

TUBERCULOSIS DETECTION

using Chest X-ray

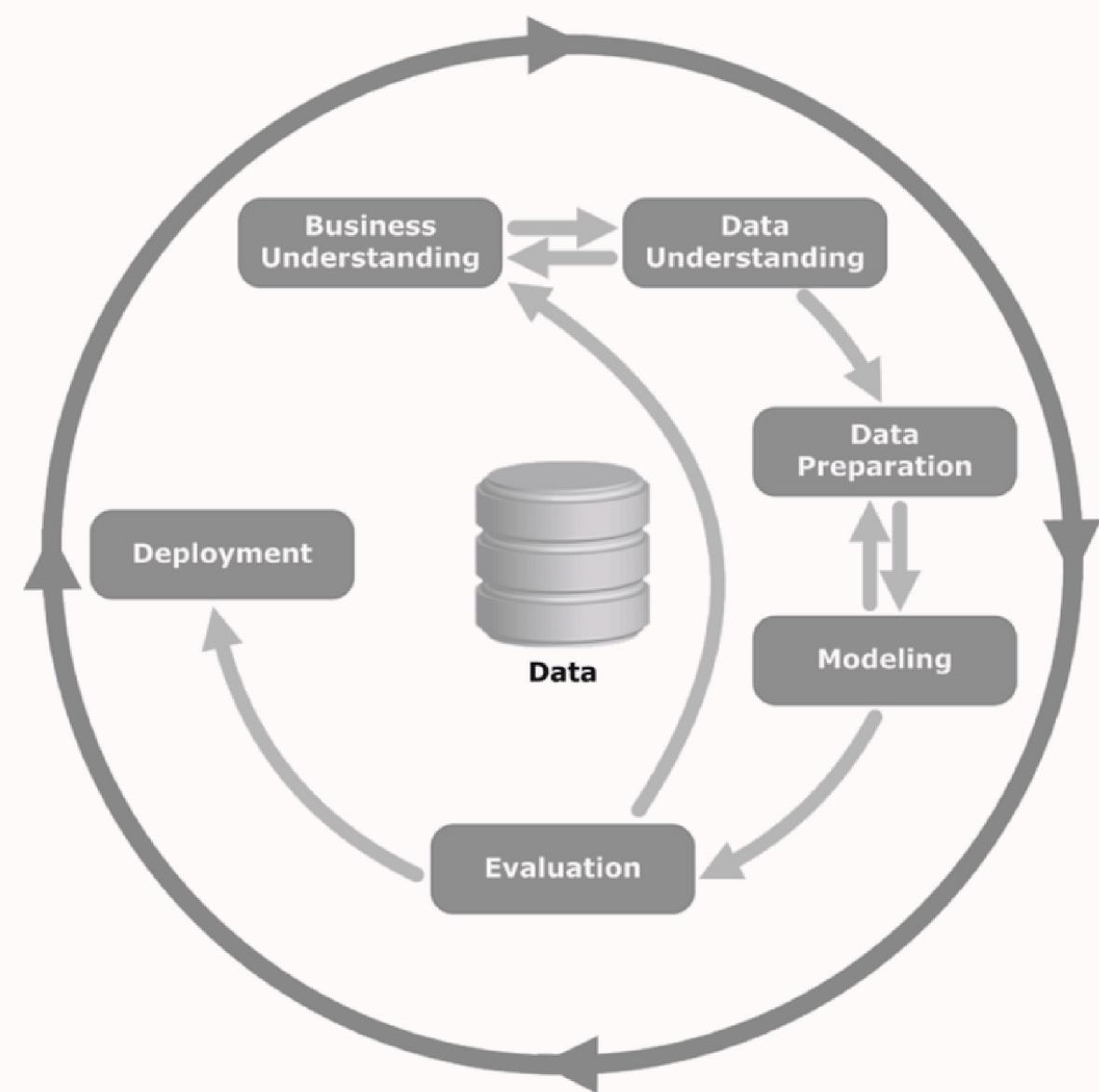


Pisit Janthawee (Data Scientist)
Bachelor of Science in Computer Science,
Bangkok University

METHODOLOGY

CRISP-DM

(Cross-Industry Standard Process for Data Mining)



BUSINESS UNDERSTANDING



BUSINESS TYPE

Healthcare, Medical



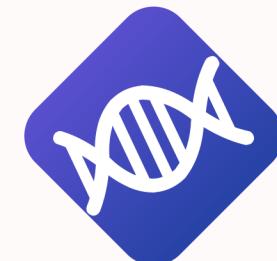
PROBLEM?

Hard diagnose

Tuberculosis can present with a wide range of symptoms such as Pneumonia, Lung Cancer, Bronchitis, etc. making it difficult to accurately diagnose.

Key facts

Tuberculosis (TB) remains a global health challenge, with 1.6 million TB-related deaths in 2021. It is the **second** leading infectious killer worldwide, following COVID-19.



GOAL

Develop an accurate and efficient deep learning model for the automated detection and classification of Tuberculosis (TB) in chest X-ray images. The goal is to assist healthcare, thereby **improving patient outcomes** and **reducing the spread of the disease**.

DATA UNDERSTANDING

01

DATA

- **Amount:** 700 Tuberculosis images / 3500 normal images.
- **File type:** PNG
- **Image size:** 512x512
- **Channel:** 3 channel (RGB)
- **Target:** Disease = 1, No disease = 0

02

ANALYSIS

- **Contrast of X-ray Image:** Which Chest X-ray Image Processing/Enchantment gives better performances?
- **X-ray Image Input Sizes:** Exploring Different Image Input Sizes What input sizes give the best performance and avoid the need for unnecessarily larger inputs?

03

TARGET

Classes

Disease	No disease
1	0



DATA PREPARATION

DATA TRANSFORMING

- **Tabulating:** PNG Files to NumPy Arrays
- **Color Conversion:** RGB to Grayscale
- **Image Resizing:** 512x512 to 128x128

DATA AUGMENTATION + CLINICAL PERSPECTIVE

- Solve class Imbalance
- 700 Tuberculosis images / 3500 normal images.

VARIABLE



DATA PROCESSING

IMAGE PROCESSING/ENCHANTMENT

- Contrast Stretching
- Histogram Equalization
- Adaptive Histogram Equalization

NORMALIZATION

- Min-Max Normalization

READY TO TRAIN!

