Optimizing CNN Models for COVID-19 Detection: A Comparative Analysis of Optimizers and Loss Functions

Tianshu Li, Huipeng Lei, Yupeng Chang
University of Toronto
jasonts.li@*, huipeng.lei@*, yupeng.chang@*, *=mail.utoronto.ca

Abstract

The growing significance of Convolutional Neural Networks (CNNs) in various real-world medical applications, including COVID-19 detection, has been widely recognized. In this research, we propose an improved approach by refining the architecture of three state-of-the-art CNN models for COVID-19 detection, utilizing lung X-ray images. Additionally, we investigate the impact of employing different optimizers and loss functions on the model's performance.

1 Introduction

The outbreak of COVID-19 has caused a global health crisis, making early detection of infection crucial. With the rapid development of effective machine learning methodologies, we propose using Convolutional Neural Networks (CNN) to classify patients as COVID or Non-COVID based on lung X-ray images. CNNs are ideal for image tasks due to their ability to capture local information, reduce complexity, and handle data augmentation. We will explore and improve upon three state-of-the-art CNN models (ResNet-152 [4], COVID-Net [11], and Shallow CNN [8]) by changing their architectures, optimizers, and loss functions. With efficient architectures like Small COVID-Net and high-performing architectures like Modified ResNet-152, our results could both increase usage scalability and efficiency as well as improve diagnosis and treatment efficacy for medical professionals.

2 Related Works

The journal by Sneha Kugunavar and C.J. Prabhakar[6] lists related works on implementing CNN models for COVID-19 classification using chest CTs and X-rays. We re-implemented three methods: one proposed by Rahul Kumar et al.[7] using ResNet-152 and SMOTE algorithm with various classifiers, one using Covid-net[11] which leverages selective long-range connectivity and architectural diversity, and another using Shallow CNN that is very efficient [8].

In the paper of Rahul Kumar et al., they used ResNet-152 with Adam optimizer to extract 1024 features from COVID-19 X-ray images, and addressed imbalanced data using the SMOTE algorithm. The transformed features were input to machine learning classifiers achieving 97.3% accuracy with Random Forest. Another paper we focused on is COVID-Net[11] by Wang et al., which uses tailored deep CNN design for COVID-19 detection from chest X-ray images, emphasizing selective long-range connectivity and architectural diversity. In addition, we compared Mukherjee et al.'s shallow CNN[8] with the two above methods to evaluate trade-offs between computational efficiency and classification performance in different CNN architectures for COVID-19 detection from chest X-rays.

3 Dataset

We used one particular dataset across all three of the proposed models, which is publicly available on Kaggle: COVIDx CXR-2[1], which will be referred to as COVIDx. This dataset is comprised of a total of 13,975 CXR images across 13,870 patient cases. According to COVIDx's creators, they combined and modified five different datasets[11]: (1) COVID-19 Image Data Collection[3], (2) COVID-19 Chest X-ray Dataset Initiative[2], (3) ActualMed COVID-19 Chest X-ray Dataset Initiative[1], (4) RSNA Pneumonia Detection Challenge dataset[10], and (5) COVID-19 radiography database[9]. Every image in the dataset is labelled with a positive or negative label for the purpose of supervised learning.

4 Method

4.1 Modified Resnet-152

Data Pre-processing

It could be seen that the X-ray images are in different shapes and some have low resolutions/qualities, so image cropping and resizing to the data is deemed necessary for performance. First, each image is transformed to grayscale and resized to a fixed size of 224x224 pixels

[1]https://www.kaggle.com/datasets/andyczhao/covidx-cxr2

with 3 output channels. Then, the input images are normalized because normalizing images before feeding them into a CNN can improve the performance of the network[5].

