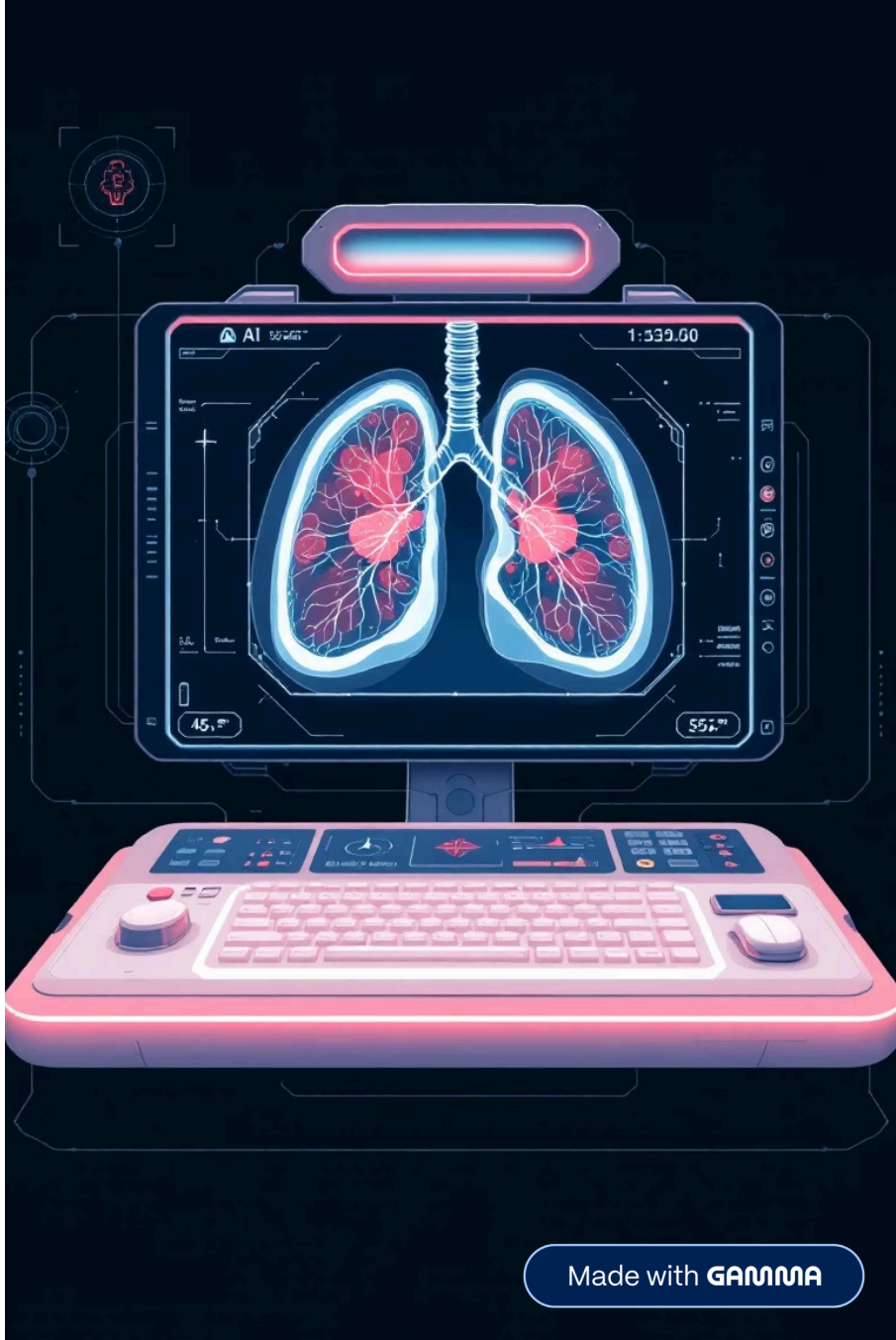
Detective StarLung: AI for Early Cancer Detection

