COL780 Assignment - 2

Vidushi Maheshwari (2021CS10083)

March 30, 2025

Contents

Introduction **3**

I Standard Network Implementation and Ablation Studies **3**

1 Overview . . . 3
2 Baseline Model . . . 4
3 Observations . . . 5

Ablation Studies: ResNet **6**

1 Learning Rate . . . 6
2 Number of Layers . . . 8
3 Effect of Skip Connections . . . 10
4 Loss Function Comparision . . . 11
5 Learning Rate Scheduling . . . 12
6 Data Augmentation . . . 14
7 Optimizer Comparision . . . 15

Ablation Studies: VGG **16**

1 Learning Rate . . . 16
2 Number of Layers . . . 18
3 Loss Function Comparision . . . 20
4 Learning Rate Scheduling . . . 21
5 Data Augmentation . . . 23
6 Optimizer Comparision . . . 24
7 Observations and Analysis . . . 25

II Custom Architecture **28**

1 Overview . . . 28

2 Baseline Model 28

Ablation Studies **30**

3 Learning Rate 30

4 Data Augmentation 32

5 Optimizer Comparision 33

6 Observations and Analysis 34

7 Comparision with Part 1 35

III Model Improvement Competition **36**

1 Overview 36

2 Model Description 36

3 Results on Test Data 36

Google Drive Link **37**

Introduction

The goal of this assignment is to develop models for classifying histopathologic scans of lymph node sections using the PatchCamelyon (PCam) dataset. The dataset consists of 96×96 pixel RGB images, each labeled as containing metastatic tissue (positive) or not (negative).
The assignment is divided into three parts:

- Standard Architectures and Ablation Studies - Implement and analyze standard deep learning architectures, specifically ResNet and VGG, while conducting ablation studies to examine the impact of different hyperparameters and architectural choices.
- Custom Convolutional Neural Network (CNN) - Design and train a custom CNN incorporating residual blocks and an inception-style module.
- Model Improvement - Improve model performance using advanced techniques such as data augmentation, regularization, and learning rate scheduling.

This report provides a detailed analysis of model performance, including accuracy, precision, recall, and F1-score, along with comparisons between different architectures and training strategies.

Part I
Standard Network Implementation and Ablation Studies

§1 Overview

In this section, we evaluate ResNet and VGG on the PCam dataset, analyzing hyperparameter choices and architectural variations.

§1.1 ResNet

- ResNet (Residual Networks) is a deep convolutional neural network designed to address the vanishing gradient problem.
- It uses residual connections (skip connections) that help in training very deep models efficiently. The architecture is built with stacked residual blocks, each containing multiple convolutional layers.

§1.2 VGG

- VGG is a deep convolutional neural network characterized by its uniform architecture, which stacks multiple 3×3 convolutional layers with ReLU activation.
- It follows a sequential design with max pooling layers to reduce spatial dimensions and fully connected layers at the end for classification.

§1.3 Training and Hyperparameters

- **Batch Size**: 32
- **Epochs**: 25
- **Loss Functions**:
  - Cross-Entropy Loss
  - Focal Loss
- **Optimizers**:
  - Stochastic Gradient Descent (SGD) with momentum = 0.9
  - Adam Optimizer

§2 Baseline Model

§2.1 Hyperparameters Used

- Learning Rate: 10^{-3}
- Loss function: Cross-Entropy Loss
- Optimizers: Stochastic Gradient Descent (SGD) with momentum = 0.9

§2.2 Results on Test Data

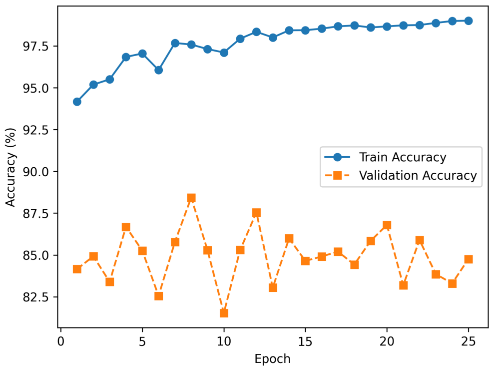
Model	Accuracy	Precision	Recall	F1-score
ResNet18	0.8125	0.8515	0.8124	0.8071
VGG16	0.8069	0.8528	0.8068	0.8003

Table 1: Baseline Performance of ResNet18 and VGG16 on Test Data

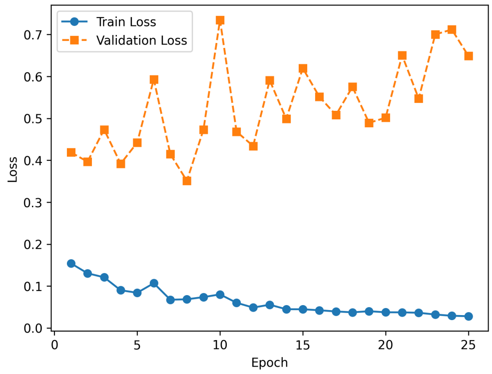
Model	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
ResNet18	97.10	98.66	99.01	81.53	86.80	84.76
VGG16	97.93	98.75	99.06	87.84	89.11	87.08

Table 2: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§2.3 Plots

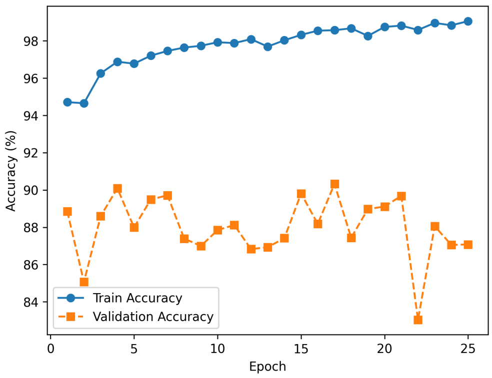


(a) Accuracy vs Epochs

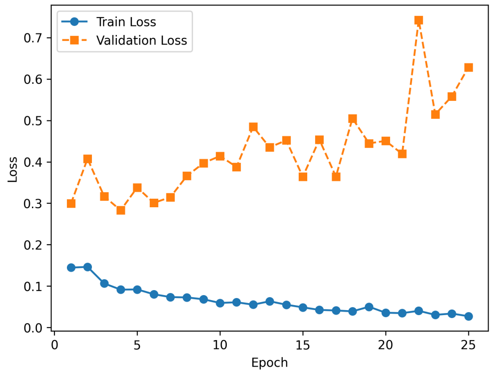


(b) Loss vs Epochs

Figure 1: Baseline ResNet18 Model



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 2: Baseline VGG16 Model

§3 Observations

- ResNet18 achieved an accuracy of 81.25%, slightly outperforming VGG16 (80.69%) on the test set.
- VGG16 demonstrated higher validation accuracy (89.11%) compared to ResNet18 (86.80%) at 20 epochs, suggesting better generalization.
- VGG16 had the highest precision (85.28%), indicating fewer false positives.
- Both models showed signs of overfitting, with training accuracies exceeding 97% while test accuracy remained in the 80s.

Ablation Studies: ResNet

§1 Learning Rate

§1.1 Results on Test Data

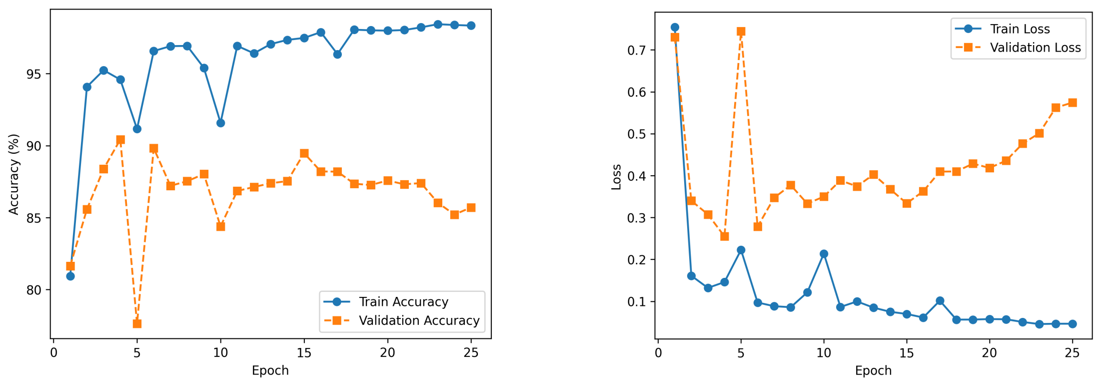
Learning Rate	Accuracy	Precision	Recall	F1-score
10^{-2}	0.8089	0.8497	0.8088	0.8031
10^{-3}	0.8125	0.8515	0.8124	0.8071
10^{-4}	0.8072	0.8478	0.8071	0.8014
10^{-5}	0.8138	0.8409	0.8137	0.8099

Table 3: Performance of ResNet18 on Test Data

Learning Rate	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
10^{-2}	91.59	97.99	98.34	84.39	87.58	85.71
10^{-3}	97.10	98.66	99.01	81.53	86.80	84.76
10^{-4}	95.84	97.00	97.36	86.19	85.72	85.01
10^{-5}	91.07	92.58	93.41	85.04	85.28	86.56