Architecture (Fig 1)

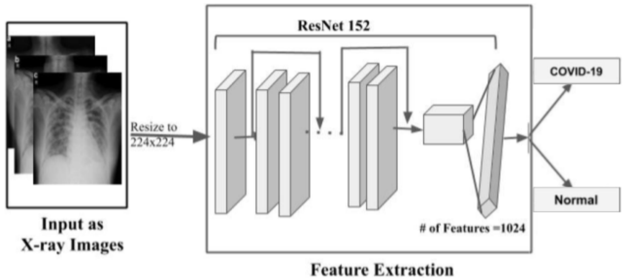


Figure 1: The proposed Resnet-152 model algorithm block diagram for COVID-19 Classification

Our model differs from previous approaches by not utilizing SMOTE or any additional ML classifiers such as Logistic Regression, KNN, RF, or XGBoost. Instead, our approach utilizes the ModifiedResNet152 architecture, a modified version of the ResNet152 backbone pre-trained on ImageNet. The last fully connected layer of ResNet152 is replaced with a new fully connected layer containing 1024 hidden units and a ReLU activation function, followed by a final fully connected layer with a log softmax activation function that outputs predicted class probabilities. The network takes input images, applies a series of convolutional, pooling, and activation layers, and uses adaptive average pooling to reduce the output spatial dimensions to (1,1) before passing the result through the fully connected layer.

ResNet-152[4] is known to be a superior architecture, but it has high time complexity due to its large number of layers. However, using a bottleneck design can reduce the complexity. Additionally, the problem of vanishing gradients is resolved using skip connections in the network.

Experimental Setup: ModifiedResnet is implemented using Python 3.9 with an Intel® Core i7-12700kF CPU @3.60 GHz running on Windows 11 with NVIDIA Geforce RTX3060ti with 8GB graphics.

4.2 Small COVID-Net

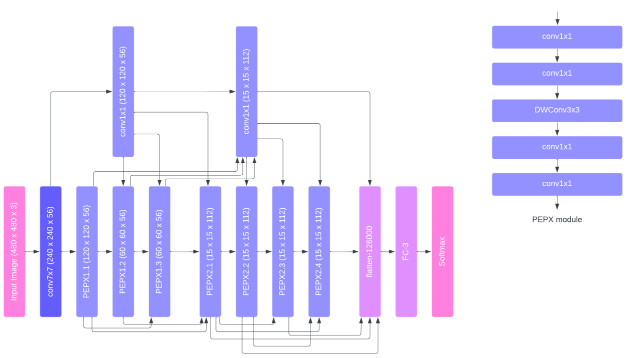


Figure 2: Small COVID-Net

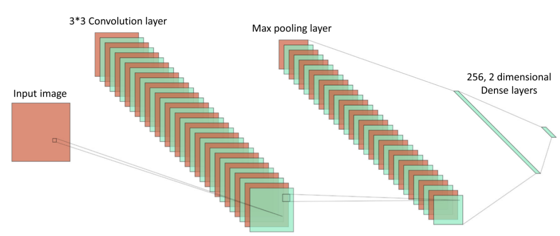


Figure 3: Shallow CNN[8]

Data Pre-processing

First, the input image is converted to grayscale while keeping the number of output channels at 3 and resized to a fixed size of 480x480 pixels. Next, the resized grayscale image is converted to a PyTorch tensor and is normalized using specified mean and standard deviation values. This preprocessing procedure ensures that the input images are of the same size, converted to the appropriate data type for CNNs, and normalized for better training performance.

Architecture (Fig 2)

Small COVID-Net is a simplified variant of COVID-Net proposed in [11] (Fig 6), employing the same PEPX modules with reduced layers, resulting in only a quarter of the parameters. This allows it to be computed on a single 16-GB Nvidia T4 GPU on Kaggle, which is cost and time efficient. The aim is to make it practical and scalable for everyday medical use while still achieving accurate results.

4.3 Shallow Convolutional Neural Network (Fig 3)

The ShallowCNN consists of a single convolutional layer, followed by a max-pooling layer, and two fully connected layers. The model architecture is designed to minimize the number of parameters while maintaining high performance in terms of accuracy and sensitivity. We resize the image to 50×50 grayscale input for the network.