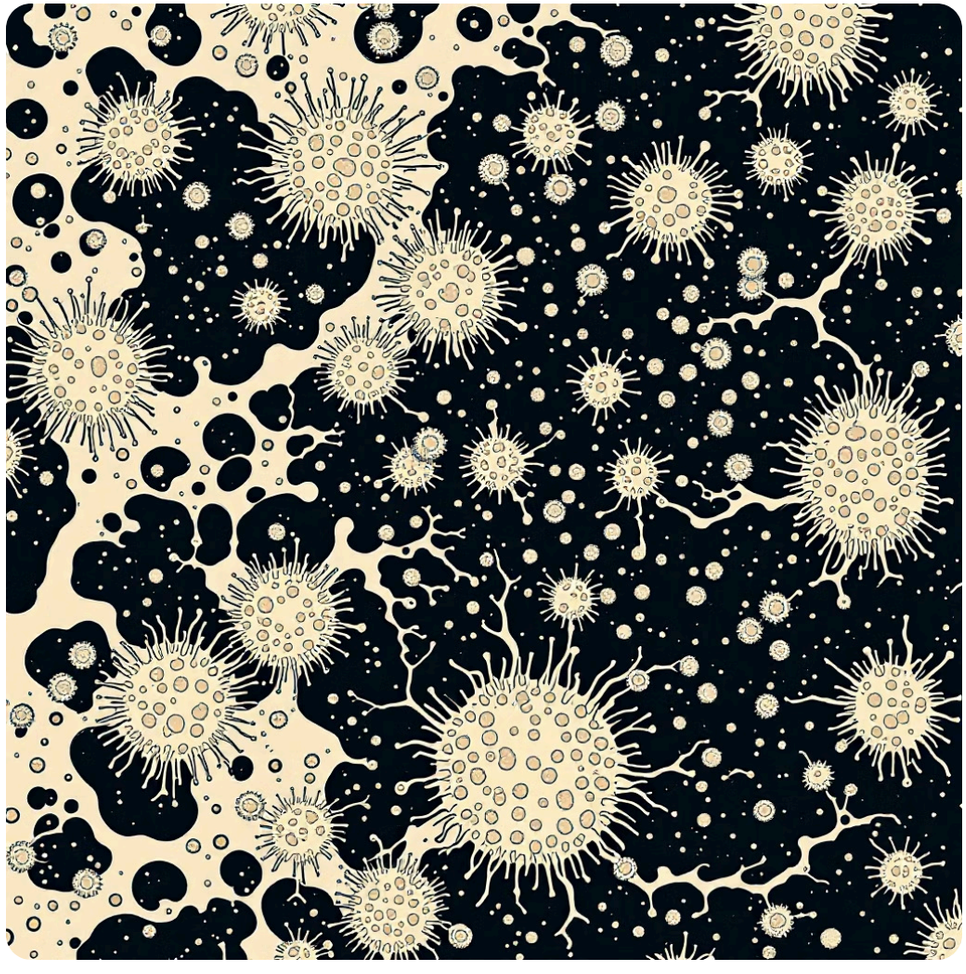
This project aims to develop an AI model, Detective StarLung, for automated classification of lung cancer types from CT scans, assisting in medical diagnostics.



Data Source & Real-World Impact

Data Source & Purpose

Using a dataset of .jpg lung CT scans to train a Convolutional Neural Network (CNN) for automated cancer type classification.



Applications

Early & Quick Analysis

Rapid identification for timely treatment.

Second Opinion System

Reduces human error and fatigue in diagnostics.

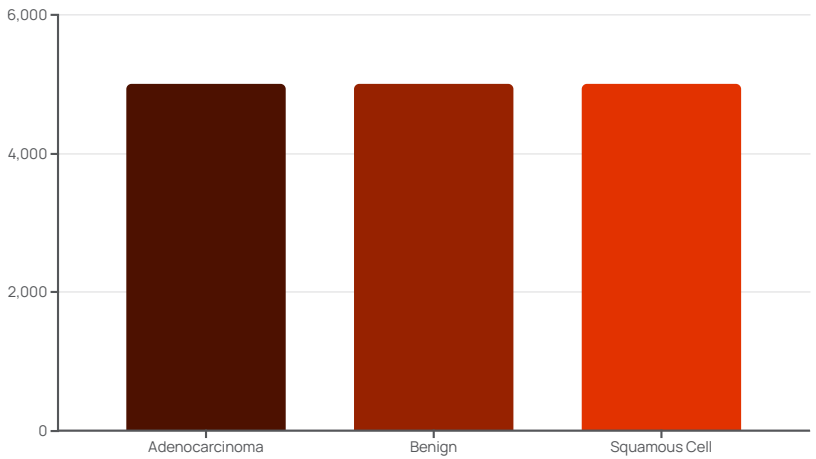
Remote Diagnostics

Enables access to specialized care globally.

Dataset Overview & Exploration

- Globals: Batch size = 32, Image resolution = 256x256 pixels.
- Libraries used: **os** (file system interaction), **numpy** (numerical operations), **matplotlib.pyplot** (data visualization), **tensorflow** (machine learning framework).

The dataset contains 15,000 images, with 5,000 images for each of the three classes: adenocarcinoma, benign, and squamous cell carcinoma, ensuring a balanced dataset.