MODELING

Experimentation
Hyperparameter Tuning
Evaluation

EXPERIMENTATION

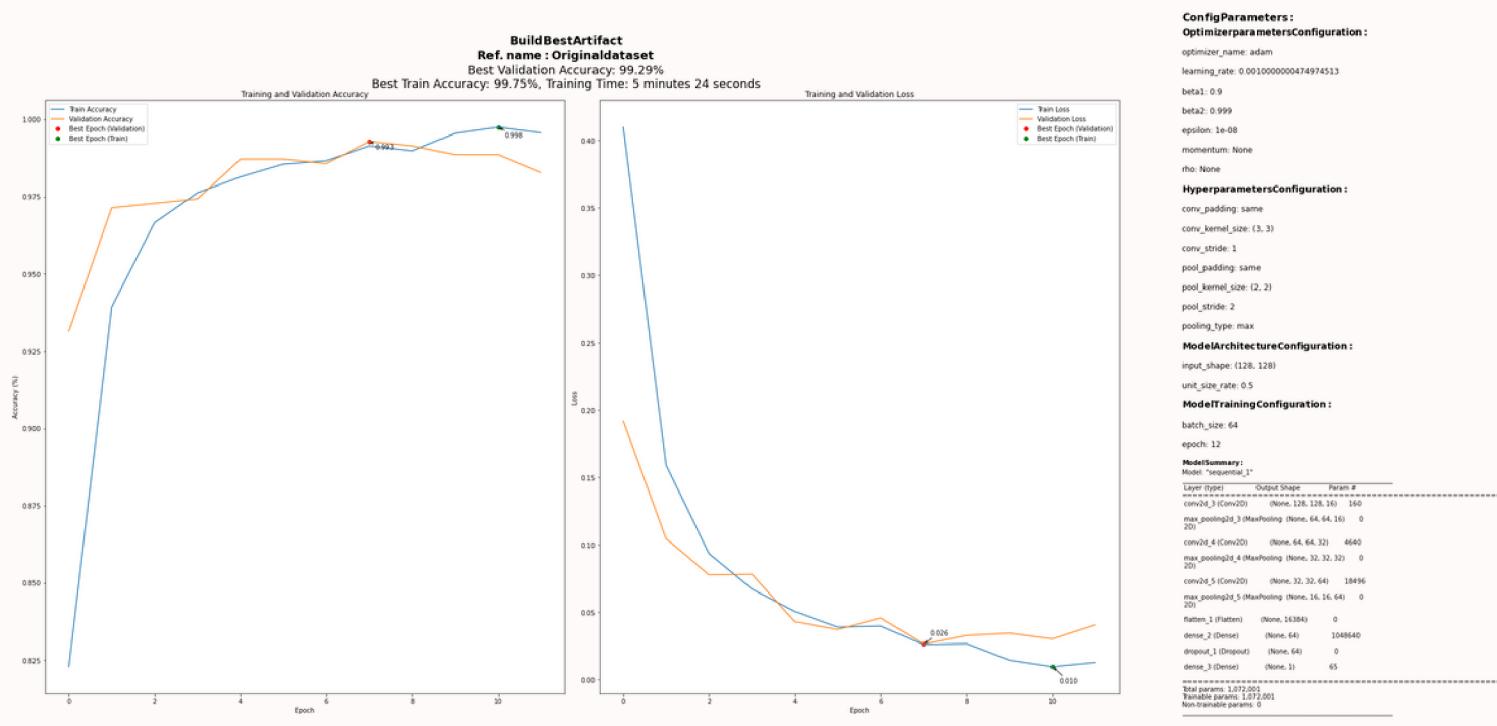
Training and Evaluation and Experiment

- **Experiment 1:** Building the First CNN Model
- **Experiment 2:** Comparing Optimizers
- **Experiment 3:** Comparing Pooling Techniques
- **Experiment 4:** Tuning Hyperparameters - Padding, Kernel Size, and Stride
- **Experiment 5:** Hyperparameter Optimization
 - **Experiment 5.1:** Learning Rate (λ) for Adam
 - **Experiment 5.2:** Epsilon (ε) for Adam
 - **Experiment 5.3:** Batch Size
 - **Experiment 5.4:** Number of Epochs
- **Experiment 6:** Image Enhancement Techniques
- **Experiment 7:** Impact of Varying Unit Sizes in Neural Network Layers
- **Experiment 8:** Exploring Different Image Input Sizes

