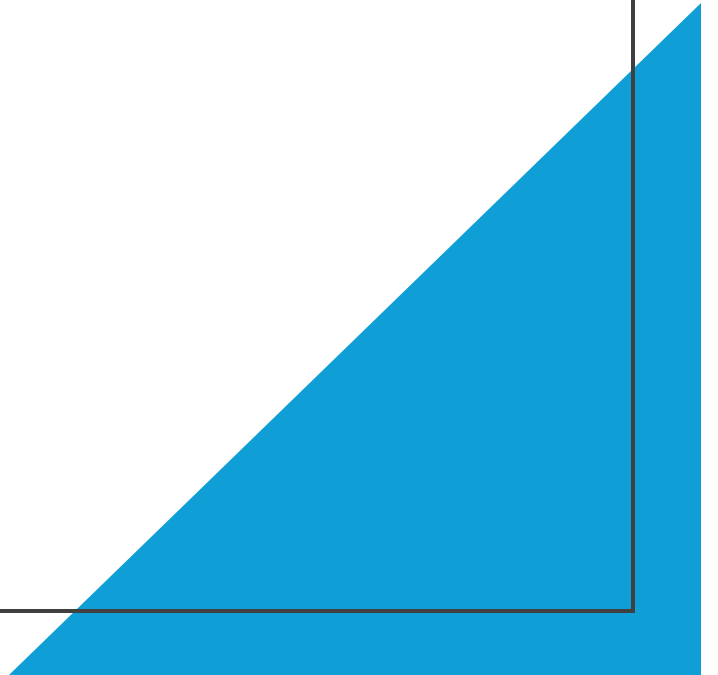  - Text (Reports)
  - Audio

# Input: Modality

- Models can be
    - Uni-Modal
        - Using only a single modality as Input
    - Multi-Modal
        - Using more than one modality of same form (ex: CBCT + Panorama)
        - Using more than one modality of different forms (ex: CBCT + Text Reports)

# Input: Dimensionality

- Dimensionality often corresponds to the spatial, spectral, and temporal dimensions of the data.

# Input: Dimensionality

**Spatial Dimensions:**

- Medical images typically have two or three spatial dimensions, represented as height, width (2D), and sometimes depth (3D) → CT, CBCT

**Spectral Dimensions:**

- Capture data across different wavelengths or functional responses → Functional MRI (fMRI) or multi-spectral CT

**Temporal Dimensions:**

- Temporal dimension is added, capturing changes over time → 4D CT or real-time ultrasound
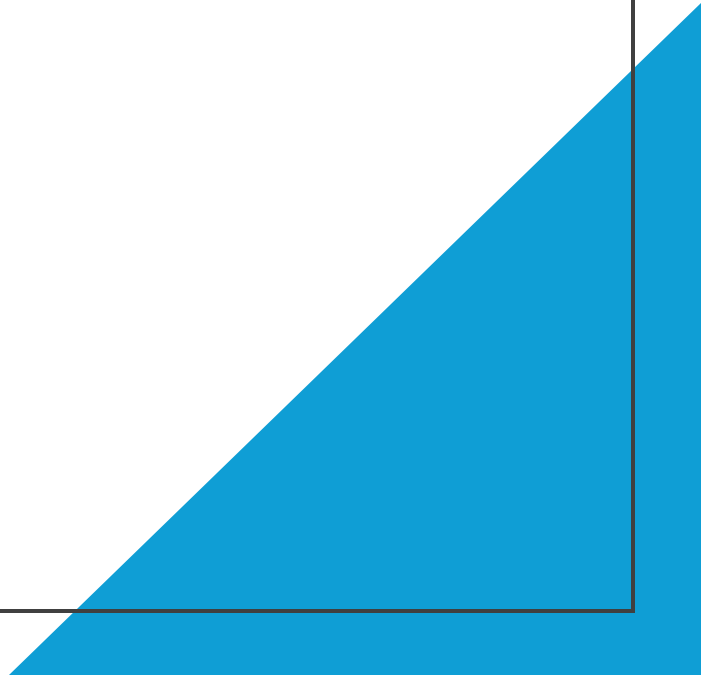
# Input: Dimensionality