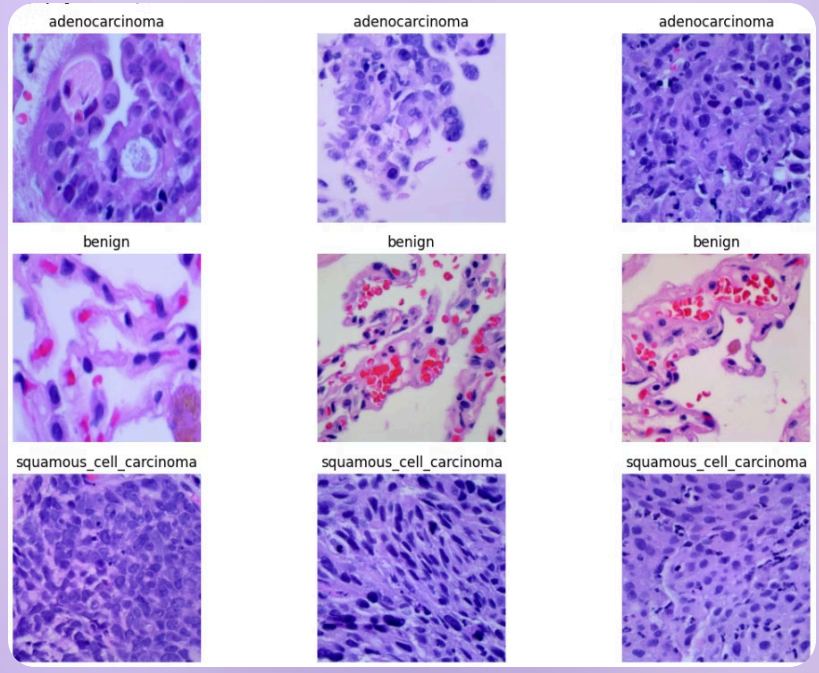
Balanced dataset: 5,000 images per class.

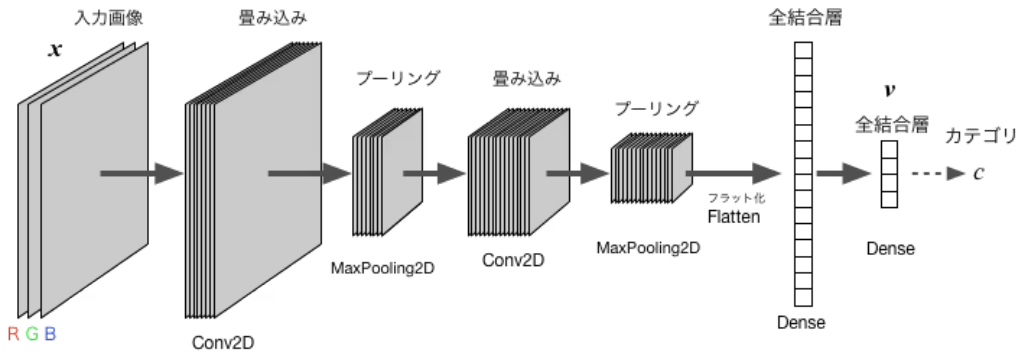
Visualizing Lung CT Scan Samples

Our `plot_samples` function randomly displays 3 images from each class, ensuring visual representation across the dataset.

This function utilizes nested loops to iterate through each class and select random images for display.

```
•[20]: def plot_samples(data_dir, classes, samples_per_class=3):
           plt.figure(figsize=(12, 8))

           for i, class_name in enumerate(classes):
               class_path = os.path.join(data_dir, class_name)
               images = np.random.choice(os.listdir(class_path), samples_per_class, replace=False)

               for j, img_name in enumerate(images):
                   img_path = os.path.join(class_path, img_name)
                   img = plt.imread(img_path)

                   plt.subplot(len(classes), samples_per_class, i * samples_per_class + j + 1)
                   plt.imshow(img, cmap='gray')
                   plt.title(class_name)
                   plt.axis('off')

           plt.tight_layout()
           plt.show()

       print("Displaying random samples from each class...")
       plot_samples(DATA_DIR, CLASSES)
```





Data Preparation & Initial Model Architecture

V1.0 Data Strategy

- **No Geometric Transformations:** Baseline established without rotation, zoom, or shear.
- **Rescaling:** Pixel values normalized from 0-255 to 0-1 for faster model convergence.
- **Validation Split:** 20% of data reserved to monitor for overfitting during training.

Draft CNN Architecture

01

Conv2D Layers

Detects edges, textures, and shapes within images.

02

MaxPooling2D

Reduces spatial dimensions, lowering computational load.

03

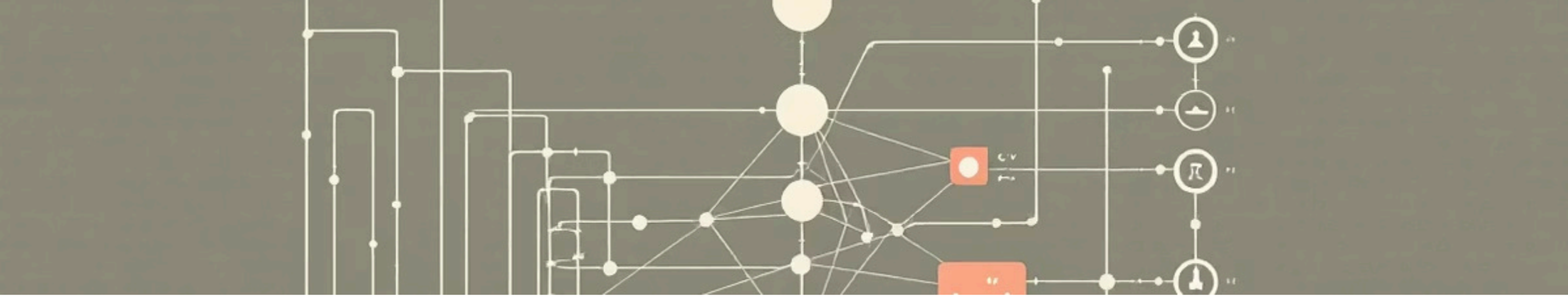
Flatten

Converts 2D feature maps into a 1D vector.