5 Experiments and Results

Evaluation Criteria

To evaluate the efficacy of the models, we evaluated the models with the necessary metrics. The classification of COVID-19 patients and Normal is termed as **Accuracy**, **Sensitivity**, **Specificity**, and **$\mathbf{F_1}$-Score**, necessary metrics are calculated and recorded in tables: ResNet-152 Table 1, COVID-Net Table 2, and Shallow CNN Table 3.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1) \qquad Specificity = \frac{TN}{TN+FP} \quad (3)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (2) \qquad F_1 - score = \frac{2*TP}{2*TP+FP+FN} \quad (4)$$

Figure 4: Formula for calculating the metrics, where TP, TN, FP, and FN are True Positive, True Negative, False Positive, and False Negative, respectively.

5.1 Modified ResNet-152

Table 1: Evaluation metrics of the Finetuned Modified Resnet-152 model
hyperparameters: {learning rate=2e-4, batch size=16, epochs=10}

Metric	Accuracy	Sensitivity	Specificity	F_1-Score
ADAM+CE	0.9675	0.935	1.00	0.966
ADAM+NLLLoss	0.9575	0.915	1.00	0.956
ADAM+MSE	0.9675	0.935	1.00	0.966
RMSprop+CE	**0.9800**	**0.960**	**1.00**	**0.980**
SGD+CE	0.9475	0.955	1.00	0.977
AdaDelta+CE	0.9775	0.895	1.00	0.945
AdamW+CE	0.9500	0.900	1.00	0.947
No Pretrained (Epochs=22)	0.9100	x	x	x

The Modified ResNet-152 model was evaluated using various optimization algorithms and loss functions, and the best performance was achieved with **RMSprop** and **Cross-Entropy (CE)** loss (Fig**??**), resulting in an accuracy of **98%** (Table 1), higher than the SMOTE+Resnet152+MLClassifiers[7] model proposed by Kumar. Notably, our model achieved better results despite not using SMOTE or integrating any ML classifiers. The model had high sensitivity and specificity scores and a good F_1-score, indicating its ability to correctly identify positive and negative cases of COVID-19 from X-ray images. All models had perfect specificity scores, indicating the model is able to classify negative cases perfectly. The appropriate combination of hyperparameters, optimization algorithm, and loss function is crucial for achieving optimal performance.

RMSprop (Root Mean Square Propagation) is an optimization algorithm used in training neural networks. It was proposed by Geoffrey Hinton in 2012[2]. RMSprop uses an adaptive learning rate for each parameter, which means that it adjusts the learning rate based on the historical gradient information for each parameter.

Overall, the Modified ResNet-152 model with RMSprop and CE loss proved to be effective in detecting COVID-19 from X-ray images. Moreover, during our experiment, the best accuracy achieved for the Modified ResNet-152 model **without** pre-trained weight is 91%, significantly lower than the model with ImageNet-pretrained weights.

5.2 Small COVID-Net

Table 2: Evaluation metrics of Small COVID-Net
hyperparameters: {learning rate=2e-4, batch size=64, epochs=10}

Metric	Accuracy	Sensitivity	Specificity
ADAM+NLLLoss	0.6675	0.525	0.810
ADAM+MSE	0.8350	0.725	0.945
SGD+CE	0.6550	0.525	0.785
ADAM+CE	0.8575	0.755	0.960
AdamW+CE	0.8600	0.730	0.990
AdamW+CE (epochs=34)	**0.8950**	**0.825**	**0.965**

The best-performing combination of optimizer and loss function for Small COVID-Net was found to be **AdamW** optimizer with **Cross-Entropy (CE)** loss, resulting in an accuracy of **89.5%** (Table 2). AdamW is a variant of the Adam optimizer that uses a "decoupled weight decay" approach, applying weight decay directly to the parameters during the update step, separately from the gradient. It appears that the weight decay in AdamW leads to better generalization compared to Adam for Small COVID-Net.