Table 4: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

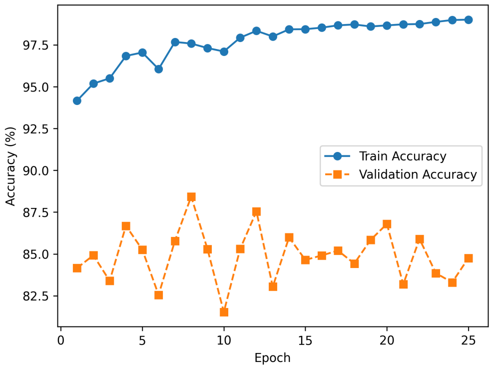
§1.2 Plots



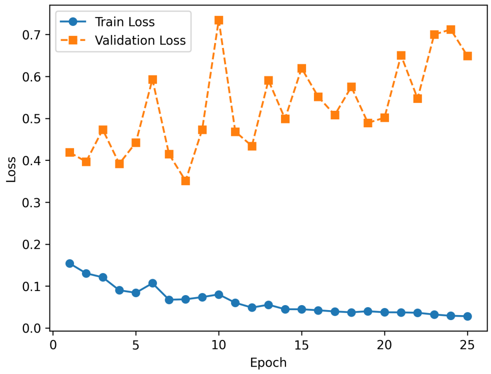
(a) Accuracy vs Epochs

(b) Loss vs Epochs

Figure 3: ResNet18 Model with learning rate = 10^{-2}

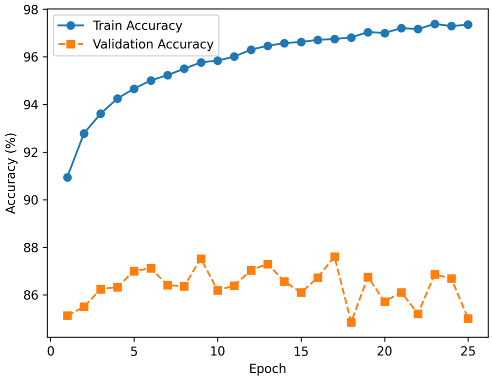


(a) Accuracy vs Epochs

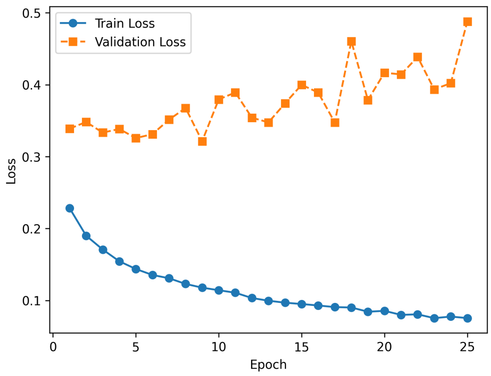


(b) Loss vs Epochs

Figure 4: ResNet18 Model with learning rate = 10^{-3}

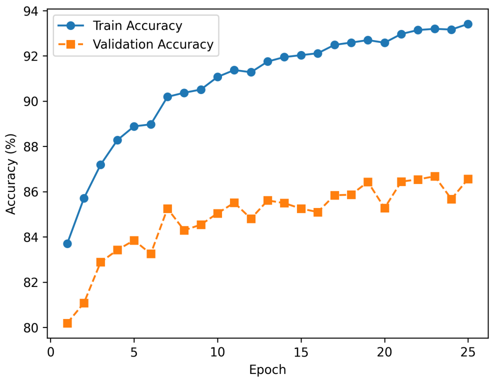


(a) Accuracy vs Epochs

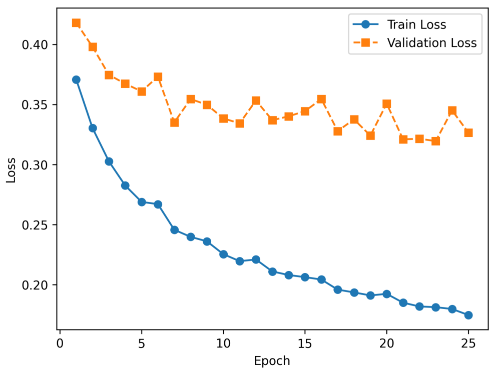


(b) Loss vs Epochs

Figure 5: ResNet18 Model with learning rate = 10^{-4}



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 6: ResNet18 Model with learning rate = 10^{-5}

§2 Number of Layers

§2.1 Results on Test Data

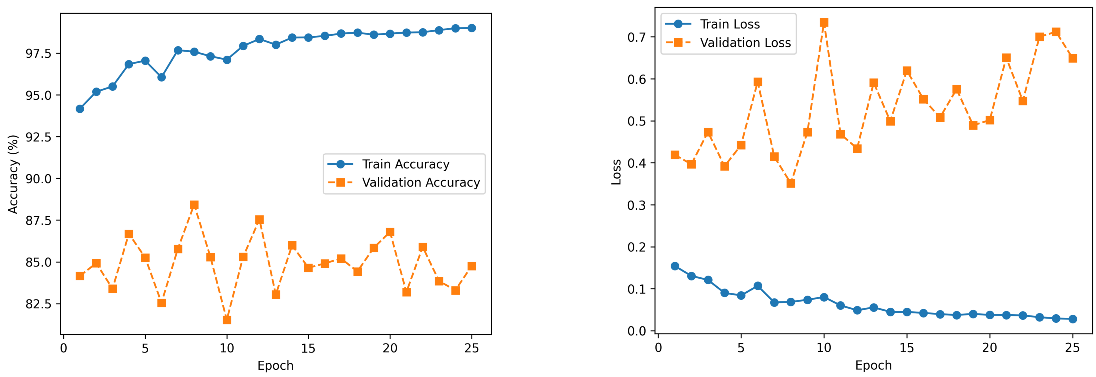
Model	Accuracy	Precision	Recall	F1-score
ResNet18	0.8125	0.8515	0.8124	0.8071
ResNet34	0.7994	0.8410	0.7993	0.7931
ResNet50	0.8015	0.8472	0.8015	0.7948

Table 5: Performance of ResNet on Test Data

Learning	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
ResNet18	97.10	98.66	99.01	81.53	86.80	84.76
ResNet34	99.57	99.85	99.88	84.84	85.11	85.01
ResNet50	99.57	99.84	99.92	87.19	84.14	85.00

Table 6: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

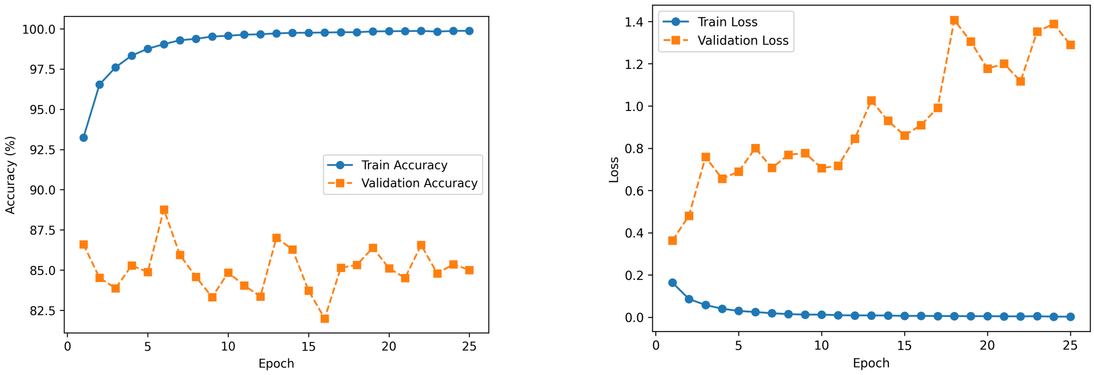
§2.2 Plots



(a) Accuracy vs Epochs

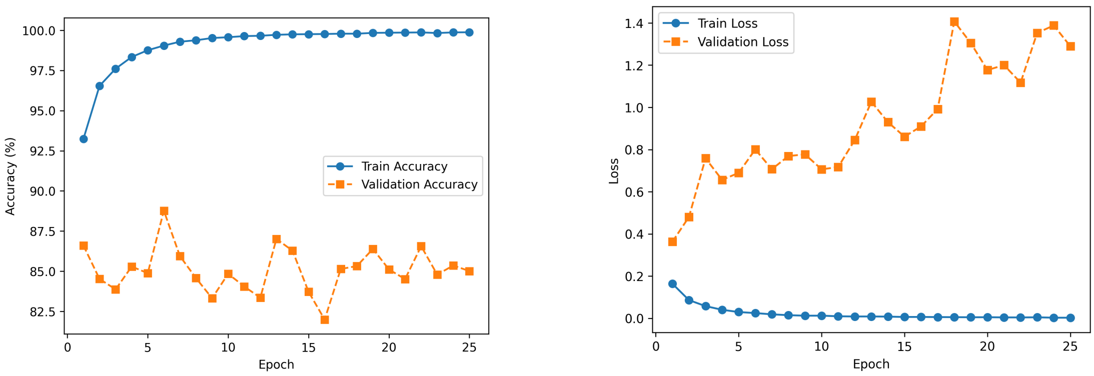
(b) Loss vs Epochs

Figure 7: ResNet18 Model



(a) Accuracy vs Epochs (b) Loss vs Epochs

Figure 8: ResNet34 Model



(a) Accuracy vs Epochs (b) Loss vs Epochs

Figure 9: ResNet50 Model

§3 Effect of Skip Connections

§3.1 Results on Test Data

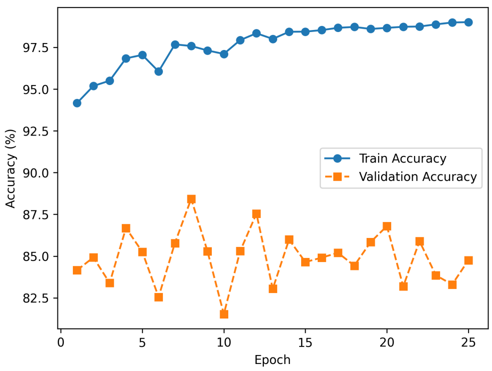
Skip	Accuracy	Precision	Recall	F1-score
True	0.8125	0.8515	0.8124	0.8071
False	0.8148	0.8505	0.8147	0.8099

Table 7: Performance of ResNet18 on Test Data

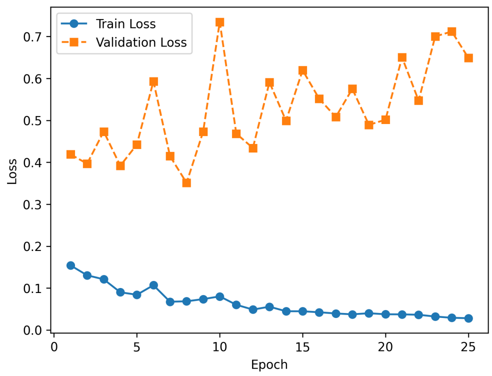
Skip	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
True	97.10	98.66	99.01	81.53	86.80	84.76
False	99.12	99.82	99.85	83.07	83.00	84.67

Table 8: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§3.2 Plots

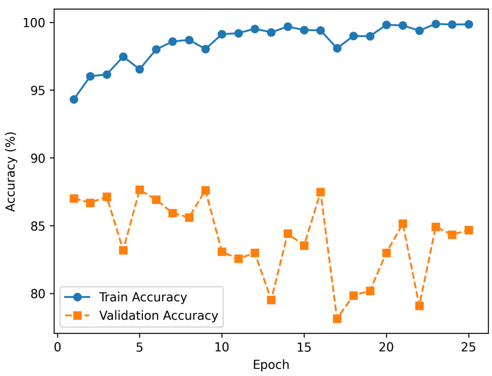


(a) Accuracy vs Epochs

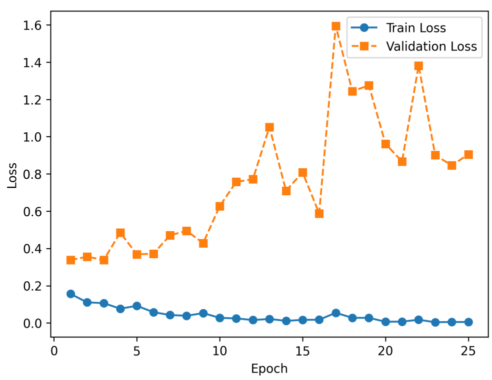


(b) Loss vs Epochs

Figure 10: ResNet18 Model with Skip Connections



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 11: ResNet18 Model without Skip Connections

§4 Loss Function Comparision

§4.1 Results on Test Data

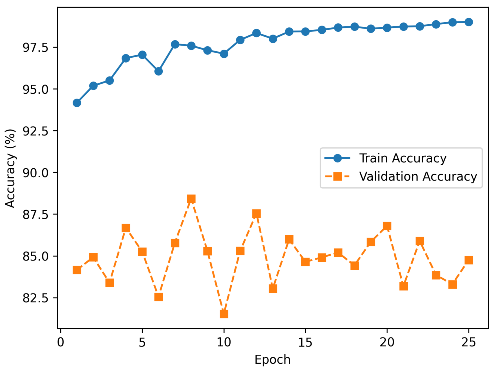
Loss	Accuracy	Precision	Recall	F1-score
Cross Entropy	0.8125	0.8515	0.8124	0.8071
Focal Loss	0.8129	0.8450	0.8128	0.8084