04

Dense Layers

Fully connected layers for final classification decisions.



Training Attempts & Model Compilation

We conducted two training attempts for our v1.0 model to assess baseline performance.

1

2

Attempt 1

- Sequential CNN: 3x Conv2D + MaxPooling2D layers.
- 10 Epochs.

Attempt 2

- Same architecture as Attempt 1.
- 5 Epochs (halved), and halved Image input resolution.

Model Compilation

Optimizer: Adam (efficient for sparse gradients, adaptive learning rates).

Loss: Categorical Crossentropy (suitable for multi-class classification).

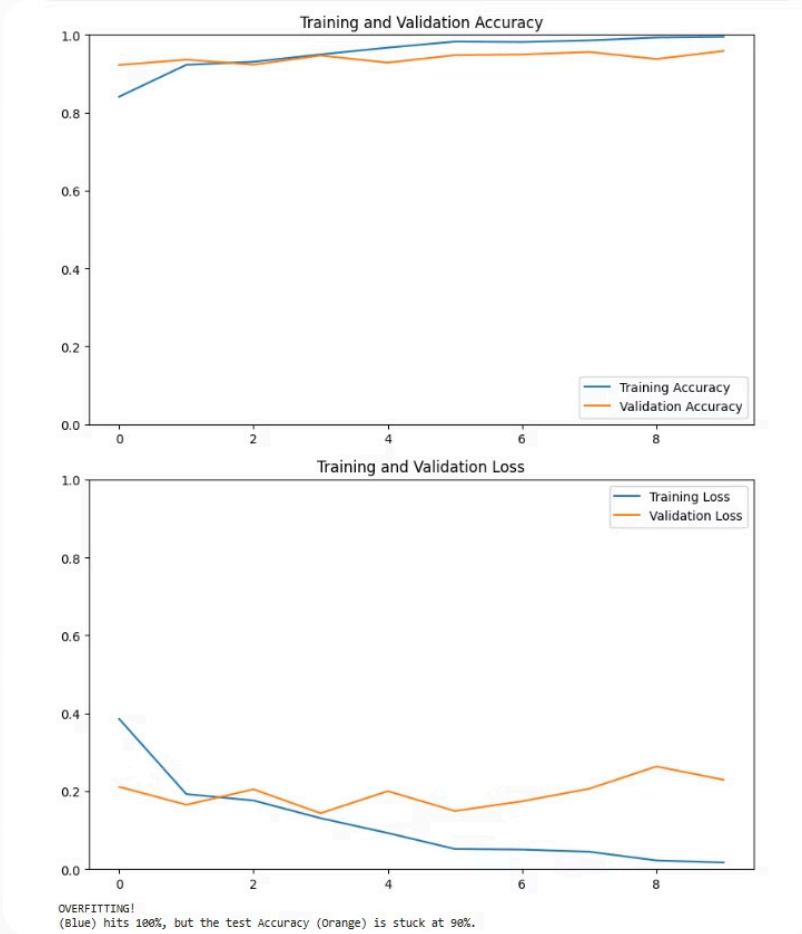
Metrics: Accuracy (primary performance indicator).

Attempt 1: Performance Output Analysis

Understanding the training and validation curves is crucial for diagnosing model behavior.