- We are more interested in Spatial Dimensionality
  - 2D
    - Panoramic scans
    - Intraoral Images
  - 3D
    - CT / CBCT Volumes
  - Sliced 2D
    - CT / CBCT Planar slices across (Axial, Coronal, Saggital)
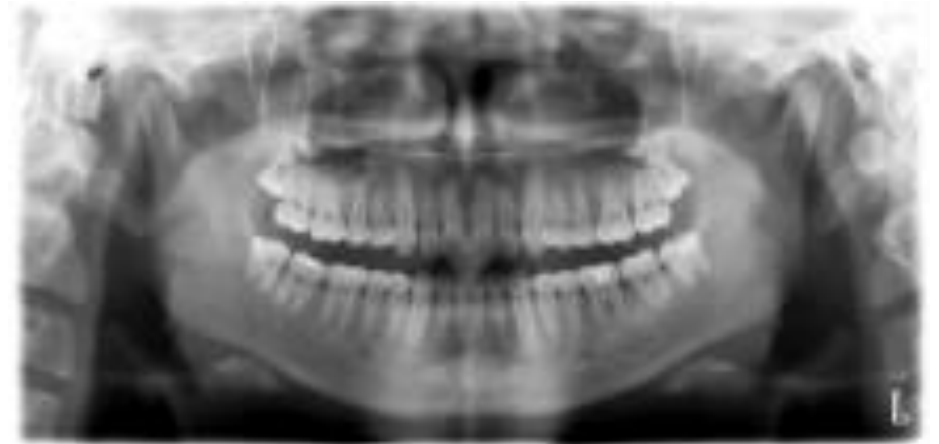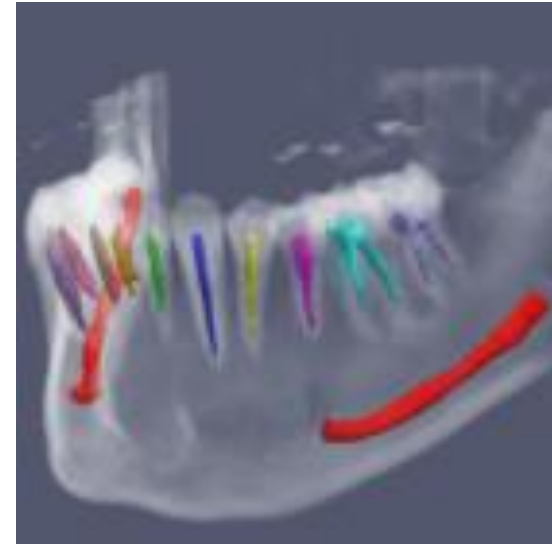
# Output: Task Definition

- The number of outputs constitues the task defintion

# Classification

- Involve categorizing medical images or specific regions within them into predefined classes or labels

- Examples
    - Caries Detection
    - Lesion Classification

- Task Definition
    - Single Output ➔ Yes or No
        - Does this image contian caries?
    - Multiple Outputs ➔ Mild, Moderate, Severe
        - What is the level of severity of caries in these images?
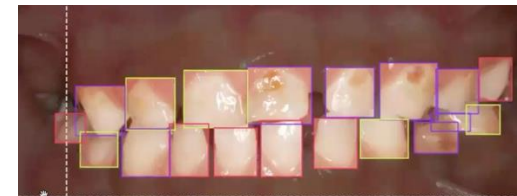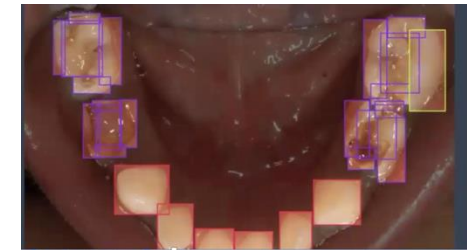
# Segmentation

- Involve partitioning an image into distinct regions or structures

- Examples:
  - Hard Tissue Segmentation
  - Pulp Instance Segmentation

- Task Definition
  - Single Output → Matches input size (Semantic)
    - Detect all lesions in the mouth
  - Multiple Outputs → Matches input size (Instance)
    - Detect all lesions in the mouth with their types respectively
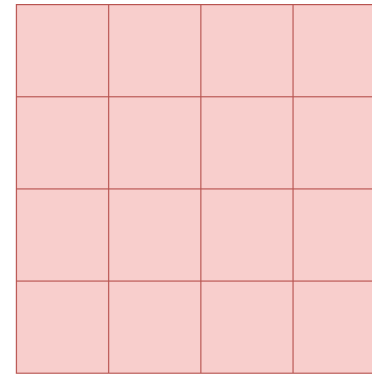
# I/O Representation