Table 9: Performance of ResNet18 on Test Data

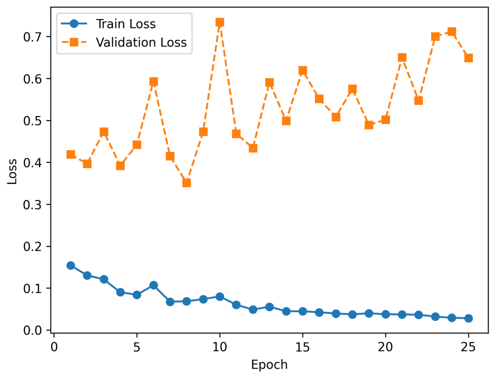
Loss	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
Cross Entropy	97.10	98.66	99.01	81.53	86.80	84.76
Focal Loss	99.65	99.91	99.92	82.67	84.03	84.09

Table 10: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§4.2 Plots

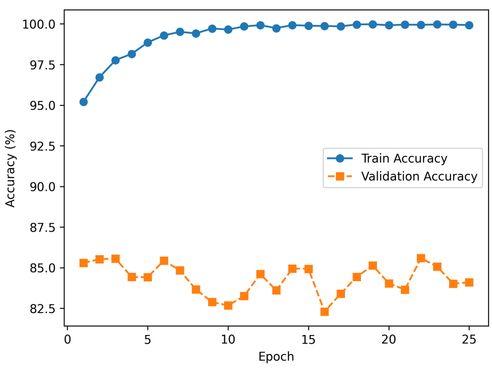


(a) Accuracy vs Epochs

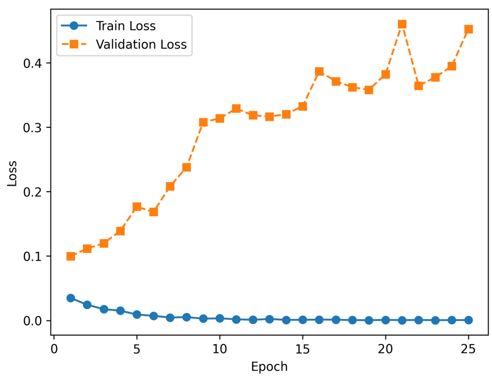


(b) Loss vs Epochs

Figure 12: ResNet18 Model with Cross Entropy Loss



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 13: ResNet18 Model with Focal Loss

§5 Learning Rate Scheduling

§5.1 Results on Test Data

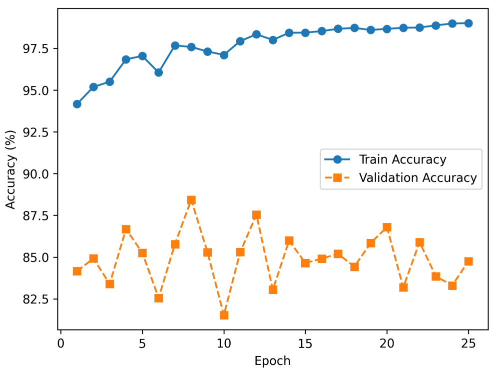
LR Scheduling	Accuracy	Precision	Recall	F1-score
None	0.8125	0.8515	0.8124	0.8071
StepLR	0.8055	0.8475	0.8055	0.7995
Cosine Annealing	0.8124	0.8511	0.8123	0.8070

Table 11: Performance of ResNet18 on Test Data

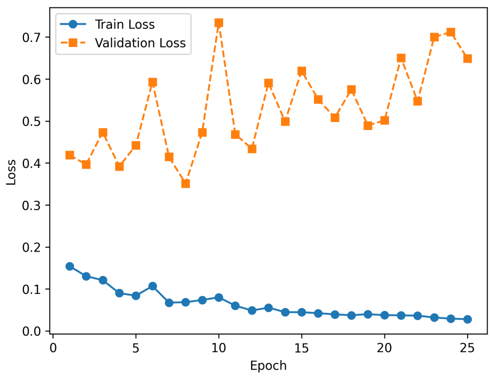
LR Scheduling	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
None	97.10	98.66	99.01	81.53	86.80	84.76
StepLR	97.36	98.65	98.72	82.03	85.85	85.03
Cosine Annealing	97.79	98.92	99.05	86.95	86.07	85.81

Table 12: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§5.2 Plots

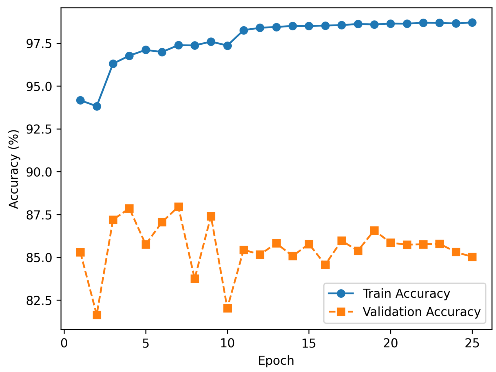


(a) Accuracy vs Epochs

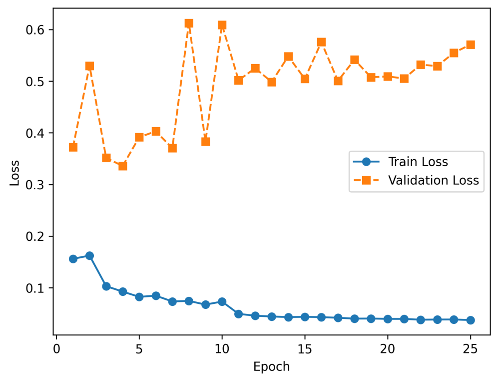


(b) Loss vs Epochs

Figure 14: ResNet18 Model with constant LR

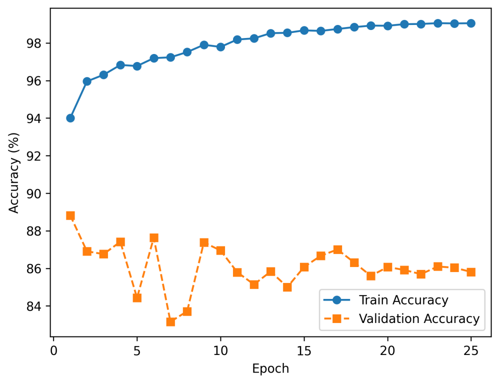


(a) Accuracy vs Epochs

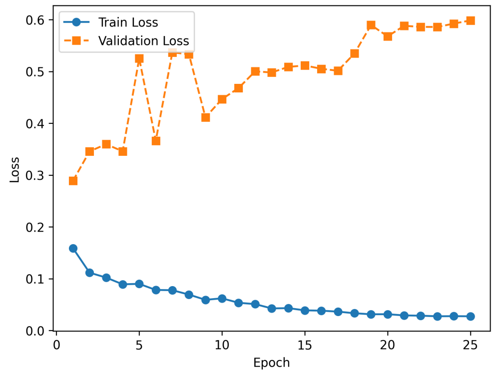


(b) Loss vs Epochs

Figure 15: ResNet18 Model with StepLR



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 16: ResNet18 Model with Cosine Annealing

§6 Data Augmentation

§6.1 Results on Test Data

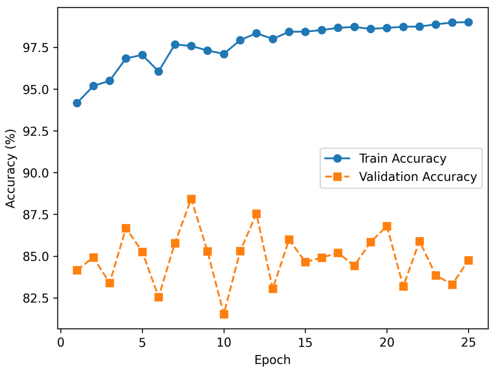
Data Augmentation	Accuracy	Precision	Recall	F1-score
False	0.8125	0.8515	0.8124	0.8071
True	0.8260	0.8577	0.8259	0.8220

Table 13: Performance of ResNet18 on Test Data

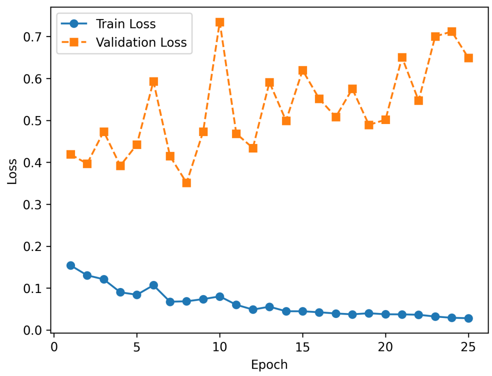
Data Augmentation	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
False	97.10	98.66	99.01	81.53	86.80	84.76
True	97.74	98.74	98.98	88.45	86.16	86.06

Table 14: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§6.2 Plots

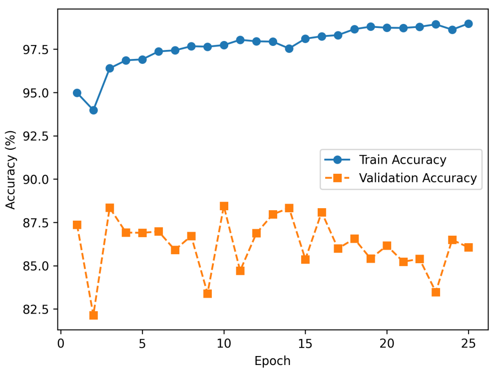


(a) Accuracy vs Epochs

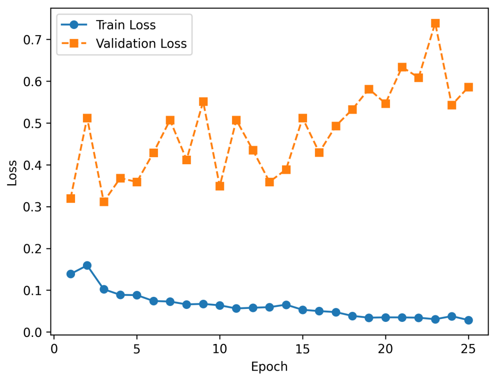


(b) Loss vs Epochs

Figure 17: ResNet18 Model without Data Augmentation



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 18: ResNet18 Model with Data Augmentation

§7 Optimizer Comparision

§7.1 Results on Test Data

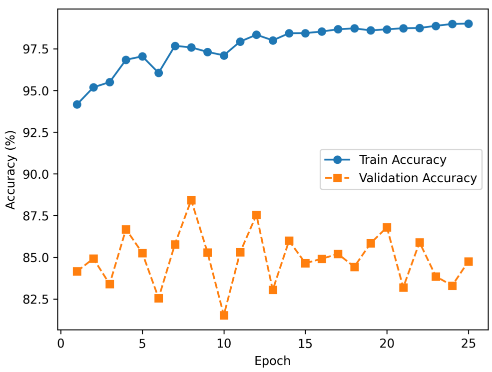
Optimizer	Accuracy	Precision	Recall	F1-score
SGD	0.8125	0.8515	0.8124	0.8071
Adam	0.7795	0.8366	0.7794	0.7697