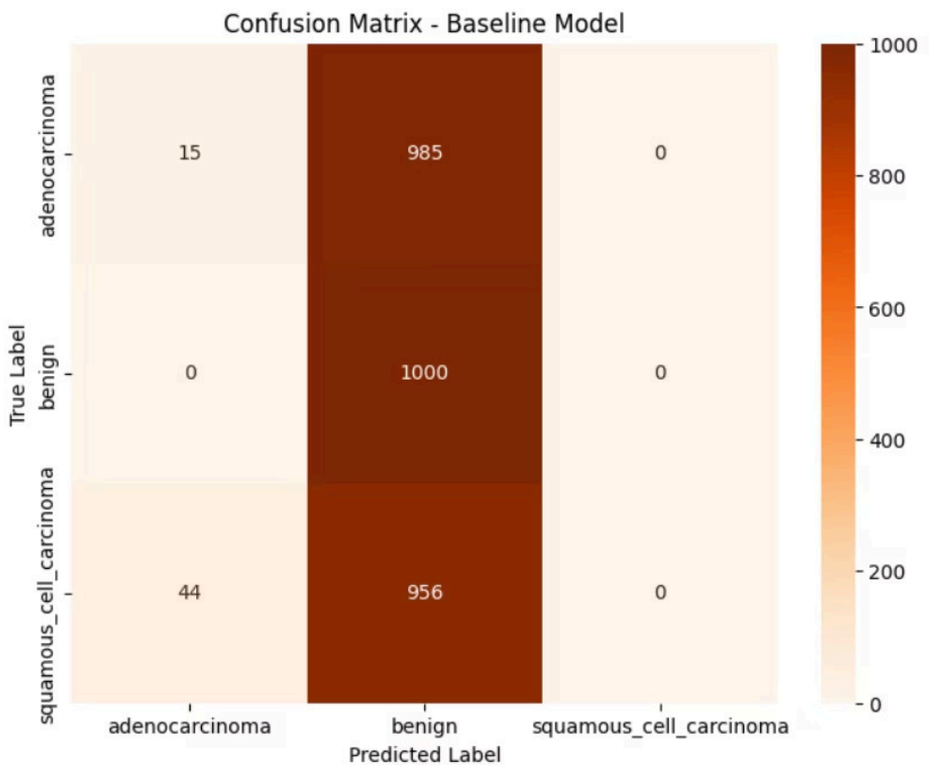
- **The Gap:** A large difference between high training accuracy (nearly 100%) and low validation accuracy (keeps at 90%) indicates overfitting.
- **The Divergence:** When training loss decreases but validation loss starts to increase, the model is learning noise, not generalizable features.

Version 1 (Baseline) Result: We observed severe overfitting and model collapse, as shown by the divergent curves.



Attempt 1: Confusion Matrix & Misclassification

The confusion matrix reveals a critical flaw in our baseline model's predictions.



The model predicts 'Benign' for almost everything in the validation set.

The model almost exclusively predicts "Benign" for all validation samples.

0.15%

Adenocarcinoma

Misclassified as Benign.

0.0%

Squamous

Misclassified as Benign.

100%

Benign

Correctly classified.

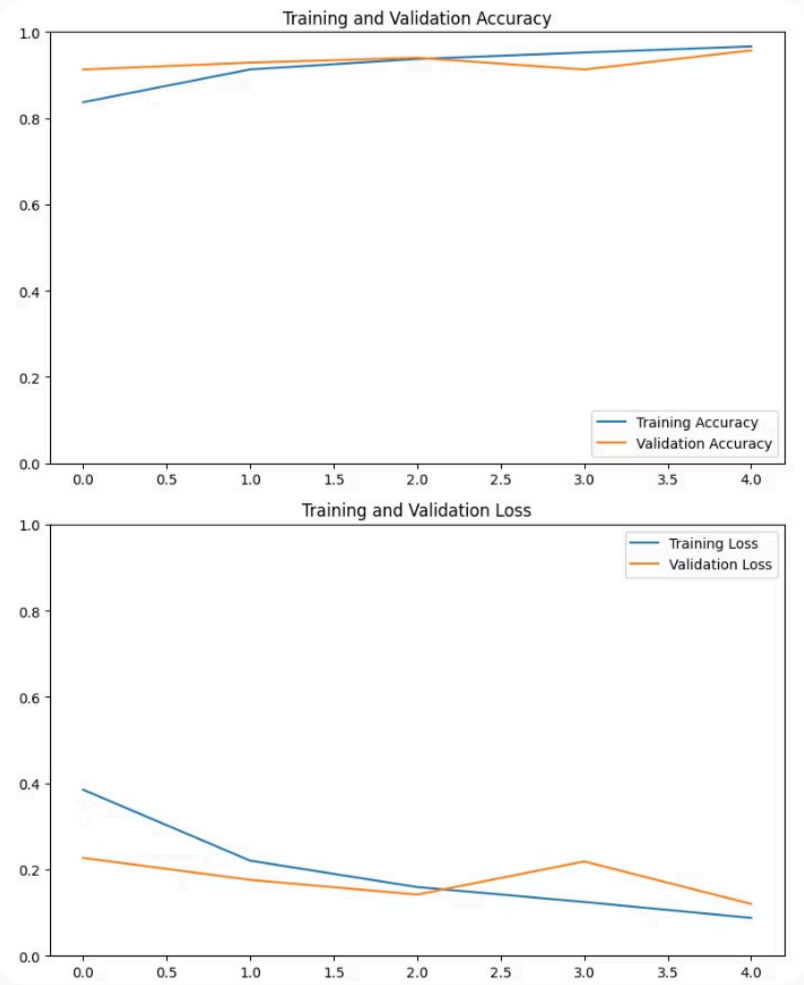
This indicates the model has failed to learn distinguishing features for adenocarcinoma and squamous cell carcinoma.

Attempt 2: Performance Output Analysis

Understanding the training and validation curves is crucial for diagnosing model behavior.

