ELEC5307 Project 2 - Fruit Classification Challenge

Group 14 - Team members: Zeeshan Ansari (510370813), Syed Hamza Kaliyadan (500585454)

Section 1: Introduction **1**
Section 2: Background **1**
Section 3: Methods and Experiments **1**
3.1 Dataset Splitting and Preprocessing 1
3.2 Training and Evaluation Process 2
3.3 Results and Observations 3
3.4 Novel Pipelines, Networks, Modules, and Tricks 4
Section 4: Discussion and Conclusion **4**
4.1 Model Performance 4
4.3 Challenges and Limitations 5
4.4 Future Enhancements and Research Directions 5
4.5 Conclusion 5
Appendix **6**
References **15**

Section 1: Introduction

Recent developments in deep learning have transformed image classification problems and produced reliable results in a variety of domains. The goal of this research is to create a deep learning-based fruit classification model that can reliably identify different kinds of fruit from pictures. There are several uses for efficient fruit categorisation, including quality control in food processing, automated sorting in agriculture, and even helping customers recognise fruits in stores.

This project's main goal is to use cutting-edge neural network architectures to design, build, and assess a reliable classification pipeline. Obtaining high accuracy on a varied dataset involves optimising the training process and making adjustments to the model's architecture. In order to improve model resilience throughout this research, data pretreatment and augmentation are essential. To replicate a variety of circumstances and variations, techniques including colour jitter, random rotations, and horizontal flips may have to be used on the training images.

The report showcases a comparative examination of several models and training methodologies while providing specifics on the project's approach, experiments, and findings. The goal being to come up with an entire workable pipeline that is both accurate and efficient while utilising useful novel elements.

Section 2: Background

Conventional techniques used machine learning algorithms such as K-Nearest Neighbours (KNN) and Support Vector Machines (SVM), which necessitated the human extraction of features (such as colour and texture). Significant gains in classification accuracy and resilience have been made possible by the development of deep learning, namely Convolutional Neural Networks (CNNs).

With accuracies of over 99% in clear settings and 96.75% on difficult photos, the study "Efficient Fruit Classification Framework Using Deep Learning" showed how successful transfer learning is for classifying fruits [1]. A hybrid CNN-RNN-LSTM model was suggested in "Fruit Image Classification Using Deep Learning" [2] that improved classification in low visibility situations and outperformed conventional techniques. When comparing the VGG16 and ResNet50 architectures, "Recognition and Classification of Fruits Using Deep Learning Techniques" found that ResNet50 was better for complex, enhanced datasets with good accuracy across 131 fruit categories, although requiring more processing power [3]. The study "Automatic Fruit Detection System Using Multilayer Deep Convolution Neural Network" highlighted CNN's capacity to handle intricate visual features and the significance of pooling for computational efficiency by developing a CNN-based system that achieved 97.4% accuracy in identifying multiple fruit types under various conditions [4].

These studies demonstrate how deep learning, particularly CNN-based models incorporating transfer learning, can improve the accuracy of fruit classification. Nonetheless, there are still issues with computing effectiveness and managing various visual circumstances, which direct this project's strategy to concentrate on data augmentation and model robustness.

Section 3: Methods and Experiments

3.1 Dataset Splitting and Preprocessing

The dataset (2857 images) had 22 classes (each being a fruit). It was gathered through students uploading pictures of the fruits in various locations and angles etc., in order to introduce variety into the model. Eighty percent of the original dataset was used for training, while twenty percent was used for validation. This division preserved a distinct validation set to track model performance while guaranteeing a sizable training set. The following modifications were used for data preprocessing which include **Data Augmentation** techniques:

- **Training Set:** After resizing the images to 224 by 224 pixels, they underwent normalisation, colour jittering (brightness, contrast, saturation, and hue), random horizontal flips, and rotations of up to one degree.
- **Validation and Test Set:** To guarantee consistent evaluation, images were resized to 224 by 224 and normalised without undergoing arbitrary modifications.

3.2 Training and Evaluation Process

The AdamW optimizer was used to train each model for 20 epochs with a learning rate of 0.001, weight decay of 0.0001, and amsgrad enabled. To facilitate convergence, the StepLR scheduler lowered the learning rate by a factor of 0.1 per ten epochs. The optimal model weights were saved depending on the validation accuracy, which was determined at the end of each epoch.

The following is an outline of how the best model was evaluated through experimentation:

1. Testing several baseline networks (pretrained on ImageNet)
2. Choosing the best network from step 1, testing various learning rate schedulers to refine training
3. Choosing the best setting from step 2, testing various gamma values for the scheduler to finally obtain the best network.

Table 1 summarises the results, comparing validation accuracy, test accuracy, and inference time across all models. The inference times below are from a NVIDIA GeForce GTX 1050 Ti machine.