Table 15: Performance of ResNet18 on Test Data

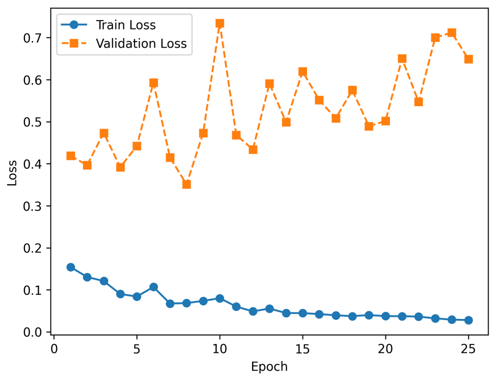
Optimizer	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
SGD	97.10	98.66	99.01	81.53	86.80	84.76
Adam	99.12	99.74	97.85	81.22	83.88	77.80

Table 16: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§7.2 Plots

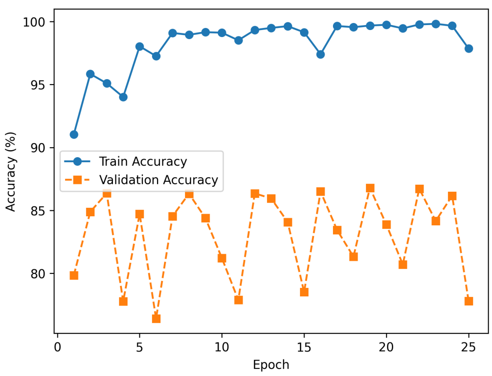


(a) Accuracy vs Epochs

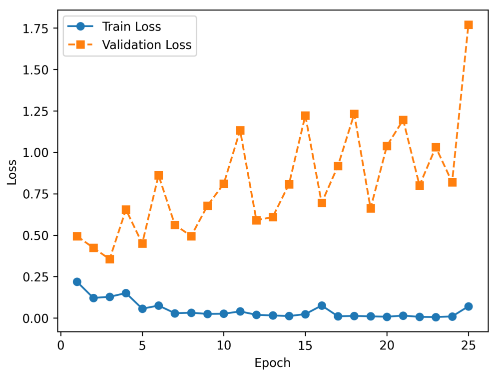


(b) Loss vs Epochs

Figure 19: ResNet18 Model with SGD Optimizer with Momentum



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 20: ResNet18 Model with Adam Optimizer

Ablation Studies: VGG

§1 Learning Rate

§1.1 Results on Test Data

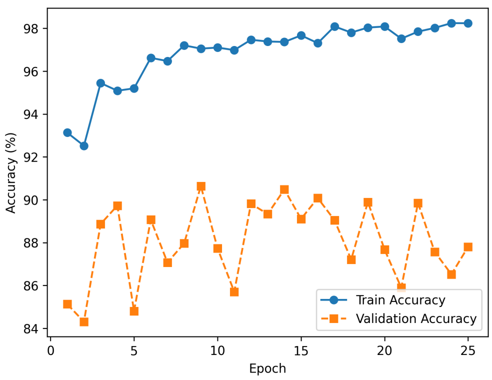
Learning Rate	Accuracy	Precision	Recall	F1-score
10^{-2}	0.8560	0.8749	0.8559	0.8541
10^{-3}	0.8069	0.8528	0.8068	0.8003
10^{-4}	0.8625	0.8823	0.8625	0.8607
10^{-5}	0.8619	0.8806	0.8619	0.8602

Table 17: Performance of VGG16 on Test Data

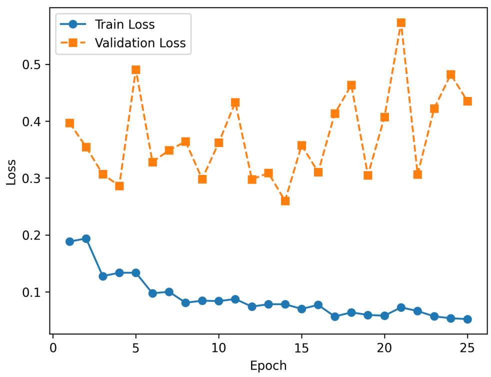
Learning Rate	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
10^{-2}	97.10	98.09	98.24	87.73	87.67	87.80
10^{-3}	97.93	98.75	99.06	87.84	89.11	87.08
10^{-4}	96.25	97.58	97.70	89.73	89.05	89.24
10^{-5}	92.76	94.17	94.76	87.98	88.75	88.12

Table 18: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§1.2 Plots

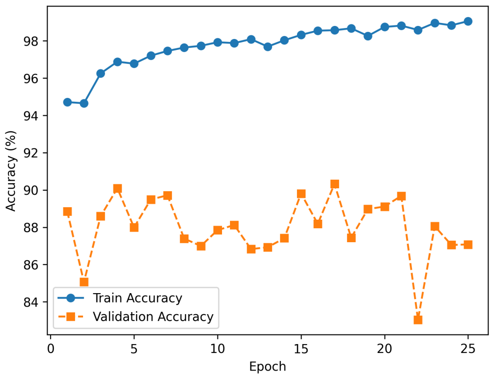


(a) Accuracy vs Epochs

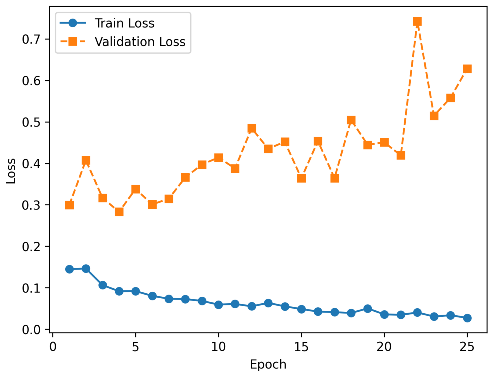


(b) Loss vs Epochs

Figure 21: VGG16 Model with learning rate = 10^{-2}

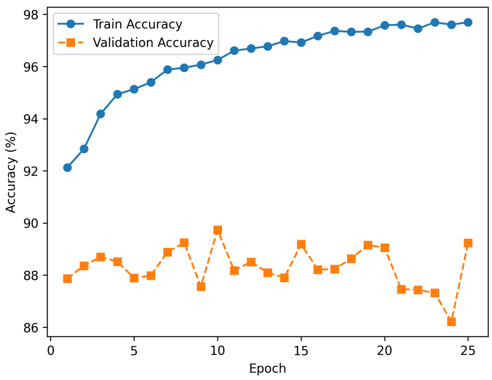


(a) Accuracy vs Epochs

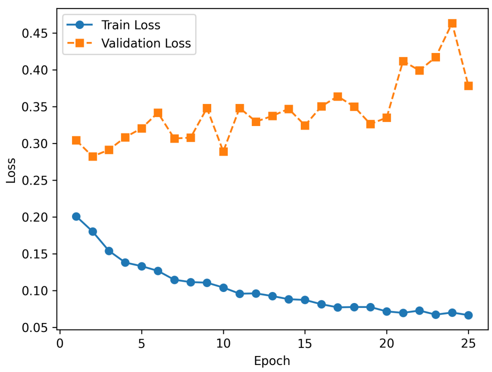


(b) Loss vs Epochs

Figure 22: VGG16 Model with learning rate = 10^{-3}

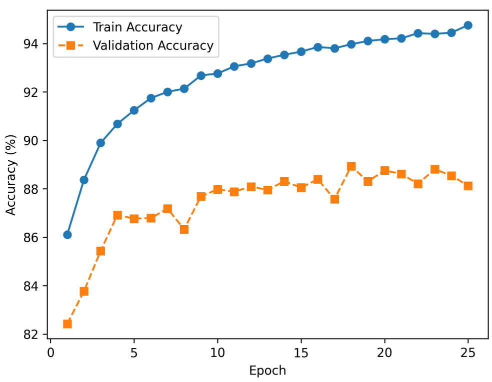


(a) Accuracy vs Epochs

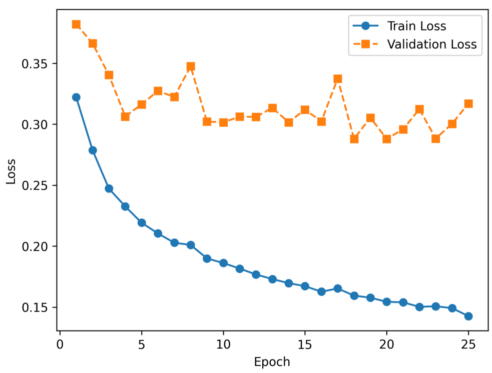


(b) Loss vs Epochs

Figure 23: VGG16 Model with learning rate = 10^{-4}



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 24: VGG16 Model with learning rate = 10^{-5}

§2 Number of Layers

§2.1 Results on Test Data

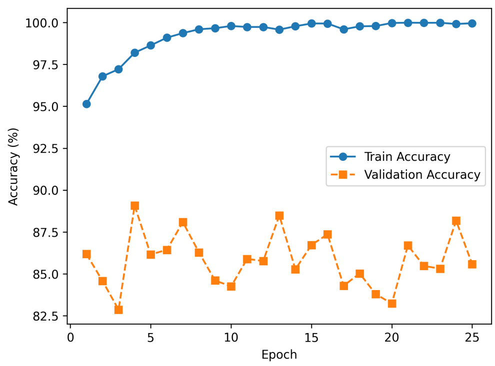
Model	Accuracy	Precision	Recall	F1-score
VGG11	0.8146	0.8540	0.8145	0.8093
VGG13	0.8478	0.8740	0.8477	0.8450
VGG16	0.8069	0.8528	0.8068	0.8003
VGG19	0.8406	0.8700	0.8405	0.8373

Table 19: Performance of VGG on Test Data

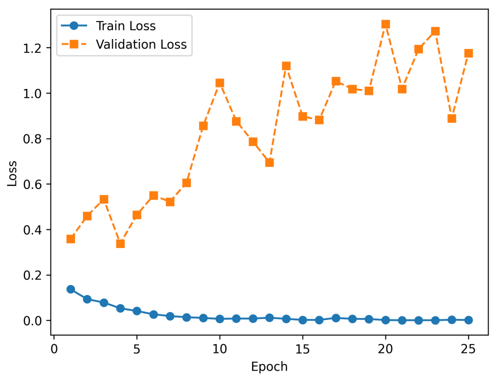
Model	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
VGG11	99.79	99.97	99.96	84.26	83.23	85.58
VGG13	99.73	99.98	99.99	88.38	87.38	87.15
VGG16	97.93	98.75	99.06	87.84	89.11	87.08
VGG19	99.55	99.87	99.63	89.07	84.65	83.83