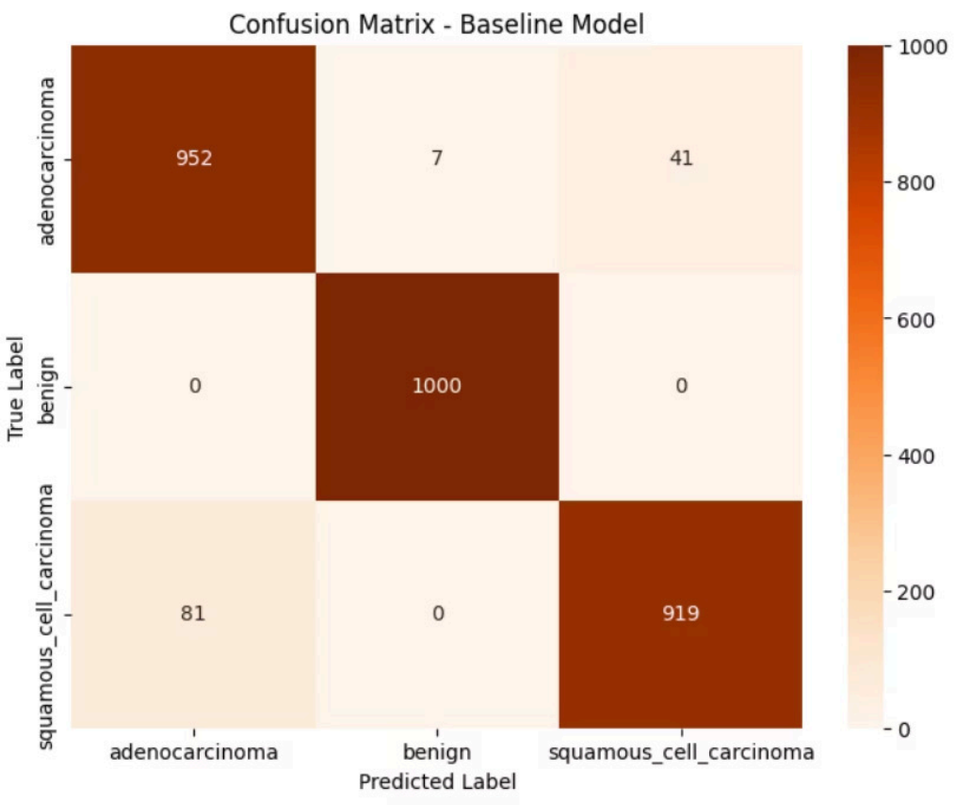
- **The Gap:** The lower difference between high training accuracy (nearly 100%) and low validation accuracy (keeps at 90%) deleted overfitting.
- The accuracy started with 90% increased to nearly 100%, shows…

Version 1 (Baseline) Result: Based on the visual evidence from the curves, we observed **no significant overfitting** and **no model collapse**.



Attempt 1: Confusion Matrix & Misclassification

The confusion matrix reveals a critical flaw in our baseline model's predictions.



This time, the model predicts more accurately for all validation samples.