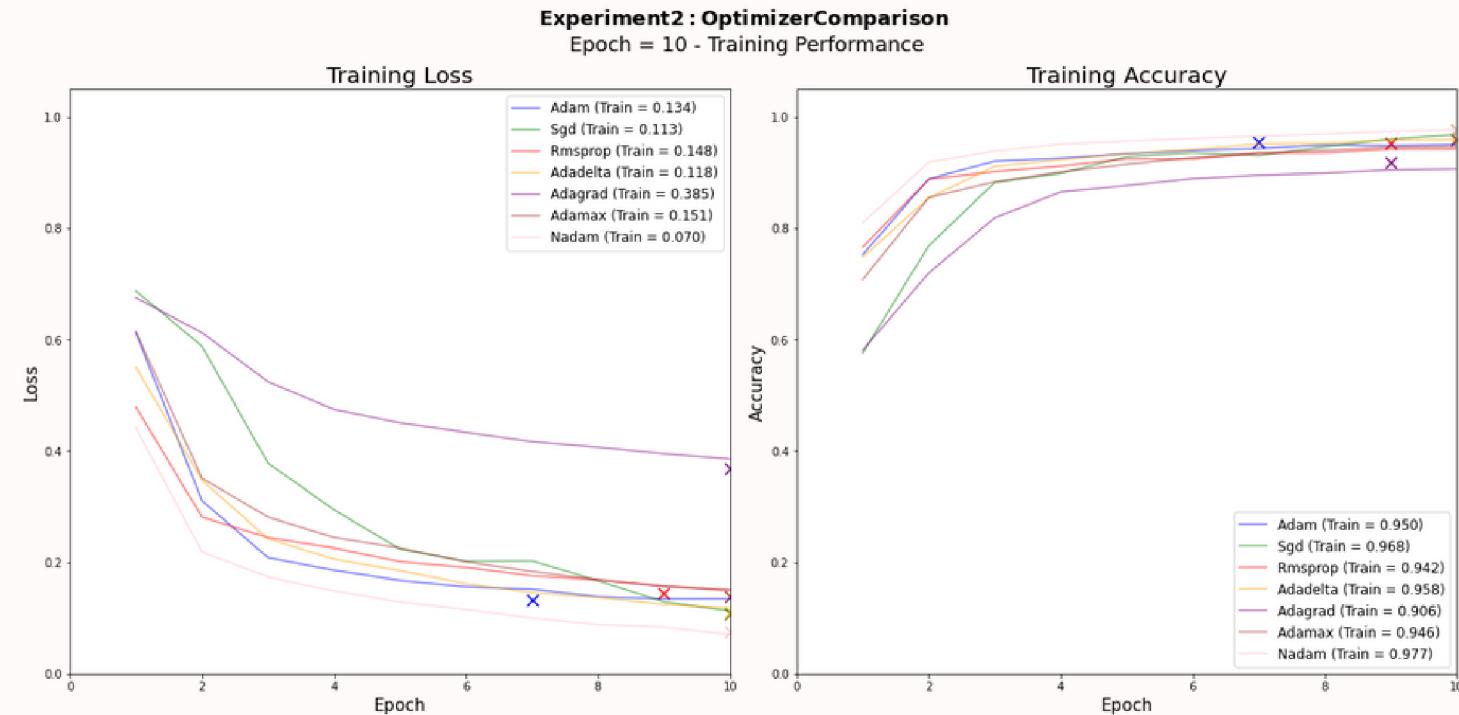
EXPERIMENTATION

Training and Evaluation and Experiment

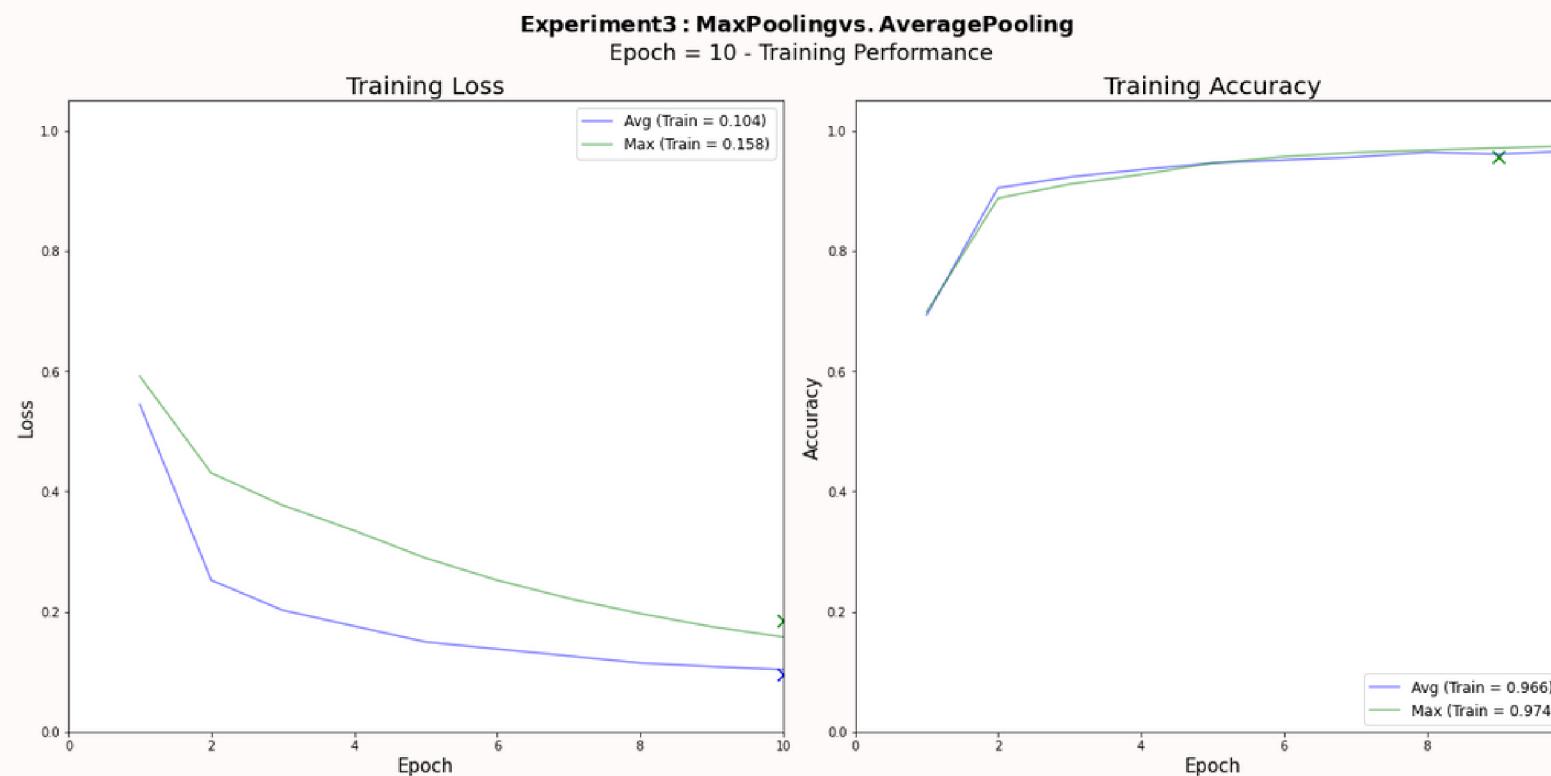
- **Experiment 1:** Building the First CNN Model



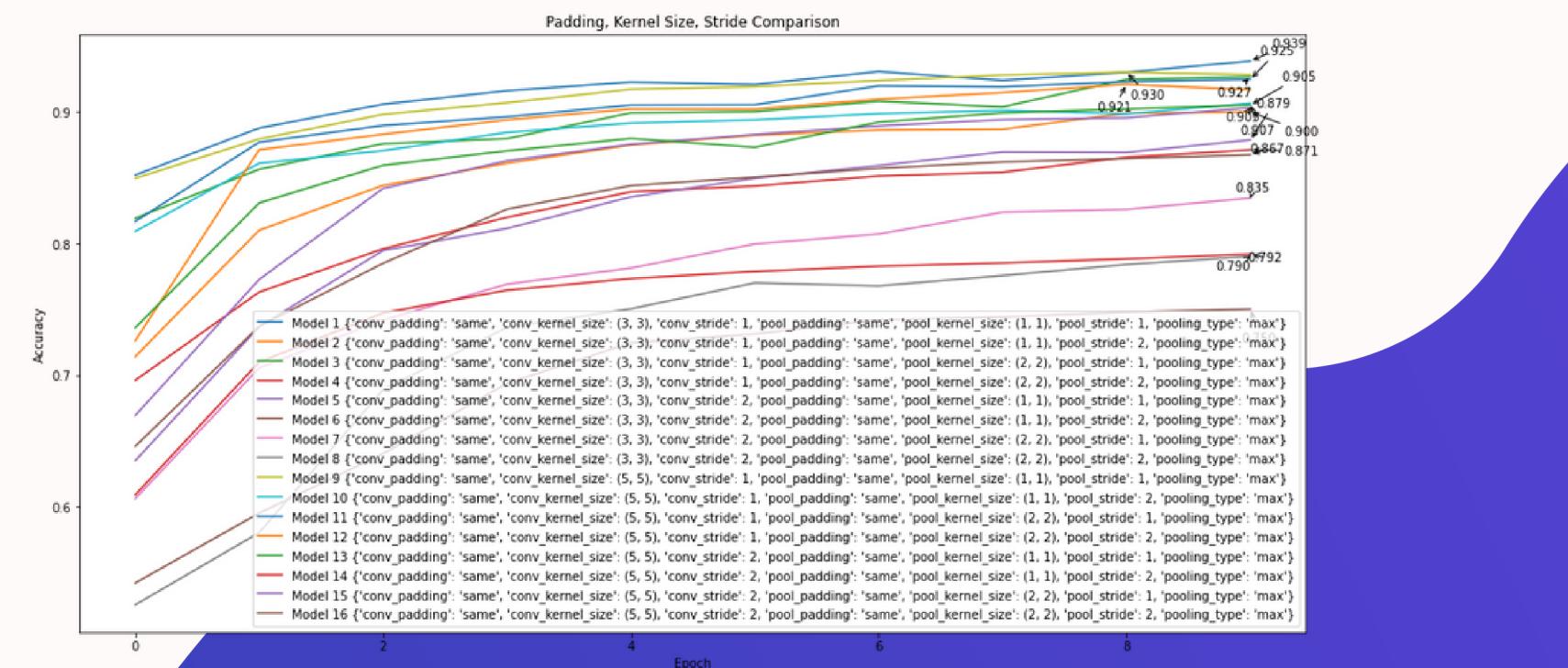
- **Experiment 2:** Comparing Optimizers