Table 20: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§2.2 Plots

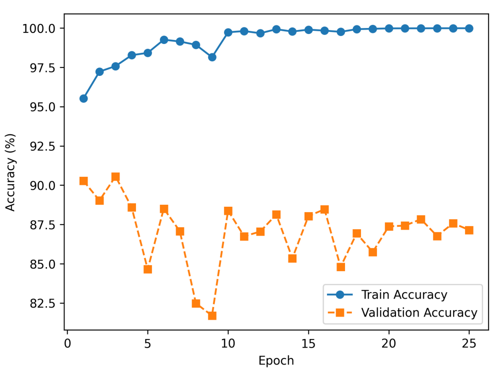


(a) Accuracy vs Epochs

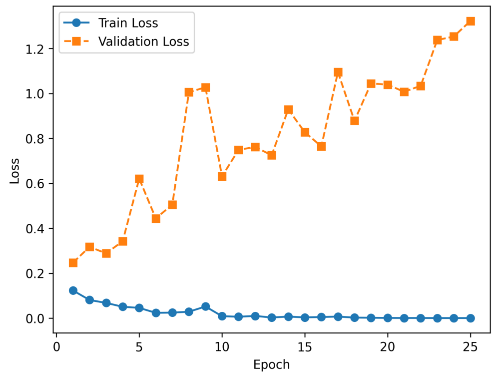


(b) Loss vs Epochs

Figure 25: VGG11 Model

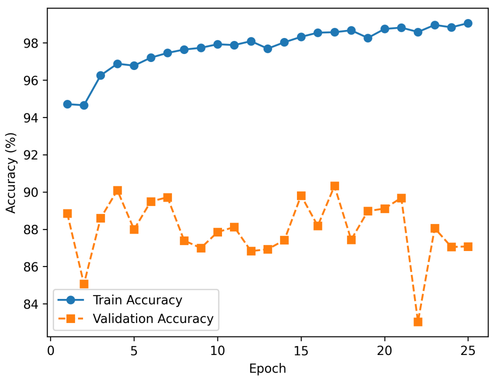


(a) Accuracy vs Epochs

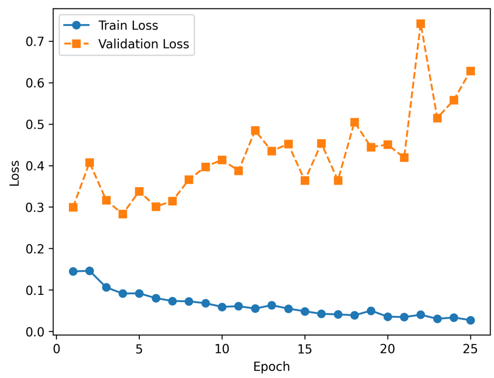


(b) Loss vs Epochs

Figure 26: VGG13 Model

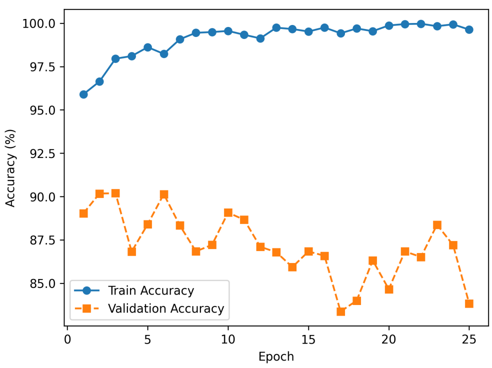


(a) Accuracy vs Epochs

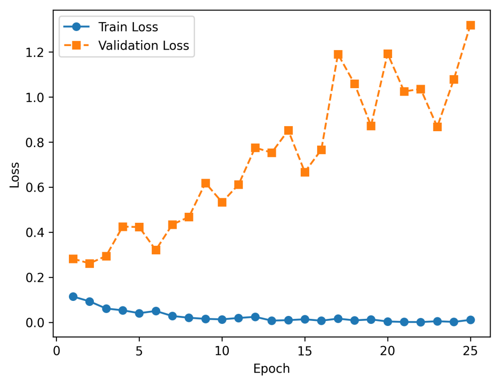


(b) Loss vs Epochs

Figure 27: VGG16 Model



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 28: VGG19 Model

§3 Loss Function Comparision

§3.1 Results on Test Data

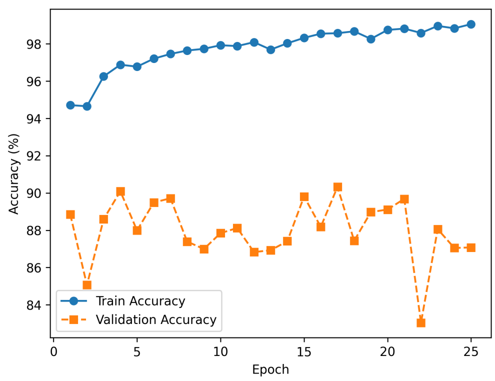
Loss	Accuracy	Precision	Recall	F1-score
Cross Entropy	0.8069	0.8528	0.8068	0.8003
Focal Loss	0.8385	0.8673	0.8385	0.8353

Table 21: Performance of VGG16 on Test Data

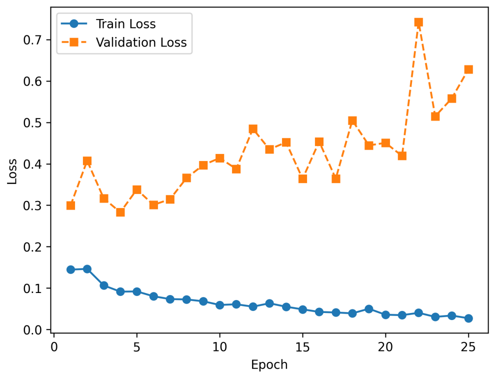
Loss	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
Cross Entropy	97.93	98.75	99.06	87.84	89.11	87.08
Focal Loss	99.24	99.96	99.99	86.01	85.24	86.12

Table 22: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§3.2 Plots

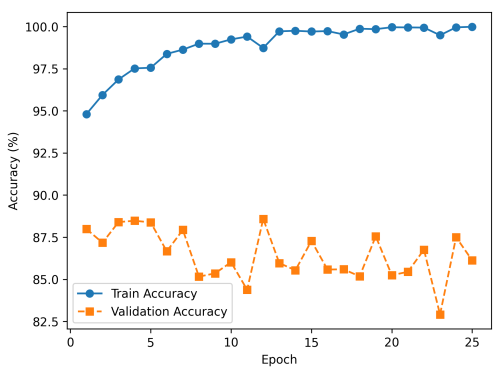


(a) Accuracy vs Epochs

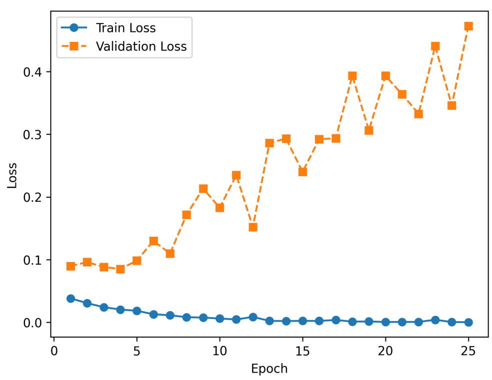


(b) Loss vs Epochs

Figure 29: VGG16 Model with Cross Entropy Loss



(a) Accuracy vs Epochs



§4 Learning Rate Scheduling

§4.1 Results on Test Data

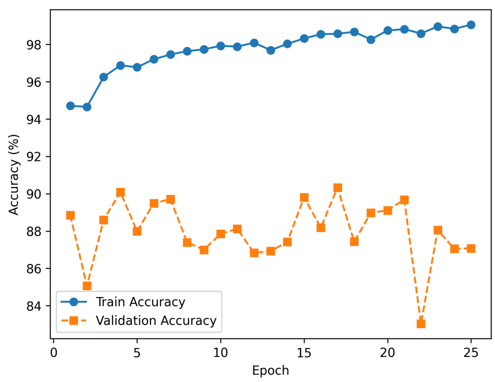
LR Scheduling	Accuracy	Precision	Recall	F1-score
None	0.8069	0.8528	0.8068	0.8003
StepLR	0.8369	0.8680	0.8369	0.8334
Cosine Annealing	0.8194	0.8579	0.8194	0.8144

Table 23: Performance of VGG16 on Test Data

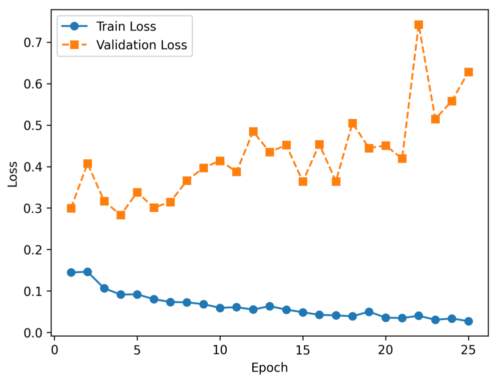
LR Scheduling	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
None	97.93	98.75	99.06	87.84	89.11	87.08
StepLR	97.75	98.74	98.84	88.46	88.74	88.14
Cosine Annealing	97.94	99.13	99.28	88.32	87.16	87.30

Table 24: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§4.2 Plots

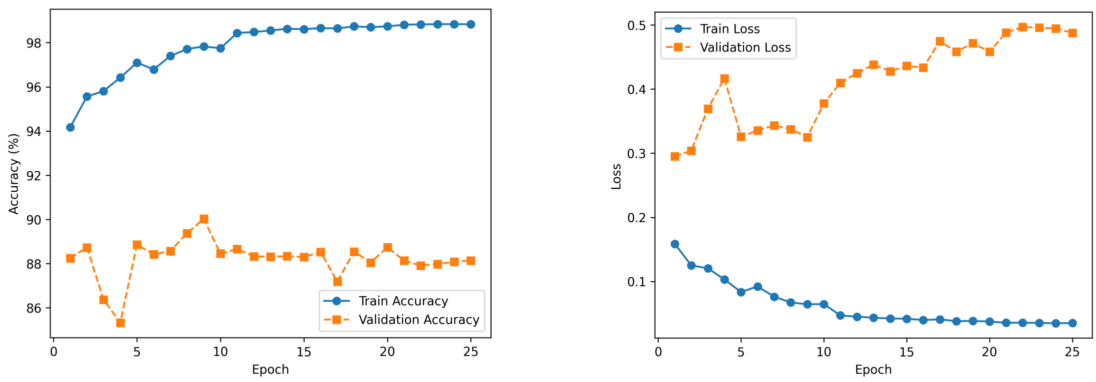


(a) Accuracy vs Epochs



(b) Loss vs Epochs

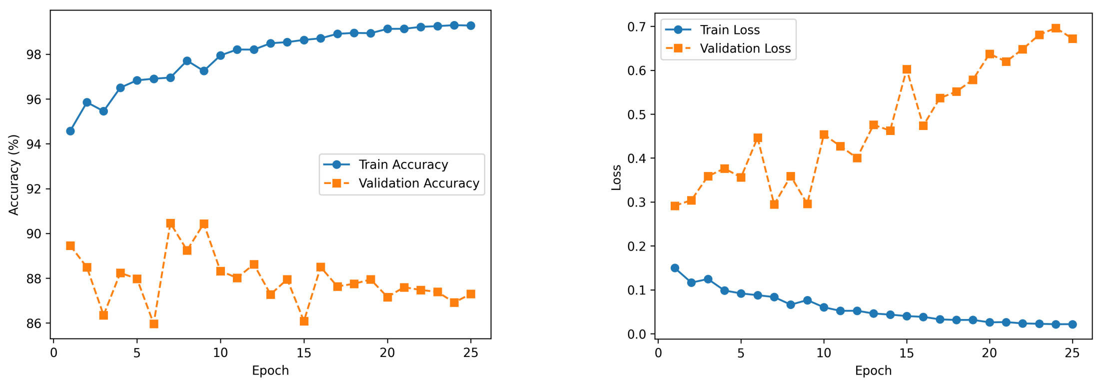
Figure 31: VGG16 Model with constant LR



(a) Accuracy vs Epochs

(b) Loss vs Epochs

Figure 32: VGG16 Model with StepLR



(a) Accuracy vs Epochs

(b) Loss vs Epochs

Figure 33: VGG16 Model with Cosine Annealing

§5 Data Augmentation

§5.1 Results on Test Data

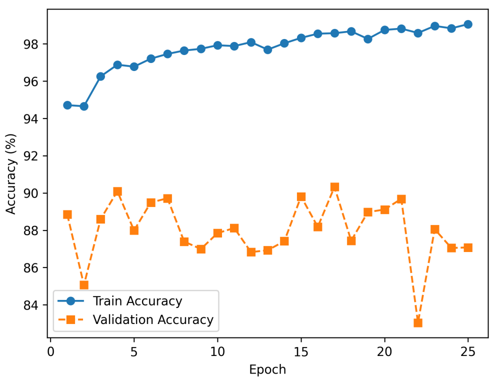
Data Augmentation	Accuracy	Precision	Recall	F1-score
False	0.8069	0.8528	0.8068	0.8003
True	0.8211	0.8580	0.8210	0.8163