When using other loss functions, NLLLoss showed an unstable training curve with around 80% accuracy (Fig 7e), while MSE slightly under-performed CE presumably because it is typically not suitable for discrete classification tasks. Among other optimizers, the model converged slowly under SGD (Fig 7f), and it appeared that adding more epochs to SGD might improve performance. However, AdamW proved to be a better alternative as it converged much faster, requiring fewer epochs and resources (Fig 7b).

[2] http://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf

Although the accuracy of our model is lower than the COVID-Net model (Fig 6) proposed by Wang et al. [11], our model is much more compact in size and trainable on only one 16GB GPU. With one-quarter of the convolutional parameters compared to the original COVID-Net model (Fig 6), our model enables faster propagation through convolutional layers and reduces the complexity of the fully connected layer. This makes the method more feasible for resource-scarce areas such as rural clinics, where access to high computational resources may be limited.

5.3 Shallow CNN

Table 3: Evaluation metrics of Shallow CNN
hyperparameters: {learning rate=0.005, batch size=25, epochs=100}

Metric	Accuracy	Sensitivity	Specificity
SGD+NLLLoss(32 filter)	0.8350	0.730	0.940
SGD+MSE(32 filter)	0.8850	0.795	0.975
SGD+CE(32 filter)	**0.8900**	**0.820**	**0.960**
SGD+CE(16 filter)	0.8675	0.760	0.975
SGD+CE(8 filter)	0.850	0.750	0.950
SGD+CE(4 filter)	0.8500	0.775	0.925
SGD+CE(2 filter)	0.8375	0.750	0.925

In the Shallow CNN experiment, it was observed that the model with Mean Squared Error (MSE) loss achieved higher accuracy compared to the model with Negative Log Likelihood (NLLLoss) loss. A possible explanation for this difference is the distinct label representations required by these two loss functions. MSE loss uses one-hot encoded labels, which may provide a more informative gradient for the model to learn from, whereas NLLLoss relies on class indices. Consequently, the one-hot encoded labels in the MSE loss model may have contributed to better generalization and improved accuracy. It is crucial, however, to consider the specific context of the experiment and dataset, as the impact of label representation may vary across different tasks and data distributions.

As the number of filters decreases from 32 to 2, a clear trend of reduced accuracy is observed, and it is also noticeable that the validation loss curve becomes less smooth, as seen in Fig 9. This can be explained by increasing the number of filters in a convolutional layer allows the model to learn more complex and diverse features from the input images. When the number of filters is reduced, the model's capacity to learn complex features is limited, leading to decreased performance. For example, with 32 filters, the model can learn a broader set of features, which helps it generalize better and achieve higher accuracy than models with fewer filters.

5.4 Model Comparison

When evaluating all the models, it is evident that specificity is generally higher than sensitivity, indicating proficient identification of true negative cases. However, there may be instances where positive cases are misclassified as negative, potentially due to limitations such as dataset scarcity or model complexity. Little false positives were observed, meaning patients are very unlikely to be falsely diagnosed as having COVID-19, which prevents panic from false disease diagnosis.

Upon examining all three models, we considered the trade-off between model complexity and accuracy. The Finetuned Modified ResNet-152 achieved the highest accuracy at 98%, but required a large number of parameters during training, making pretraining critical. In contrast, Small COVID-Net and Shallow CNN had significantly fewer parameters, approximately 1-2% of ResNet-152, yet still achieved decent accuracies of around 89%. This suggests that improving model performance at later stages is challenging as more layers are needed to process all important pixels and patterns.

Regarding loss functions, Cross Entropy performed the best, which is expected for binary classification problems. Interestingly, MSE (Mean Squared Error) yielded good results for all three models, despite being designed mostly for regression problems.

It is noteworthy that the optimal optimizer differed for each model: Modified ResNet with RMSProp, small COVID-Net with AdamW, and Shallow CNN with SGD. Therefore, it is crucial to test out a few optimizers on any model to find the best fit.