Model	Validation Accuracy (%)	Test Accuracy (%)	Average Inference Time (s)	Appendix
ResNet50	92	99.15	26.33	Figure 4
DenseNet	95	99.53	28.82	Figure 5
EfficientNet	93	99.26	20.64	Figure 6
MobileNetV3	95	99.59	20.37	Figure 7
AlexNet	91	99.43	20.37	Figure 8
SqueezeNet	86	97.5	19.17	Figure 9
ViT Transformer	97	99.429	122.28	Figure 10

ConvNeXt	96.95	99.27	44.79	Figure 11

Table 1: Comparison of various baseline networks (all pretrained)

With a test accuracy of 99.59% and an inference time of 0.0068 seconds per image, MobileNetV3 struck the optimum balance between accuracy and computational efficiency. While the ViT transformer outperforms it, it has a disproportionately higher inference time as was therefore not chosen.

Three learning rate schedulers were experimented with (Step, MultiStep and Exponential) as shown in table 2, the original Step scheduler was retained due to slightly better performance.

Scheduler	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Appendix**
Lr_scheduler: Step	95	99.59	Figure 12
Lr_scheduler: MultiStep	94	99.22	Figure 13
Lr_scheduler: Exponential	95	99.53	Figure 14

Table 2: Comparison of various scheduler performances

Three scheduler gamma rates were then experimented with using the Step scheduler (0.05, 0.1 and 0.2). While these do not differ much in their results, the 0.1 rate seemed to provide the most opportunity to get a higher validation accuracy.

Scheduler Gamma Rates	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Appendix**
0.05	94	99.56	Figure 15
0.1	95	99.59	Figure 16
0.2	94	99.63	Figure 17

Table 3: Comparison of various scheduler gamma values

3.3 Results and Observations

The following figures (see Appendix) show the training and evaluation outcomes for MobileNetV3, our top-performing model:

- Training and validation loss are displayed in Figure 1: Loss Curve for Epochs. A sharp drop in loss over the first few epochs suggests successful learning. By epoch 10, the model had stabilised its loss levels and converged.

- The precision increases over epochs, reaching near-perfect precision by the end of training, as shown in Figure 2: Precision Curve for Epochs.
- Figure 3: Validation and Training The Precision Curve, which shows the tight alignment between the two curves and indicates effective generalisation, shows the model's precision for both training and validation sets. After epoch 10, the validation precision stabilises and reflects the training precision curve. While slight overfitting may be noticed, it is not significant.

These findings validate MobileNetV3's applicability for real-world applications needing quick and precise fruit categorisation by highlighting its great generalisation and processing economy.

3.4 Novel Pipelines, Networks, Modules, and Tricks

We experimented with a variety of sophisticated neural network designs and integrated several strategies to improve model performance in order to attain the best possible accuracy and efficiency for fruit classification. The new methods and improvements made to our pipeline are listed below:

1. **Transfer Learning with Diverse Architectures**: We assessed a number of pre-trained models, such as Vision Transformer (ViT), AlexNet, SqueezeNet, MobileNetV3, DenseNet, EfficientNet, and ResNet50. These models offered a foundation of strong feature representations being pre-trained on extensive datasets (ImageNet).
2. **Efficient Feature Extraction with MobileNetV3**: By striking a balance between computational efficiency and accuracy, MobileNetV3 proved to be the most effective model. It is especially well-suited for deployment in resource-constrained contexts due to its lightweight architecture and depth wise separable convolutions, which are optimised for real-time applications.
3. **Layer Freezing for Stability**: The first feature extraction layers were frozen during training for models such as MobileNetV3. This method reduced overfitting and accelerated training by fine-tuning only the classification layers while maintaining the valuable pre-trained features.
4. **Data Augmentation Methods:** The training set was subjected to data augmentation methods such colour jitter, random horizontal flips, small rotations, and normalisation. By adding diverse representations to the dataset, these modifications increased the model's resilience to real-world circumstances.
5. **Dropout Layer -** A dropout layer with p=0.2 is being utilised before the final fully connected layer. This is to reduce overfitting, which is very commonly found when a network is being trained on a dataset as small as the one used.
6. **Using an EarlyStopper method**: This allowed for a check on the learning by using the validation accuracy to stop the training early if no significant improvements were observed.

Section 4: Discussion and Conclusion

4.1 Model Performance

In order to find the optimum method, this study explored different architectures in an effort to create an efficient deep learning model for fruit categorisation. With a quick inference time of 20.37 seconds and a high test accuracy of 99.59%, MobileNetV3 was the best-performing model. This outcome highlights how MobileNetV3's lightweight design and effective feature extraction make it a good choice for real-time applications. The model's performance demonstrates the benefits of MobileNetV3's architecture, especially in situations when precision is crucial but computational resources are scarce.