Table 25: Performance of VGG16 on Test Data

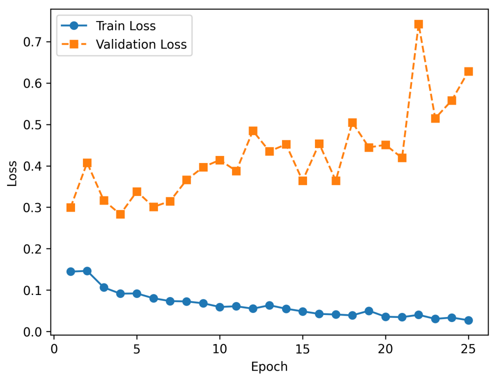
Data Augmentation	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
False	97.93	98.75	99.06	87.84	89.11	87.08
True	97.77	98.07	98.91	86.79	85.86	88.73

Table 26: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§5.2 Plots

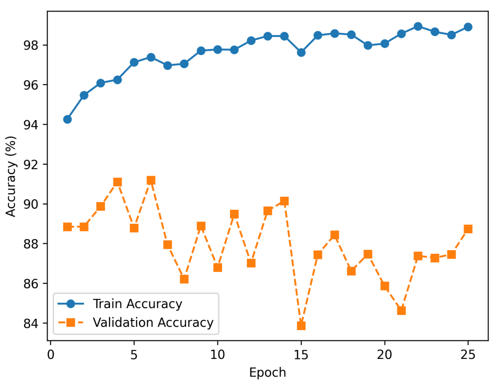


(a) Accuracy vs Epochs

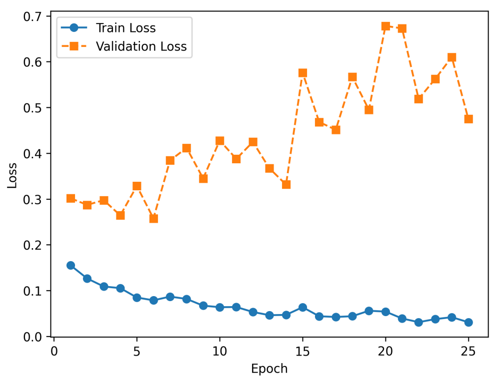


(b) Loss vs Epochs

Figure 34: VGG16 Model without Data Augmentation



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 35: VGG16 Model with Data Augmentation

§6 Optimizer Comparision

§6.1 Results on Test Data

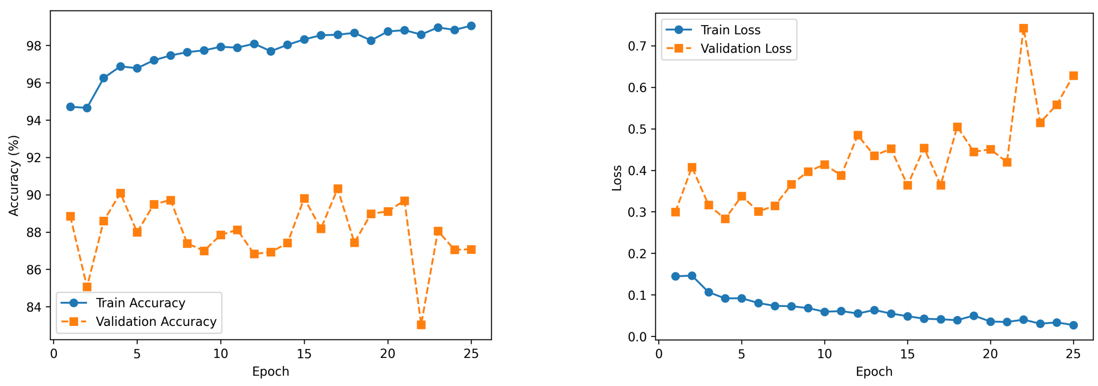
Optimizer	Accuracy	Precision	Recall	F1-score
SGD	0.8069	0.8528	0.8068	0.8003
Adam	0.5002	0.2501	0.5000	0.3334

Table 27: Performance of VGG16 on Test Data

Optimizer	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
SGD	97.93	98.75	99.06	87.84	89.11	87.08
Adam	50.00	50.00	50.00	50.05	49.95	50.05

Table 28: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

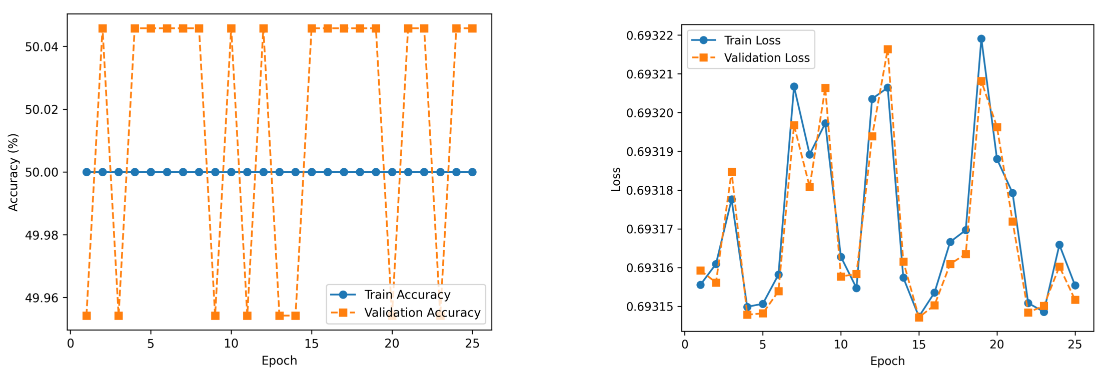
§6.2 Plots



(a) Accuracy vs Epochs

(b) Loss vs Epochs

Figure 36: VGG16 Model with SGD Optimizer with Momentum



(a) Accuracy vs Epochs

(b) Loss vs Epochs

Figure 37: VGG16 Model with Adam Optimizer

§7 Observations and Analysis

Learning Rate

- For ResNet18, learning rate 10^{-5} resulted in the highest test accuracy (81.38%), showing that a smaller learning rate improved generalization.
- Learning rate 10^{-2} led to high validation accuracy at earlier epochs but caused instability, leading to a final test accuracy of 80.89%.
- Learning rate 10^{-3} , the default setting, resulted in stable convergence but did not achieve the highest test accuracy.
- For VGG16, learning rate 10^{-4} was the best, achieving the highest test accuracy (86.25%), demonstrating that deeper networks benefited from smaller learning rates.
- Learning rate 10^{-2} also performed well for VGG16, achieving 85.60% test accuracy, but it showed unstable training behavior.
- Learning rate 10^{-3} was unstable for VGG, and 10^{-5} was too slow, leading to suboptimal results.

Number of Layers

- ResNet18 provided the best balance of depth and performance, achieving 81.25% accuracy.
- ResNet34 and ResNet50 performed slightly worse (79.94% and 80.15%), suggesting diminishing returns with increased depth.
- VGG13 (84.78%) surpassed VGG16, suggesting that a moderate decrease in depth was beneficial.
- Deeper models such as ResNet50 and VGG19 tended to overfit, achieving high training accuracy but without clear improvements in test accuracy.

Skip Connections

- Removing skip connections resulted in a minor accuracy increase (81.48% vs. 81.25%), indicating that residual learning was not as crucial for this dataset.
- However, the training accuracy without skip connections reached 99.85%, indicating a high degree of overfitting.
- The drop in validation accuracy suggests that skip connections help stabilize training by allowing gradients to propagate better.

Loss Function

- Focal Loss improved accuracy for both models, with ResNet18 achieving 81.29% and VGG16 reaching 83.85%. This suggests that Focal Loss helped the model focus on harder-to-classify samples.

- Cross-Entropy Loss performed worse, particularly for VGG16, indicating that it was more sensitive to class imbalance.
- The slight improvement with Focal Loss in ResNet suggests that misclassified samples were given higher importance during training.

Learning Rate Scheduling

- StepLR scheduling improved VGG16 performance to 83.69%, demonstrating its effectiveness in stabilizing learning.
- Cosine Annealing had minimal effect on ResNet18 (81.24%), suggesting that it did not significantly influence convergence.
- StepLR appeared more beneficial for deeper models like VGG16, likely because it helped adjust learning rates at the right time.

Effect of Data Augmentation

- Data augmentation significantly improved test accuracy for both models:
  - ResNet18: 81.25% → 82.60%
  - VGG16: 80.69% → 82.11%
- The improvements suggest that data augmentation effectively reduced overfitting by increasing training data diversity.
- Augmentation techniques such as random flips and rotations helped ResNet and VGG generalize better, indicating that more variations in training data were beneficial.

Optimizer Comparison

- SGD outperformed Adam in both models:
  - ResNet18: SGD (81.25%) vs. Adam (77.95%)
  - VGG16: SGD (80.69%) vs. Adam (50.02%)
- Adam caused severe overfitting in VGG16, where training accuracy remained at 99%, but test accuracy dropped to 50%, indicating instability and poor generalization.
- The poor performance of Adam in VGG suggests that it may not handle deeper networks well without careful tuning of the learning rate.
- SGD with momentum provided stable convergence, reinforcing its reliability for deep learning models.