95.2%

Adenocarcinoma

Correctly classified

91.9%

Squamous cell carcinoma

Correctly classified

100%

Benign

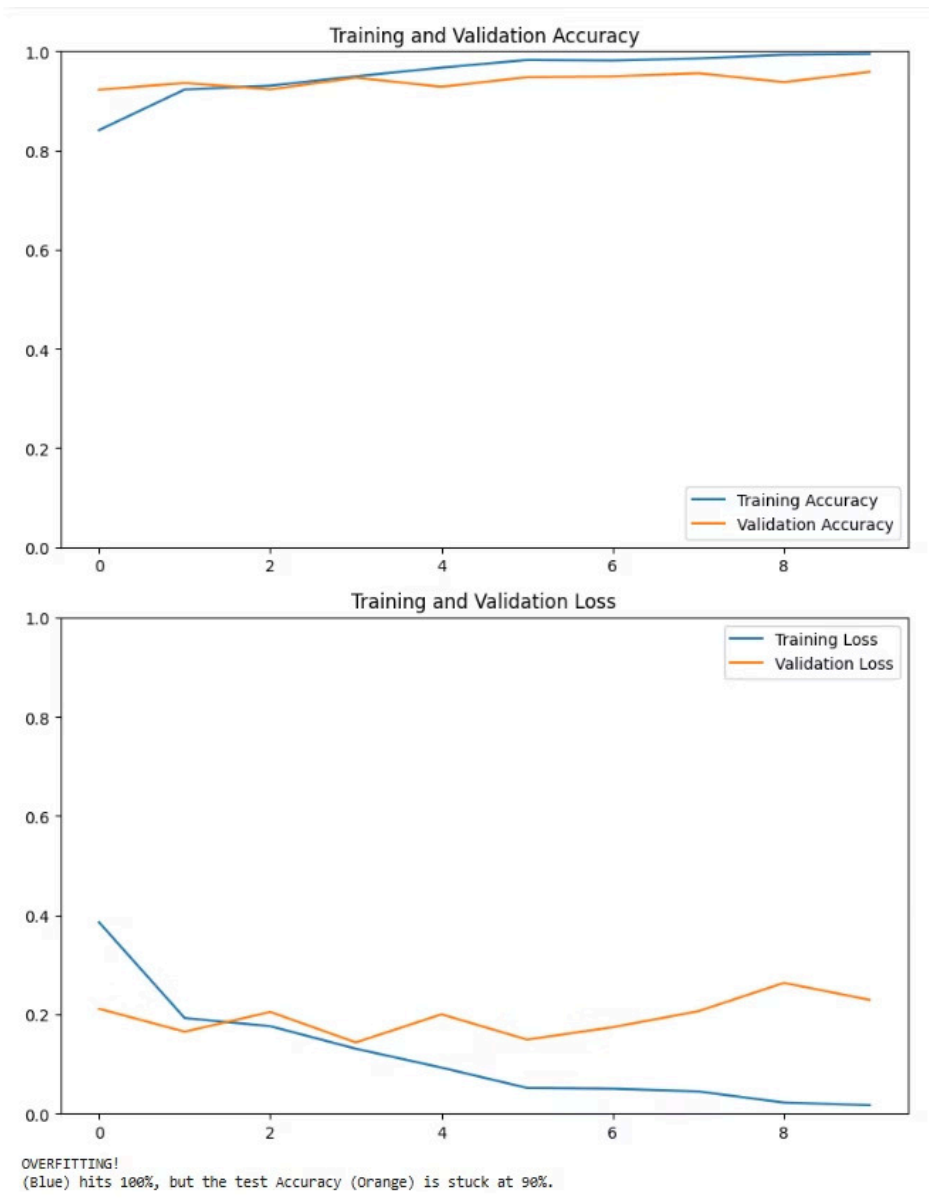
Correctly classified.

This indicates the model has failed to learn distinguishing features for adenocarcinoma and squamous cell carcinoma.

Comparing Attempts: Performance & Misclassification

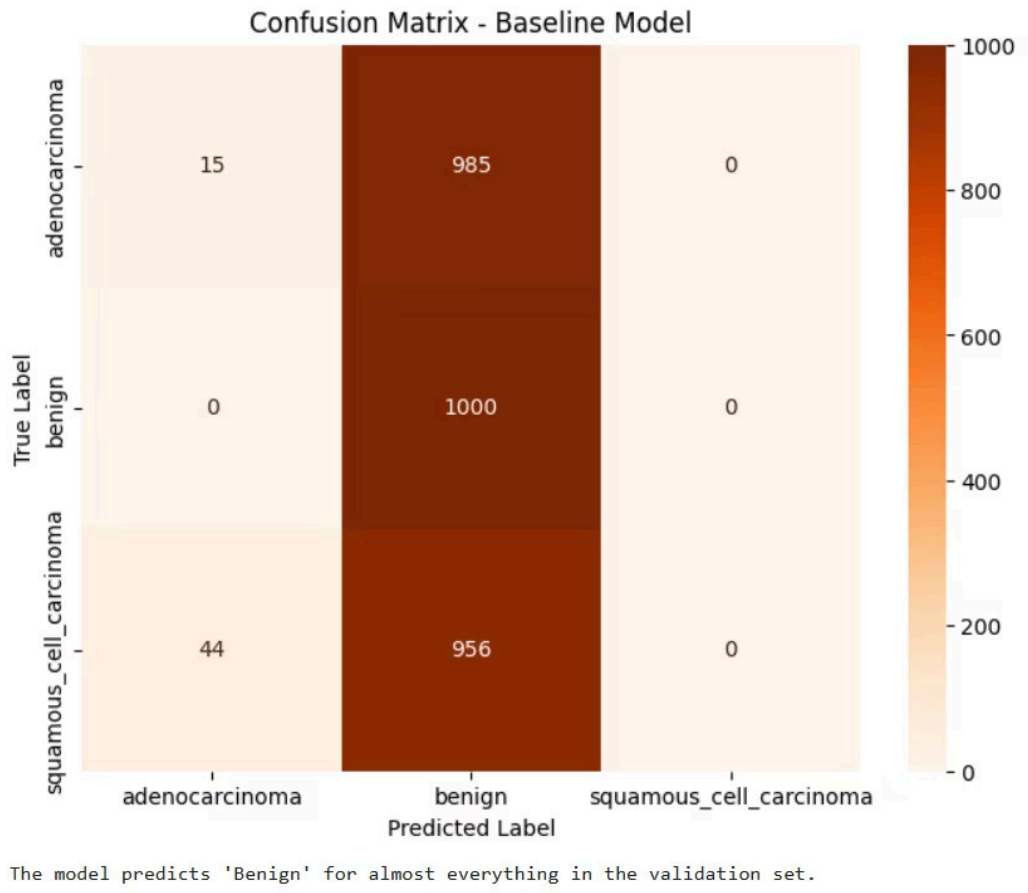
A side-by-side analysis of Attempt 1 and Attempt 2 reveals critical differences in model performance and prediction patterns.