We also found that using pre-trained weights for Convolutional Neural Network models is crucial and can significantly improve model accuracy. The fine-tuned Modified ResNet-152 achieved the highest accuracy of 98%, compared to 91% without pre-trained weights. This is because Modified ResNet-152 has pre-trained weights available from PyTorch, while Small COVID-Net and Shallow CNN do not. However, if we were to pre-train Small COVID-Net and Shallow CNN on large datasets like ImageNet, we believe their performance would improve as well. Unfortunately, we did not have the resources to pre-train them in this study.

6 Conclusion

In conclusion, our study highlights the importance of considering different optimization algorithms and loss functions when training machine learning models for COVID-19 detection. Our results indicate that pre-training the models on large datasets like ImageNet and fine-tuning them on the target task can significantly improve their accuracy. Moreover, we found that the trade-off between model complexity and accuracy must be carefully considered, especially in cases where computing resources are limited. Overall, this work provides useful insights into the development of machine-learning models for COVID-19 detection and can aid researchers and practitioners in selecting appropriate optimization strategies and model architectures. Additionally, our models could potentially be applied for transfer learning to detect other lung-related diseases.

Code Repository: https://github.com/fmt47/413_project

References

[1] A Chung. Actualmed covid-19 chest x-ray data initiative. 2020.

[2] A Chung. Covid-19 chest x-ray data initiative. 2020.

[3] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. June 2016.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[6] Sneha Kugunavar and C. Prabhakar. Convolutional neural networks for the diagnosis and prognosis of the coronavirus disease pandemic. *medRxiv*, 2021.

[7] Rahul Kumar, Ridhi Arora, Vipul Bansal, Vinodh J Sahayasheela, Himanshu Buckchash, Javed Imran, Narayanan Narayanan, Ganesh N Pandian, and Balasubramanian Raman. Accurate prediction of covid-19 using chest x-ray images through deep feature learning model with smote and machine learning classifiers. *medRxiv*, 2020.

[8] Himadri Mukherjee, Subhankar Ghosh, Ankita Dhar, Sk Md Obaidullah, K C Santosh, and Kaushik Roy. Shallow convolutional neural network for covid-19 outbreak screening using chest x-rays. *Cognitive Computation*, pages 1–14, 2021.

[9] Radiological Society of North America. Covid-19 radiography database. 2019.

[10] Radiological Society of North America. Rsna pneumonia detection challenge. 2019.

[11] Linda Wang, Zhong Qiu Lin, and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest X-ray images. *Scientific Reports*, 10(1):19549, November 2020.

Appendix

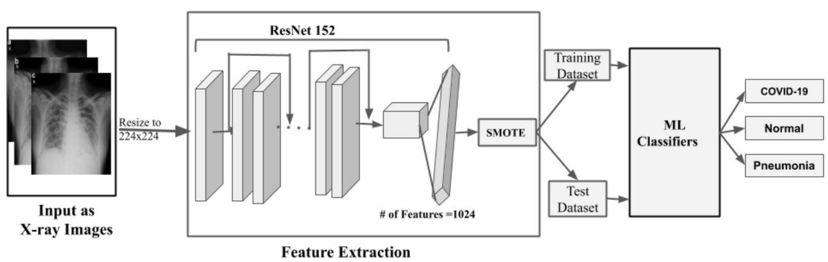


Figure 5: Resnet152 integrated with SMOTE and ML classifiers by Kumar[7]