- **Experiment 3:** Comparing Pooling Techniques



- **Experiment 4:** Tuning Hyperparameters - Padding, Kernel Size, and Stride

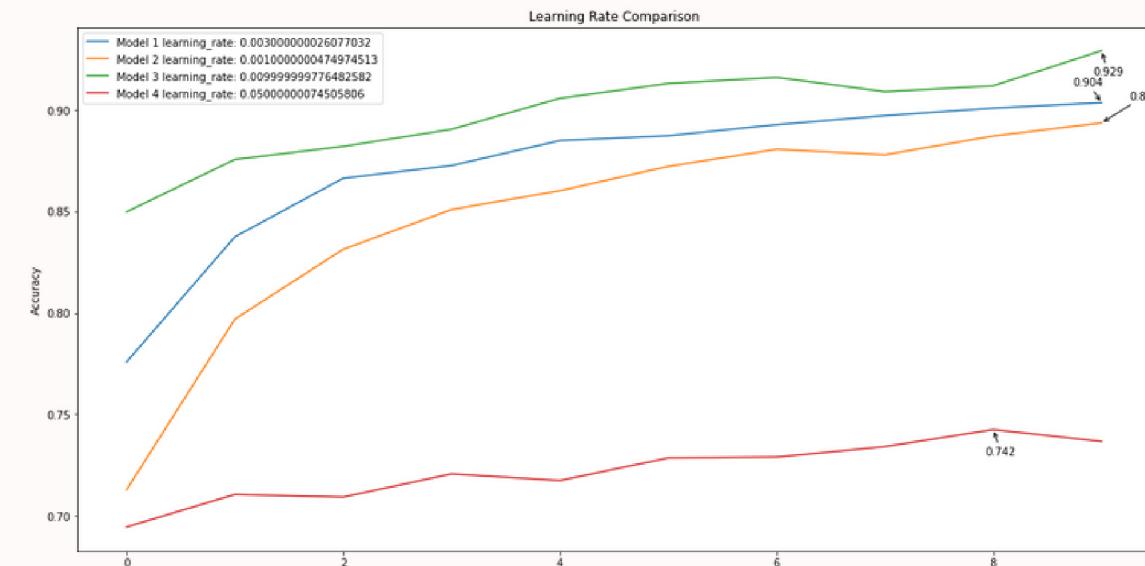


EXPERIMENTATION

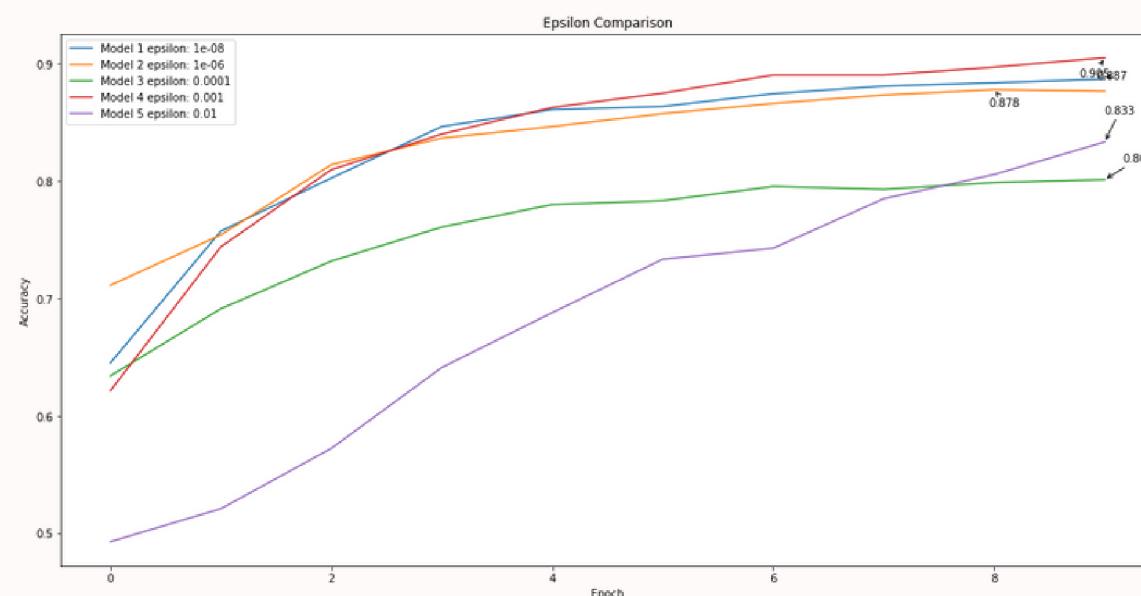
Training and Evaluation and Experiment

Experiment 5: Hyperparameter Optimization

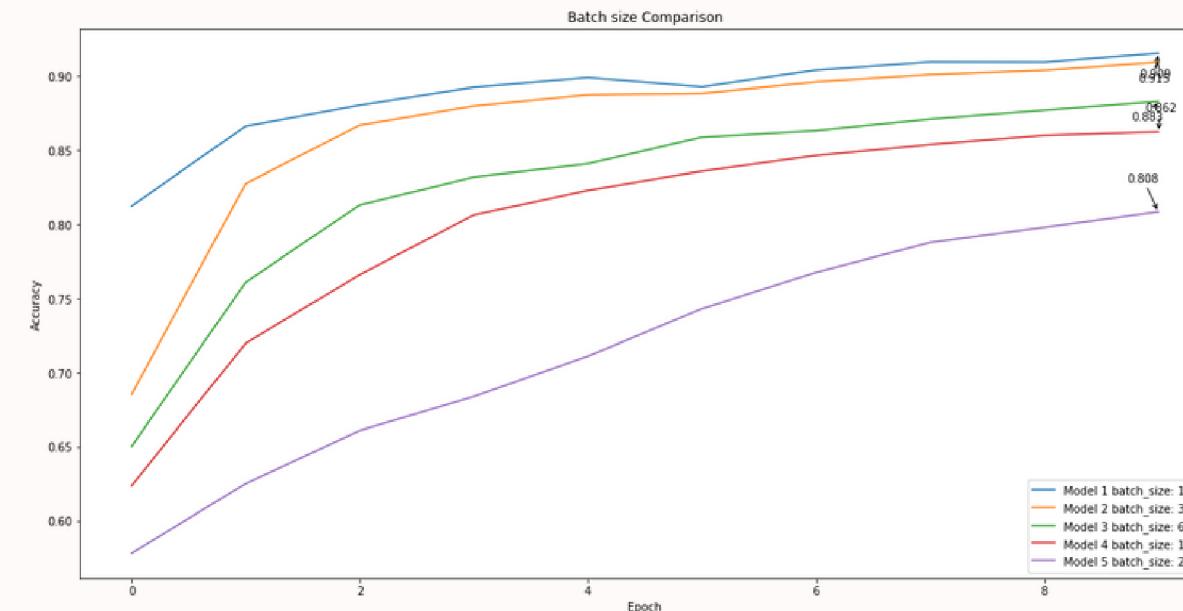
- **Experiment 5.1:** Learning Rate (λ) for Adam



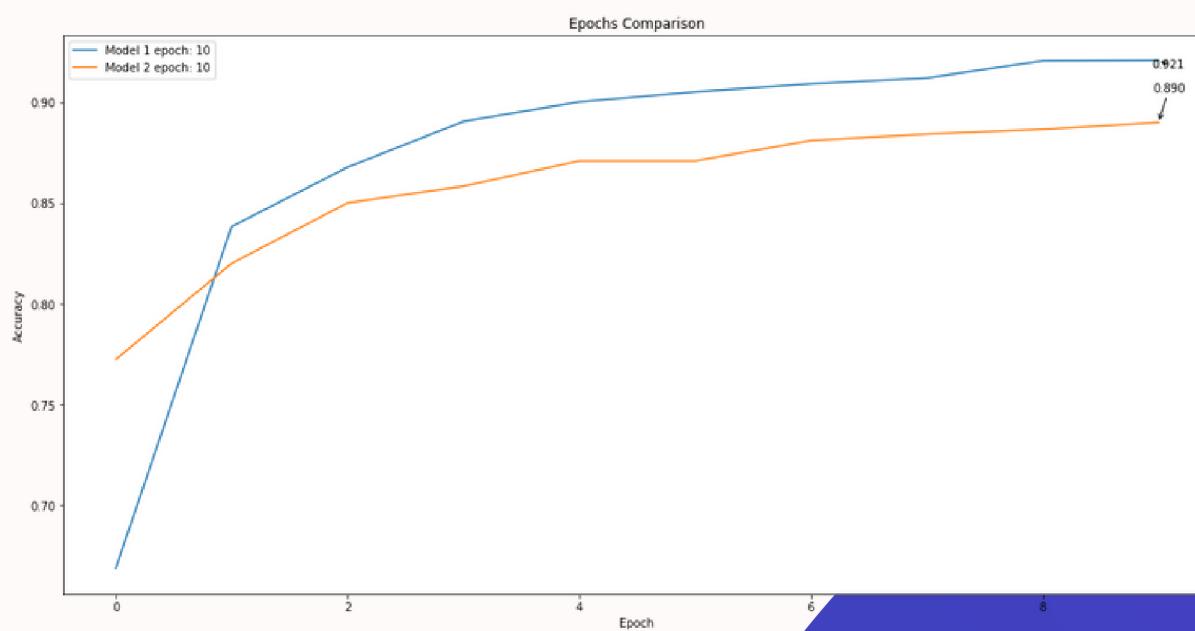
- **Experiment 5.2:** Epsilon (ϵ) for Adam