Attempt 1: Overfitting & Collapse



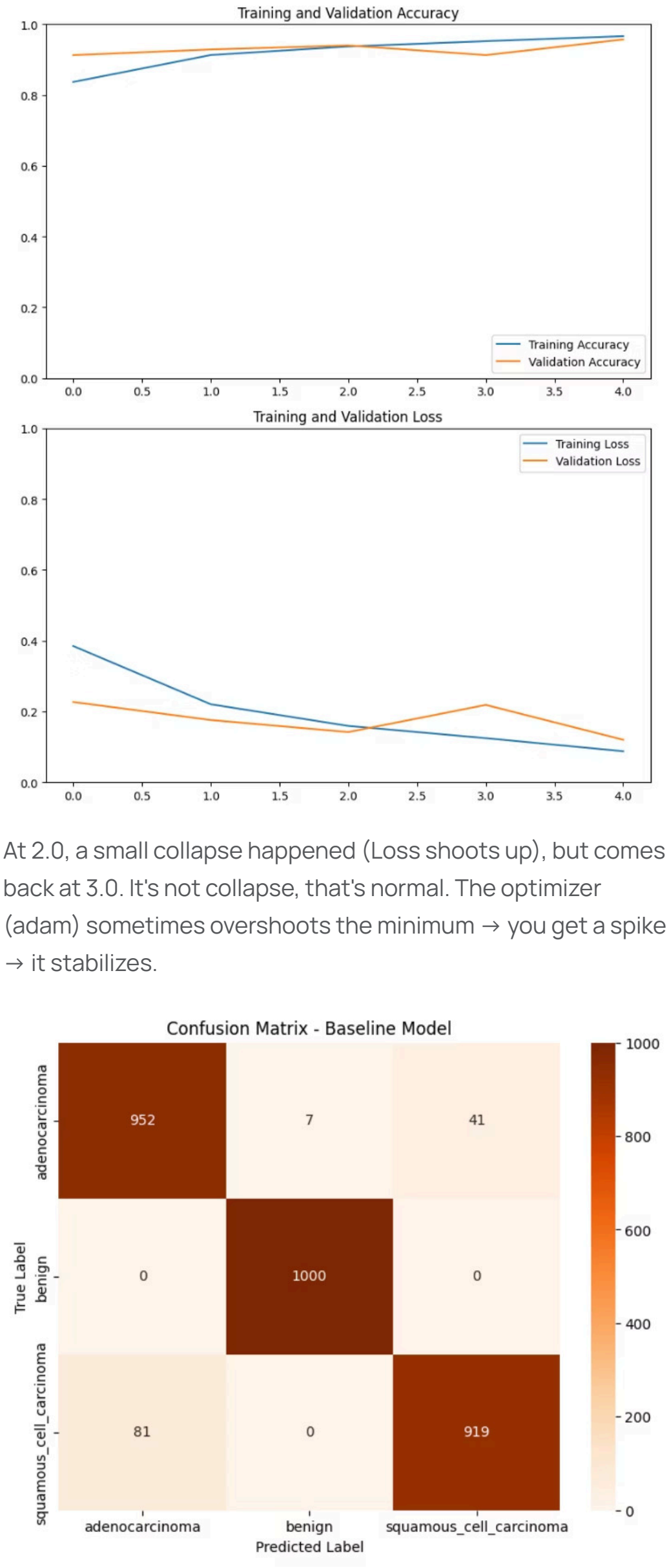
The performance curves show extreme divergence, indicating severe overfitting and model collapse.

Notice validation accuracy and validation loss are not following the Training trend.



The confusion matrix confirms that the model almost exclusively predicted "Benign" for all classes.

Attempt 2: No Overfitting



At 2.0, a small collapse happened (Loss shoots up), but comes back at 3.0. It's not collapse, that's normal. The optimizer (adam) sometimes overshoots the minimum → you get a spike → it stabilizes.

The confusion matrix indicates improved classification for Adenocarcinoma and Squamous, but not perfect.

Quantitative Comparison of Classification Performance

Attempt 1

0.15%
Adenocarcinoma Correct

0.0%
Squamous Cell Correct

100%
Benign Correct

Attempt 2

95.2%
Adenocarcinoma Correct

91.9%
Squamous Cell Correct

100%
Benign Correct

Attempt 2 significantly improved classification for non-benign cases, confirming that model adjustments can mitigate severe biases.

The Better Solution: Introducing Data Augmentation

> "This failure proves that we must introduce Data Augmentation (Rotation, Zoom, Flips) to force the model to learn the shape and texture of the tumor, regardless of its orientation."

Next Steps: Model v1.1

Goal

Implement robust data augmentation techniques.

Expectation

Training accuracy may drop (90-95%), but validation accuracy should rise (90-100%), closing the performance gap.