Best Performing Model

§7.1 Best ResNet Model

- Test Accuracy: 82.60%
- Layer: ResNet18
- Learning Rate: 10^{-3}
- Optimizer: SGD with momentum = 0.9
- Loss Function: Cross-Entropy Loss
- Learning Rate Scheduler: No scheduler used in the best-performing run
- Data Augmentation: Enabled

§7.2 Best VGG Model

- Test Accuracy: 86.25%
- Layer: VGG16
- Learning Rate: 10^{-4}
- Optimizer: SGD with momentum = 0.9
- Loss Function: Cross-Entropy Loss
- Learning Rate Scheduler: No scheduler used in the best-performing run
- Data Augmentation: Disabled (although it would have improved generalization and boosted accuracy)

The overall highest test accuracy of 86.25% observed was achieved with the VGG16 Model with learning rate 10^{-4}

Part II
Custom Architecture

§1 Overview

This section focuses on designing and implementing a custom convolutional neural network (CNN) for the PCam dataset. The goal is to build a model that incorporates both residual connections and an inception-style module to enhance feature extraction and classification performance.

§1.1 Architecture

The custom CNN consists of the following components:

- **Initial Convolution Block**: A 3×3 convolution with 64 filters, batch normalization, and ReLU activation.
- **Residual Blocks**: Three blocks with depthwise separable convolutions and skip connections.
- **Inception Module**: Parallel convolutional layers (1×1, 3×3, and 5×5) concatenated along the channel dimension.
- **Global Average Pooling**: Reduces spatial dimensions before classification.
- **Fully Connected Layers**: A dense layer with dropout, followed by an output neuron with sigmoid activation for binary classification.

§2 Baseline Model

§2.1 Hyperparameters Used

- Learning Rate: 10^{-3}
- Loss function: Binary Cross Entropy Loss
- Optimizers: Stochastic Gradient Descent (SGD) with momentum = 0.9

§2.2 Results on Test Data

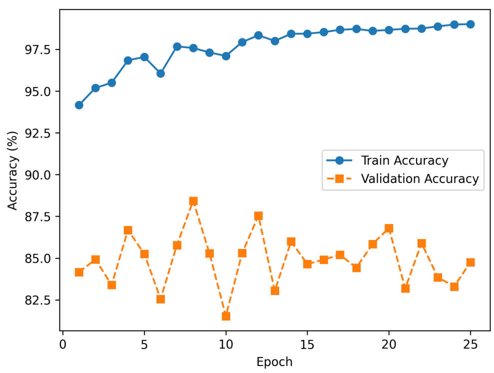
Model	Accuracy	Precision	Recall	F1-score
ResNet18	0.8125	0.8515	0.8124	0.8071
VGG16	0.8069	0.8528	0.8068	0.8003
Custom CNN	0.7646	0.8114	0.7645	0.7554

Table 29: Baseline Performance of ResNet18, VGG16 and Custom CNN

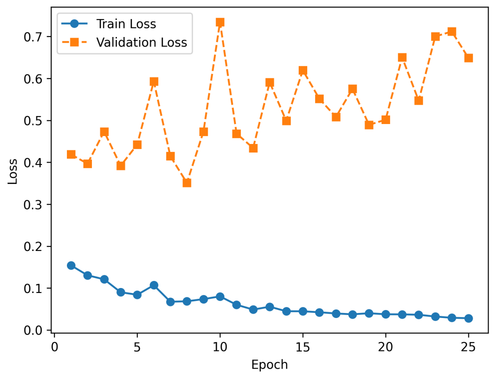
Model	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
ResNet18	97.10	98.66	99.01	81.53	86.80	84.76
VGG16	97.93	98.75	99.06	87.84	89.11	87.08
Custom CNN	80.37	94.21	94.79	71.91	86.46	84.20

Table 30: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§2.3 Plots

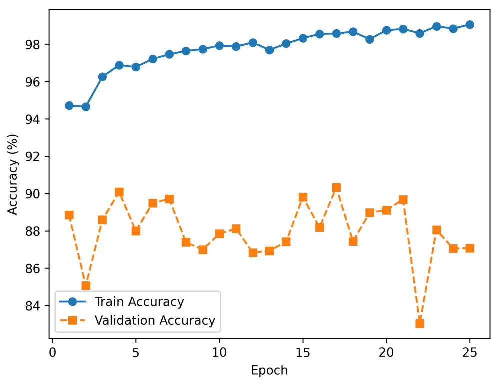


(a) Accuracy vs Epochs

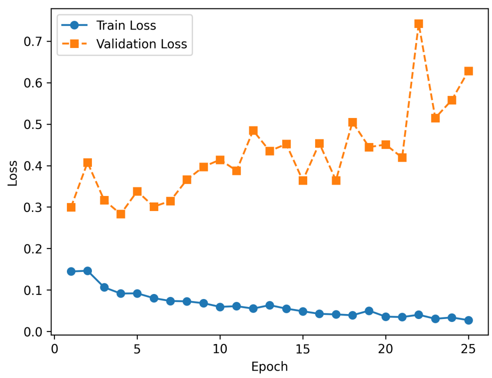


(b) Loss vs Epochs

Figure 38: Baseline ResNet18 Model



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 39: Baseline VGG16 Model

Ablation Studies

§3 Learning Rate

§3.1 Results on Test Data

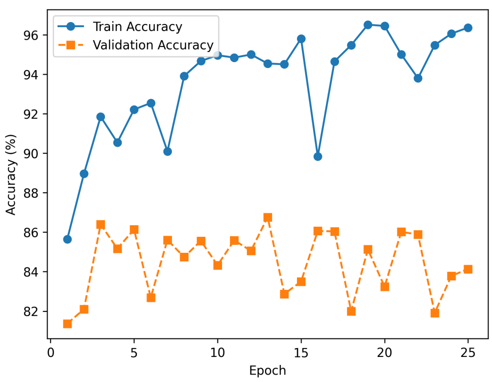
Learning Rate	Accuracy	Precision	Recall	F1-score
10^{-2}	0.8016	0.8380	0.8016	0.7961
10^{-3}	0.7646	0.8114	0.7645	0.7554
10^{-4}	0.7957	0.8068	0.7957	0.7939
10^{-5}	0.7748	0.7794	0.7748	0.7739

Table 31: Performance of Custom CNN on Test Data

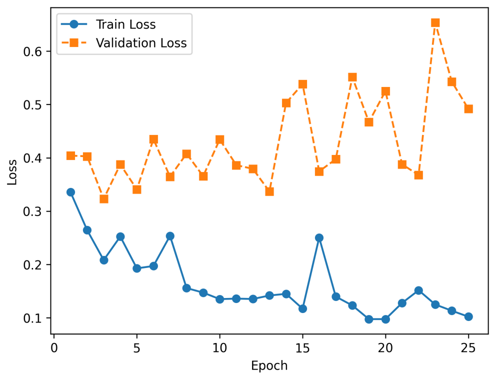
Learning Rate	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
10^{-2}	94.97	96.45	96.37	84.33	83.24	84.13
10^{-3}	80.37	94.21	94.79	71.91	86.46	84.20
10^{-4}	82.99	89.52	89.90	80.06	80.88	79.11
10^{-5}	81.27	83.25	84.04	77.69	79.63	80.41

Table 32: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§3.2 Plots

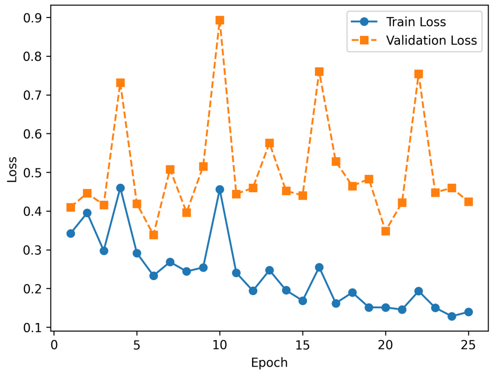


(a) Accuracy vs Epochs



(b) Loss vs Epochs

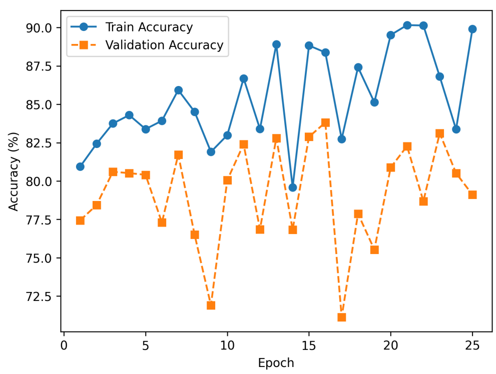
Figure 40: Custom CNN Model with learning rate = 10^{-2}



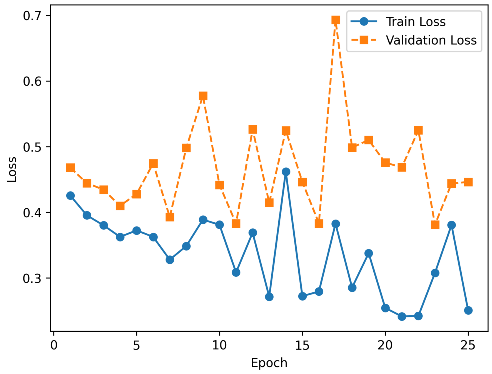
(a) Accuracy vs Epochs

(b) Loss vs Epochs

Figure 41: Custom CNN Model with learning rate = 10^{-3}

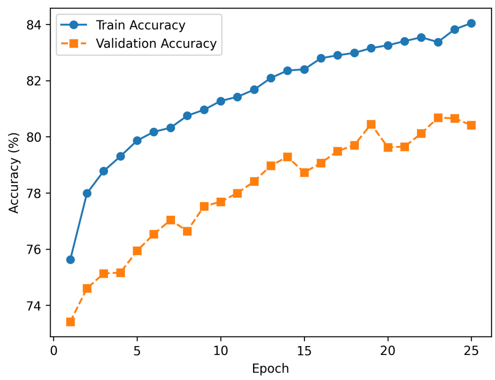


(a) Accuracy vs Epochs

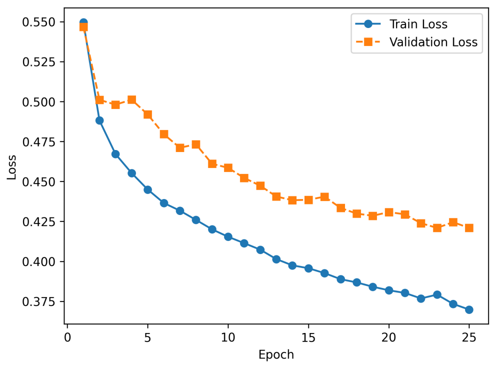


(b) Loss vs Epochs

Figure 42: Custom CNN Model with learning rate = 10^{-4}



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 43: Custom CNN Model with learning rate = 10^{-5}

§4 Data Augmentation

§4.1 Results on Test Data

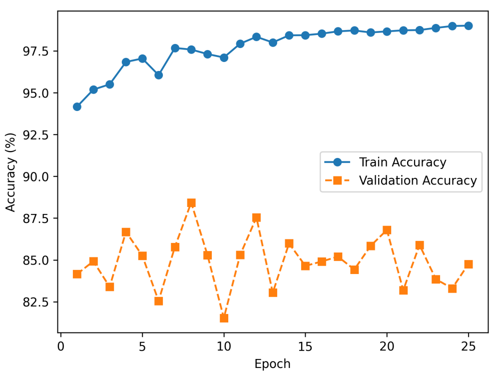
Data Augmentation	Accuracy	Precision	Recall	F1-score
False	0.7646	0.8114	0.7645	0.7554
True	0.8215	0.8464	0.8215	0.8183