- **Experiment 5.3:** Batch Size



- **Experiment 5.4:** Number of Epochs



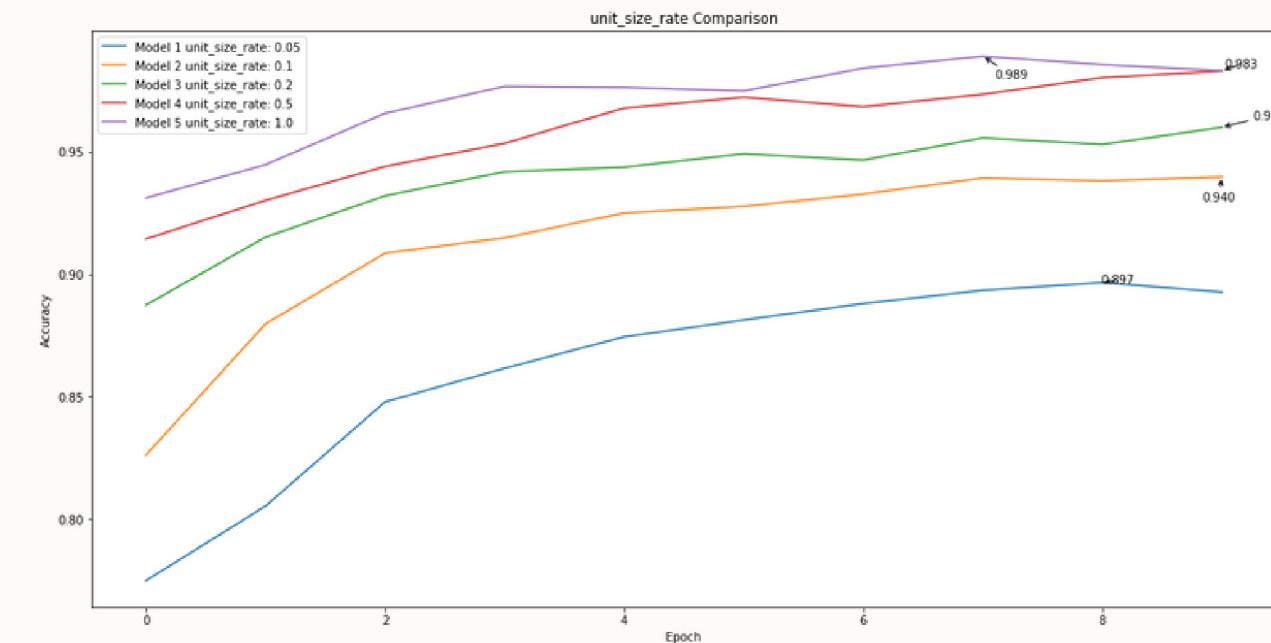
EXPERIMENTATION

Training and Evaluation and Experiment

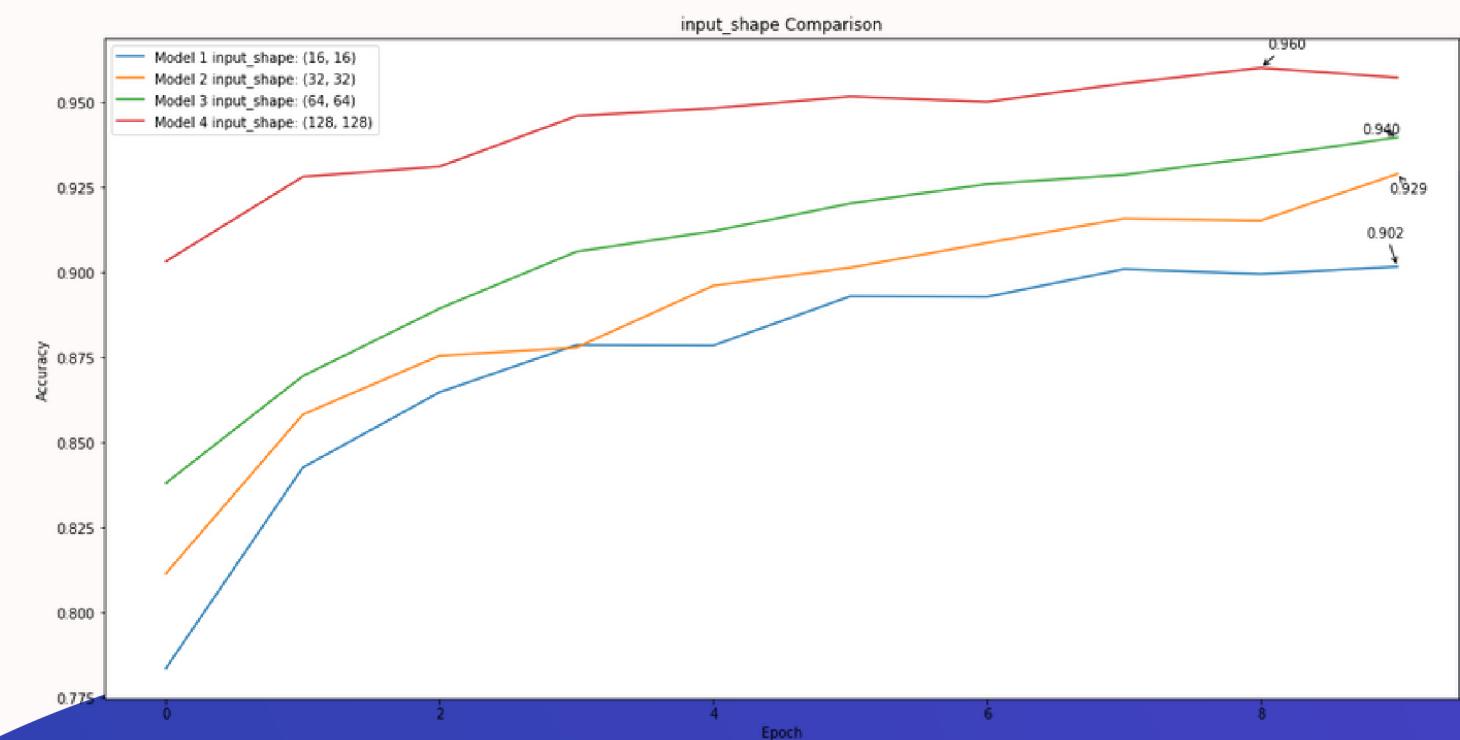
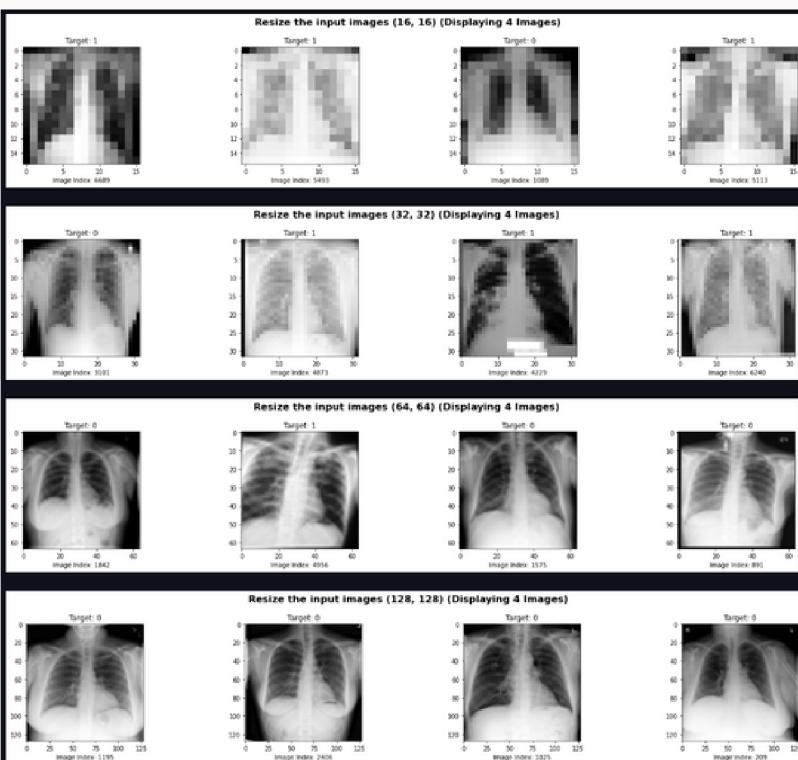
- **Experiment 6: Image Enhancement Techniques**