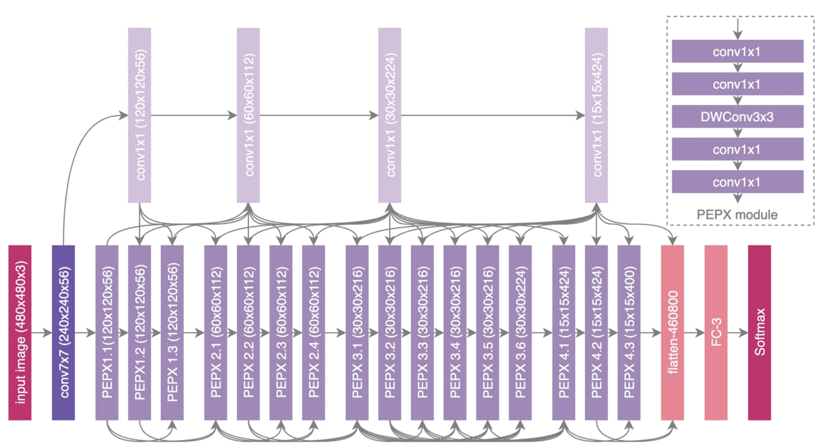
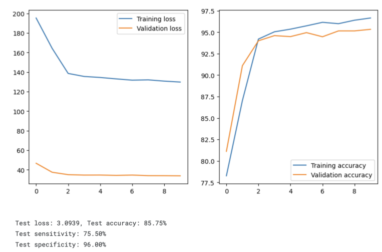
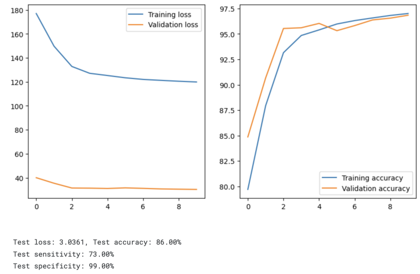


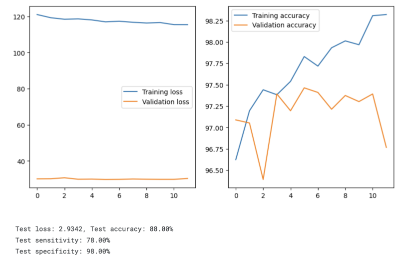
Figure 6: Original Covid-Net [11]



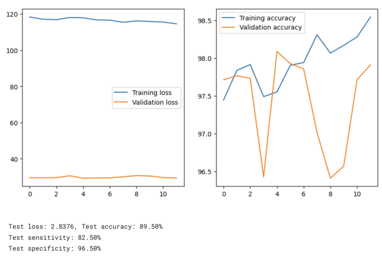
(a) Adam+CE



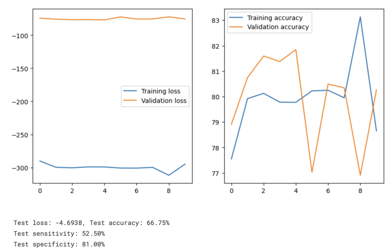
(b) AdamW+CE Epoch 1 - 10



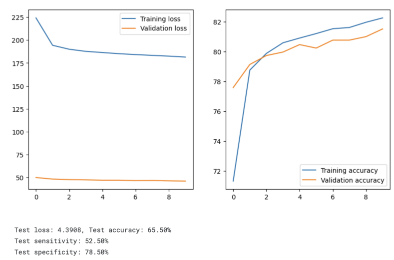
(c) AdamW+CE Epoch 11 - 22



(d) AdamW+CE Epoch 23 - 34

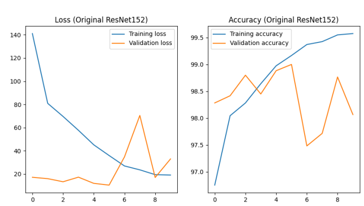


(e) Adam+NLLLoss

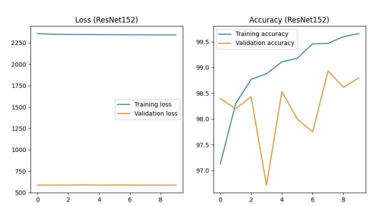


(f) SGD+CE

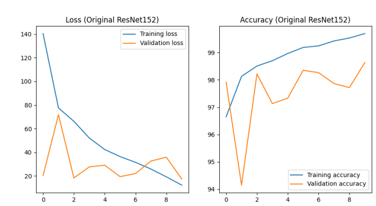
Figure 7: Small COVID-Net training curves