Table 33: Performance of Custom CNN on Test Data

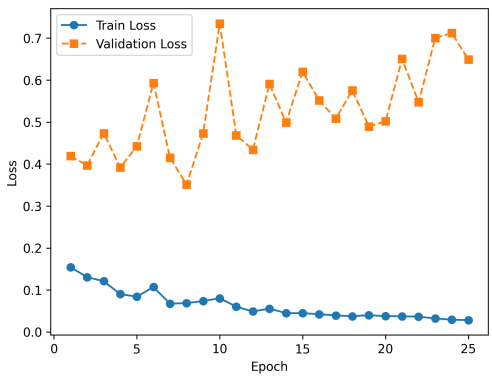
Data Augmentation	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
False	80.37	94.21	94.79	71.91	86.46	84.20
True	83.12	88.91	89.86	75.76	85.28	82.69

Table 34: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§4.2 Plots

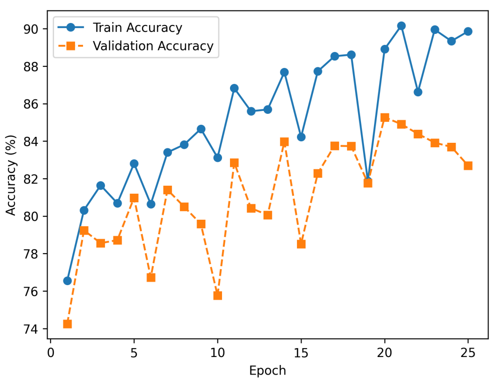


(a) Accuracy vs Epochs

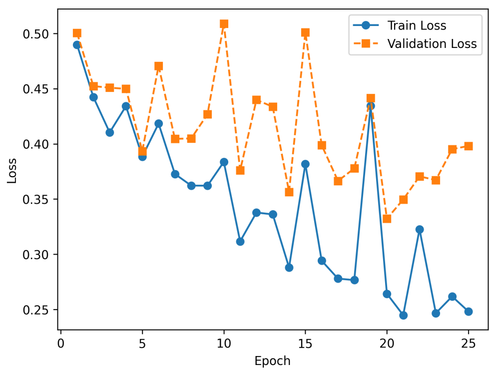


(b) Loss vs Epochs

Figure 44: Custom CNN Model without Data Augmentation



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 45: Custom CNN Model with Data Augmentation

§5 Optimizer Comparision

§5.1 Results on Test Data

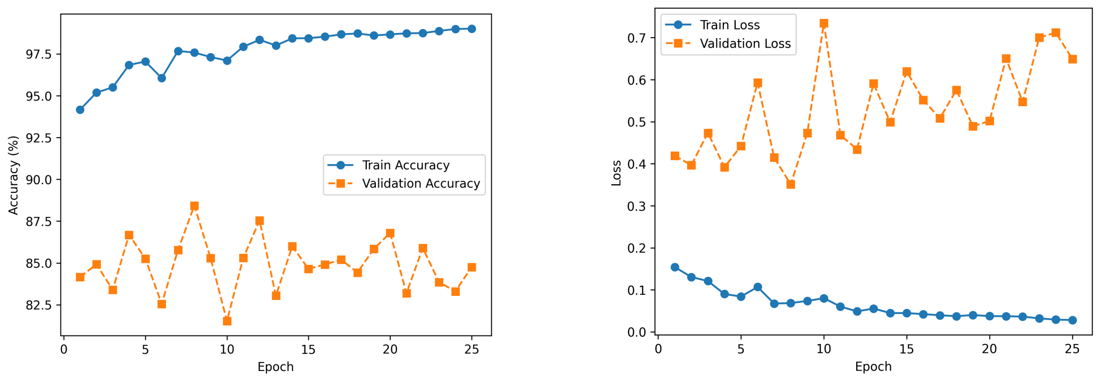
Optimizer	Accuracy	Precision	Recall	F1-score
SGD	0.7646	0.8114	0.7645	0.7554
Adam	0.7825	0.8190	0.7824	0.7760

Table 35: Performance of Custom CNN on Test Data

Optimizer	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
SGD	80.37	94.21	94.79	71.91	86.46	84.20
Adam	94.37	96.05	97.10	77.73	78.18	83.10

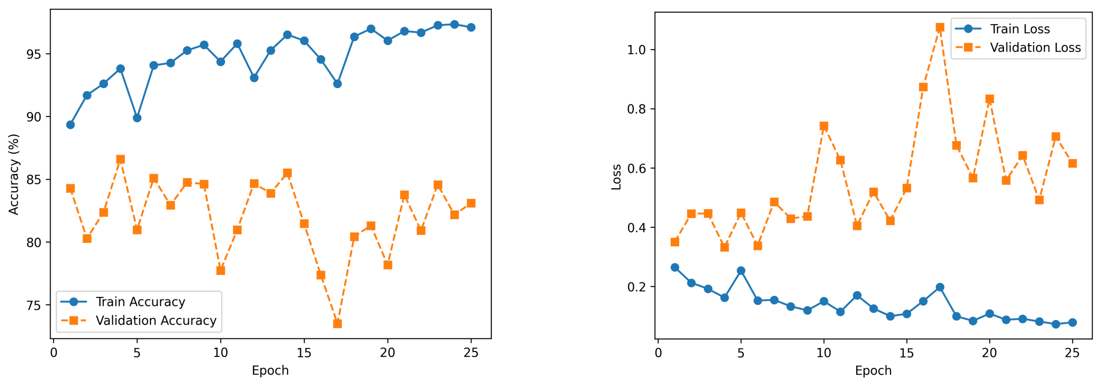
Table 36: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§5.2 Plots



(a) Accuracy vs Epochs (b) Loss vs Epochs

Figure 46: Custom CNN Model with SGD Optimizer with Momentum



(a) Accuracy vs Epochs (b) Loss vs Epochs

Figure 47: Custom CNN Model with Adam Optimizer

§6 Observations and Analysis

Learning Rate

- For the custom CNN, learning rate 10^{-4} achieved the highest test accuracy (79.57%), while 10^{-2} caused instability despite higher training performance.
- The default learning rate 10^{-3} resulted in suboptimal test accuracy (76.46%), indicating sensitivity to rate selection.

Data Augmentation

- Data augmentation improved test accuracy from $76.46\% to 82.15\%$, demonstrating its critical role in mitigating overfitting.
- Validation accuracy stabilized at 82.69% with augmentation, reflecting better generalization.

Optimizer Comparison

- Adam (78.25%) outperformed SGD (76.46%), contrasting with trends in ResNet/VGG, likely due to adaptive learning rates suiting the custom architecture.
- SGD provided stable validation accuracy (84.20%) but slower convergence.

Best Performing Model

Achieved 82.15% test accuracy with learning rate: 10^{-3} and data augmentation enabled

§7 Comparision with Part 1

Aspect	Part 1 (ResNet/VGG)	Part 2 (Custom CNN)
Learning Rate	- ResNet18: Best test accuracy (81.38%) with 10^{-5}. - VGG16: Best test accuracy (86.25%) with 10^{-4}. Larger rates (10^{-2}) caused instability in both models.	- Custom CNN: Best test accuracy (79.57%) with 10^{-4}. Larger rates (10^{-2}) caused instability despite better training performance.
Data Augmentation	- ResNet18: Test accuracy improved from 81.25% $\rightarrow$ 82.60%. - VGG16: Test accuracy improved from 80.69% $\rightarrow$ 82.11%. Augmentation reduced overfitting and stabilized validation accuracy.	- Custom CNN: Test accuracy improved from 76.46% $\rightarrow$ 82.15%. Validation accuracy stabilized at 82.69%, showing significant impact on generalization.
Optimizer	- ResNet18: SGD outperformed Adam (81.25% vs. 77.95%). - VGG16: SGD significantly outperformed Adam (80.69% vs. 50.02%), as Adam caused severe overfitting in deeper networks.	- Custom CNN: Adam outperformed SGD (78.25% vs. 76.46%), likely due to adaptive learning rates aiding convergence in the custom architecture.

Part III
Model Improvement Competition

§1 Overview

In this section, we improve the performance of our best-performing model from Part 1 or the custom architecture from Part 2. The goal is to achieve the highest possible accuracy on the PatchCamelyon (PCam) dataset by applying various enhancement techniques. These include data augmentation, regularization, optimizer tuning, learning rate scheduling, and minor architectural modifications.

§2 Model Description

- Base Model: VGG16 item Optimizer: Stochastic Gradient Descent (SGD) with momentum (0.9).
- Learning Rate: Initially set to 10^{-4}.
- Scheduler: StepLR with a decay factor of 0.1 every 10 epochs.
- Data Augmentation
  - Random horizontal and vertical flips
  - Rotation (±15)
  - Color jittering (brightness, contrast, saturation, hue adjustments)

  Size: 32
- Loss Function: Cross-Entropy Loss

§3 Results on Test Data

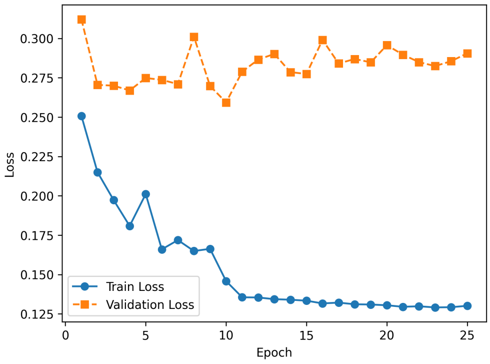
Best Models	Accuracy	Precision	Recall	F1-score
Part 1	0.8625	0.8823	0.8625	0.8607
Part 2	0.8215	0.8464	0.8215	0.8183
Part 3	0.8864	0.8952	0.8864	0.8858

Table 37: Performance of Best Models from all parts on Test Data

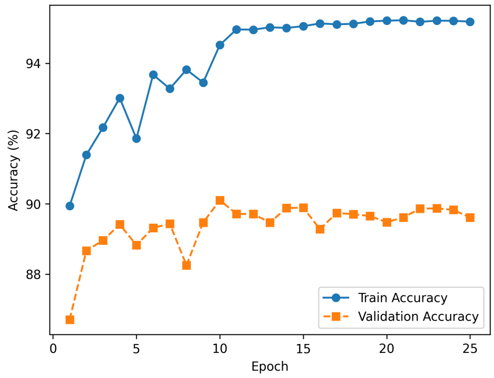
Best Model	Train Accuracy			Val Accuracy		
	10th	20th	25th	10th	20th	25th
Part 1	96.25	97.58	97.70	89.73	89.05	89.24
Part 2	83.12	88.91	89.86	75.76	85.28	82.69
Part 3	94.52	95.21	95.18	90.10	89.48	89.61

Table 38: Train and Validation Accuracies at 10th, 20th, and 25th Epochs

§3.1 Plots



(a) Accuracy vs Epochs



(b) Loss vs Epochs

Figure 48: VGG16 model with customized hyperparameters

Google Drive Link

Best Models and their Train/Val Loss and Accuracy over epochs