Model	Timestamp	Reference name	Optimizer parameters Configuration	Hyperparameters Configuration	Model Architecture Configuration	Model Training Configuration	accuracy_train	accuracy_val	Subset	Training in seconds	TP	FP	FN	TN	Precision	Recall	F1	AUC	Accuracy
CNN_Original (Normalized)	2023-09-25 16:37:42	Original dataset	{'optimizer_name': 'adam', 'learning_rate': 0.001}	{'conv_padding': 'same', 'conv_kernel_size': 3, 'unit_size_rate': 0.1}	{'input_shape': (128, 128), 'batch_size': 64, 'epoch': 10}	[0.7769643068313599, 0.9123214483261108]	[0.8471428751945496, 0.9257143139839172]	Test	91.363643	342	8	20	330	0.960541	0.960000	0.959988	0.960000	0.960000	0.960000
CNN_Original (PCA)	2023-09-25 16:39:24	Original dataset	{'optimizer_name': 'adam', 'learning_rate': 0.001}	{'conv_padding': 'same', 'conv_kernel_size': 3, 'unit_size_rate': 0.1}	{'input_shape': (128, 128), 'batch_size': 64, 'epoch': 10}	[0.7421428561210632, 0.9294642806053162]	[0.8899999856948853, 0.9428571462631226]	Test	96.033478	341	9	33	317	0.942079	0.940000	0.939929	0.940000	0.940000	0.940000
CNN_Contrast Stretching	2023-09-25 16:41:05	Original dataset	{'optimizer_name': 'adam', 'learning_rate': 0.001}	{'conv_padding': 'same', 'conv_kernel_size': 3, 'unit_size_rate': 0.1}	{'input_shape': (128, 128), 'batch_size': 64, 'epoch': 10}	[0.7076785564422607, 0.8553571701049805]	[0.8199999828474426, 0.8642857074737549]	Test	95.988213	335	15	19	331	0.951488	0.951429	0.951427	0.951429	0.951429	0.951429
CNN_Histogram Equalization	2023-09-25 16:42:45	Original dataset	{'optimizer_name': 'adam', 'learning_rate': 0.001}	{'conv_padding': 'same', 'conv_kernel_size': 3, 'unit_size_rate': 0.1}	{'input_shape': (128, 128), 'batch_size': 64, 'epoch': 10}	[0.8573214411735535, 0.9330357313156128]	[0.9342857003211975, 0.9471428394317627]	Test	94.518885	320	30	1	349	0.958865	0.955714	0.955638	0.955714	0.955714	0.955714
CNN_Adaptive Equalization	2023-09-25 16:44:23	Original dataset	{'optimizer_name': 'adam', 'learning_rate': 0.001}	{'conv_padding': 'same', 'conv_kernel_size': 3, 'unit_size_rate': 0.1}	{'input_shape': (128, 128), 'batch_size': 64, 'epoch': 10}	[0.8132143020629883, 0.9257143139839172]	[0.9057142734527588, 0.9442856907844543]	Test	92.421328	331	19	3	347	0.969553	0.968571	0.968555	0.968571	0.968571	0.968571