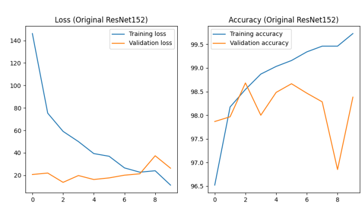
(a) Adam+CE



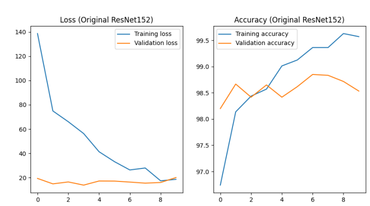
(b) Adam+MSE



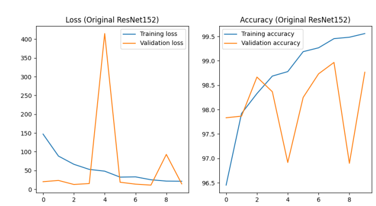
(c) Adam+NLLL



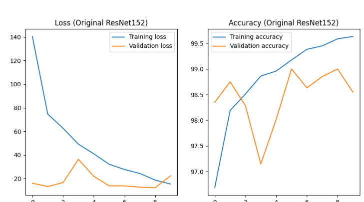
(d) AdaDelta+CE



(e) SGD+CE



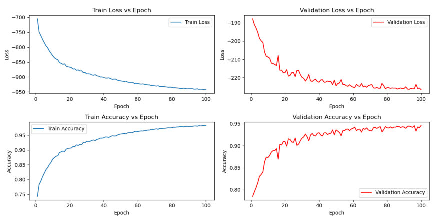
(f) RMSProp+CE



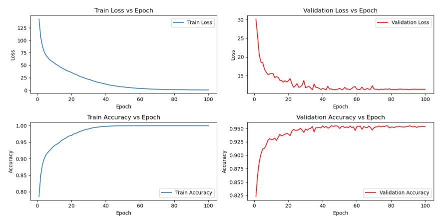
(g) AdamW+CE

(h) Without Pre-trained

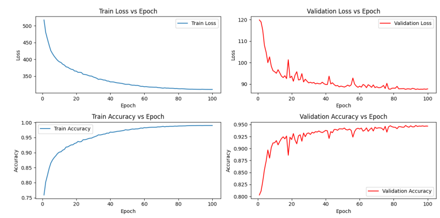
Figure 8: Modified ResNet152 training curves



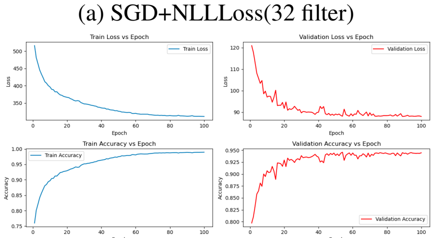
(a) SGD+NLLLoss(32 filter)



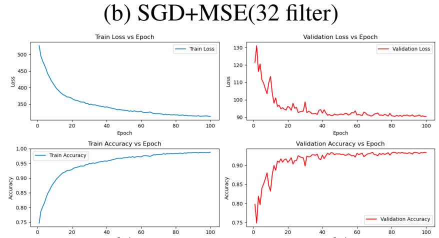
(b) SGD+MSE(32 filter)



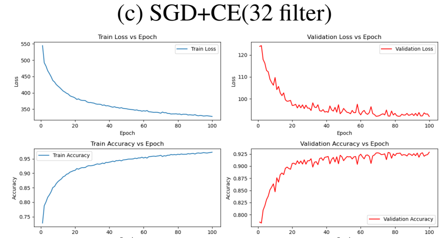
(c) SGD+CE(32 filter)



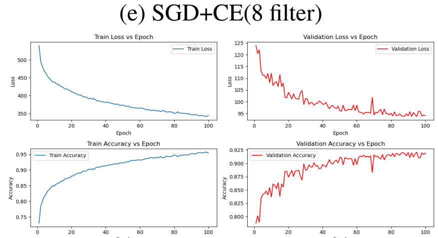
(d) SGD+CE(16 filter)



(e) SGD+CE(8 filter)



(f) SGD+CE(4 filter)



(g) SGD+CE(2 filter)

Figure 9: Shallow CNN training curves