- I/O are represented as **tensors**
- Tensors can have varying **ranks** (or dimensions)
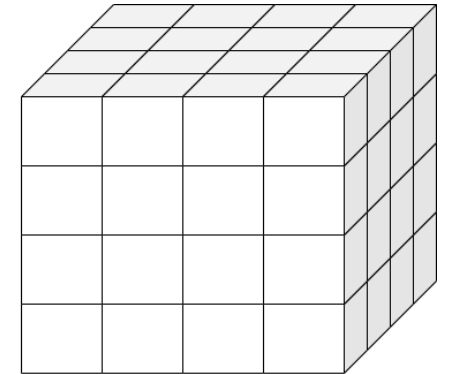  - each rank represents a level of data complexity



Scalar

Vector
1D Tensor
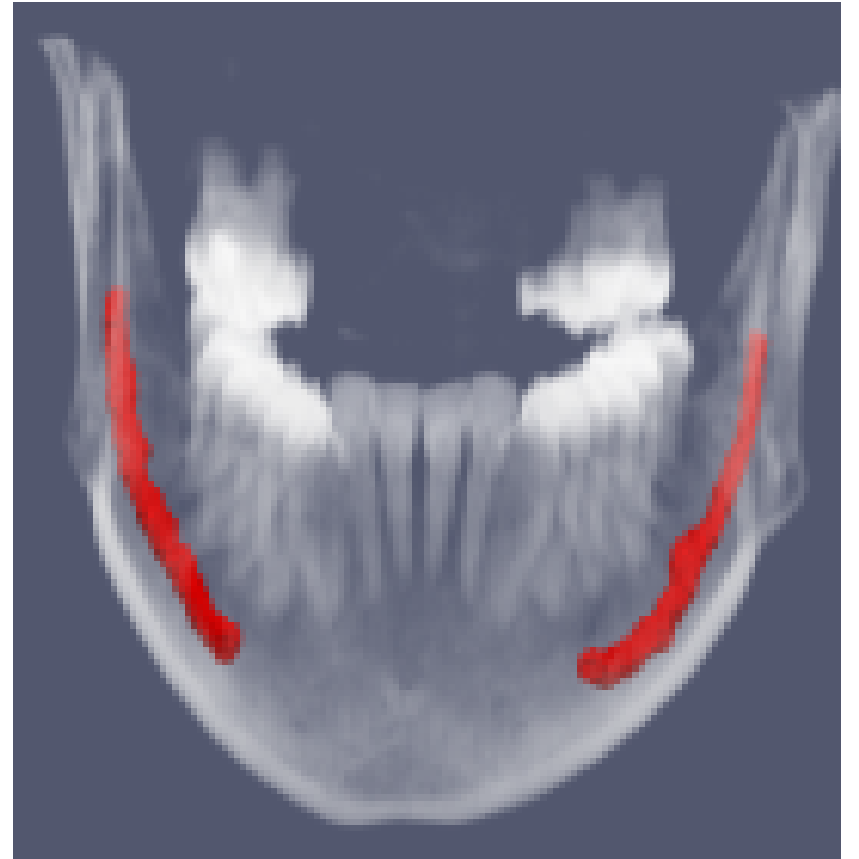
Matrix
2D Tensor

Volume
3D Tensor

# I/O Representation

- **Higher-Rank Tensors** (Rank 3 or more):
  - 3D tensor for a color image
    - (height × width × color channels)
  - 4D tensor for a batch of 3D volumes
    - (batch size × height × width × depth)
  - 5D tensor for a batch of 3D volumes each predicting a class label
    - ( # of classes x batch size × height × width × depth)
    - That's what we used to get this →
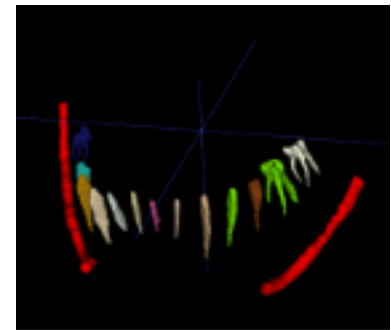
# Use Case: IAN Segmentation

- Through this workshop we will dive into segmenting the IAN

  - First, perform some data exploration

  - Second, create a pipeline

  - Third, train and evaluate the model

- What to expect

  - A pipeline ready to be used for most common deeplearning tasks

# Questions?

# Workshop Activity

- Notebooks Link
  - https://github.com/KnightsLab/EMRA-Workshop