- **Experiment 7: Impact of Varying Unit Sizes in Neural Network Layers**

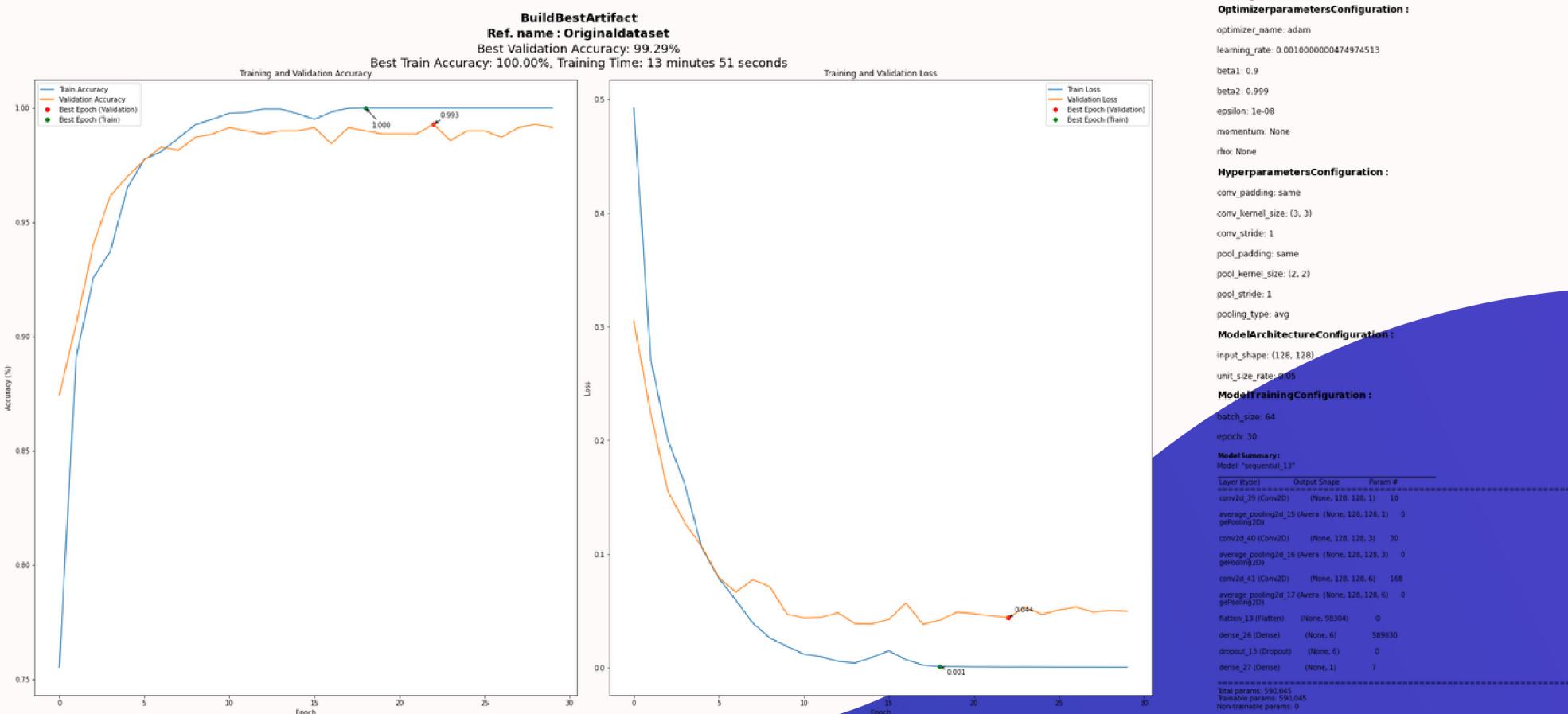


- **Experiment 8: Exploring Different Image Input Sizes**



HYPERPARAMETER TUNING

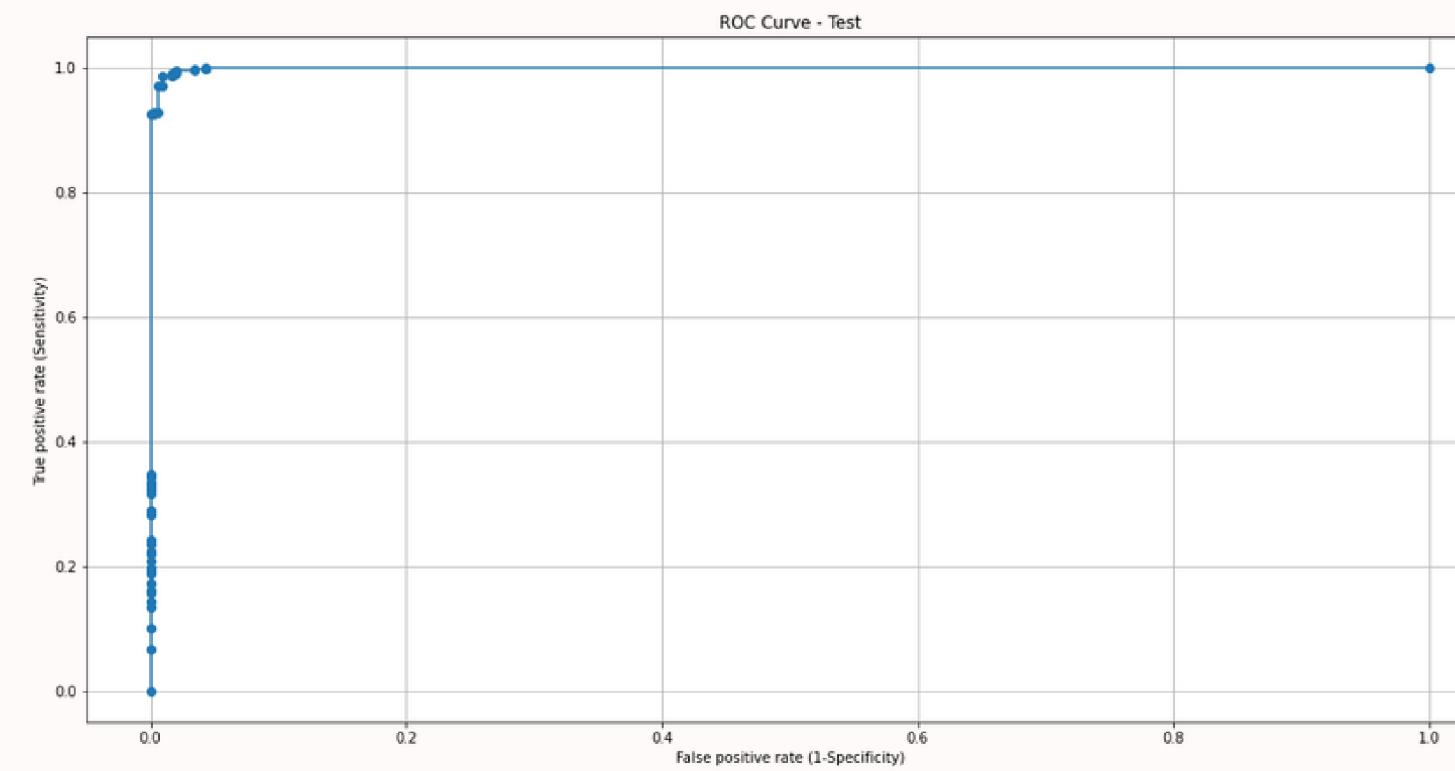
TUNING AND GET THE BEST ARTIFACT



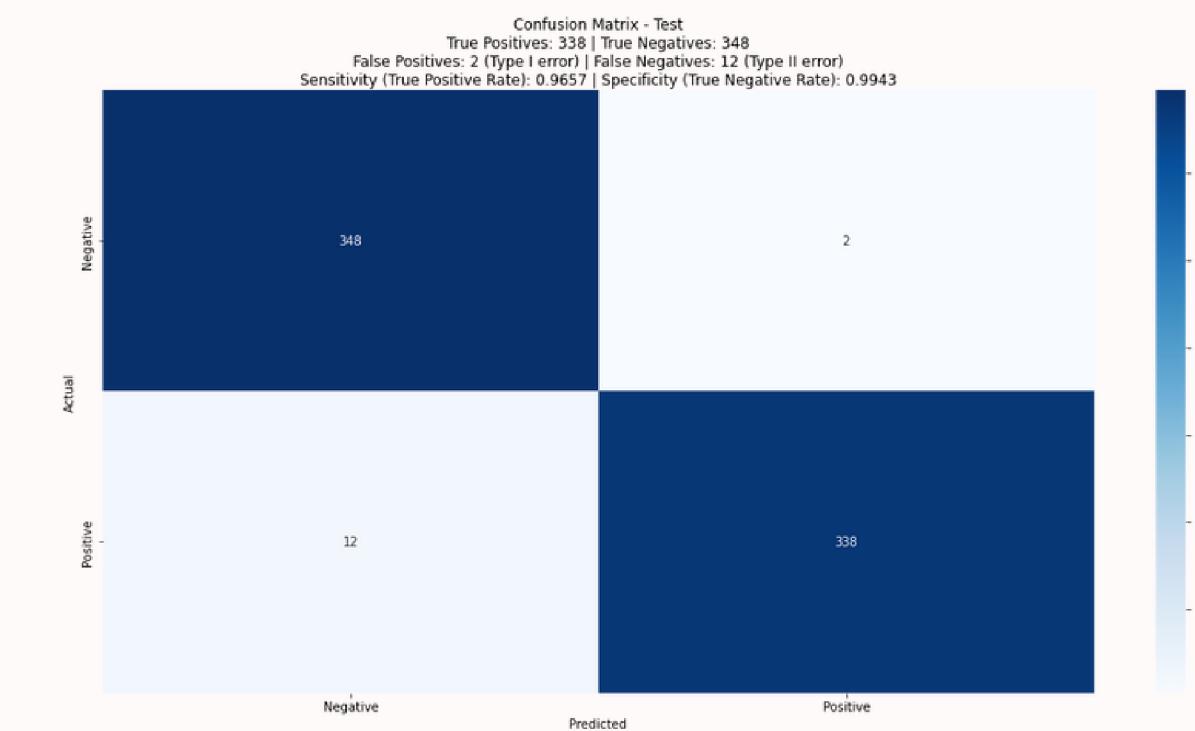
99%

EVALUATION

ROC



CONFUSION MATRIX



DEPLOYMENT

Gradio is the fastest way to demo your machine learning model with a friendly web interface so that anyone can use it, anywhere!

Tuberculosis Prediction

Upload X-ray Image and initiate the tuberculosis detection process. Clicking this submit triggers the deep learning model to analyze the uploaded X-ray image

Submit

Probability of disease

Disease

No Disease

Plot

Input

Image-Explanation

Prediction

The region has been classified as the positive class.

Flag

Pages: 1 2 3 ... 126 127 128 129 130 ... 138 139 140



THANKS YOU



CONTACT



099-465-5516



pisit.jant@gmail.com



Portfolio

<https://pisit-janthawee.mystrikingly.com/>

LinkedIn

<https://www.linkedin.com/in/pisit-janthawee/>

Github

<https://github.com/Pisit-Janthawee>



Bangkok, Bangkok City,
Thailand