4.3 Challenges and Limitations

Although the model performs well, it has some drawbacks:

- **Class Similarity Problems:** The model has trouble telling certain fruits apart (for example mangoes), which might be fixed with more intricate data augmentation or fine-tuning.
- **Resource Requirements of Other Architectures:** VVision Transformers (ViT) and ConvNeXt achieved high accuracy but required too many resources for practical real-time use.
- **Dependency on Augmentation:** Because the model depends on augmented data, it may perform worse in real-world situations where visual circumstances are not reflected in the augmented training set.

These restrictions offer insightful information on how the model's robustness and applicability might be improved in the future.

4.4 Future Enhancements and Research Directions

A number of future study avenues are suggested in order to enhance the model even more and increase its possible applications:

1. **Advanced Methods for Augmenting Data:** The model may generalise better to a variety of environments and lighting situations if more complex data augmentation techniques are incorporated, such as random cropping, lighting changes, and simulated background fluctuations.
2. **Investigation of Lightweight Transformer Models:** In light of the encouraging outcomes of Vision Transformer (ViT) models, testing lighter transformer-based designs, such MobileViT, may produce a balance between high accuracy and efficiency on par with MobileNetV3.
3. **Group Models to Improve Accuracy:** By utilising the advantages of each architecture to boost performance on difficult classes, creating an ensemble of complementary models (such as combining MobileNetV3 with ViT or EfficientNet) could increase overall accuracy.
4. **Extended Real-World Testing and Fine-Tuning:** Gathering more information from various settings (such as fruits in outdoor markets) may allow the model to accommodate variances found in real-world applications by fine-tuning on such a domain-specific dataset.

4.5 Conclusion

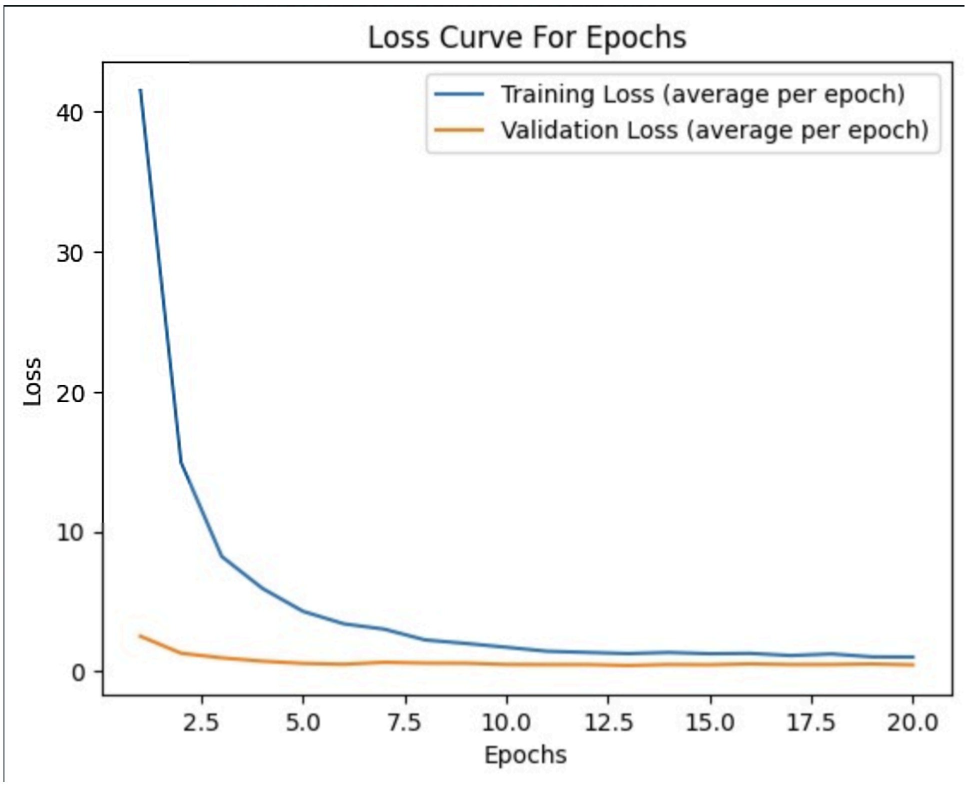
In conclusion, this study effectively created a fruit categorisation model that is both accurate and efficient, with MobileNetV3 turning out to be the best option for practical implementation. The model is appropriate for use in retail, food processing, and agriculture due to its high accuracy and quick inference time. This study highlights the importance of integrating data augmentation, transfer learning, and lightweight architectures to produce reliable and implementable machine learning systems.

With improved data augmentation, sophisticated fine-tuning, and additional research into other lightweight models, future work could overcome the model's shortcomings. With these enhancements, the model may become even more accurate and flexible, allowing it to accommodate a wider variety of needs and deployment situations.

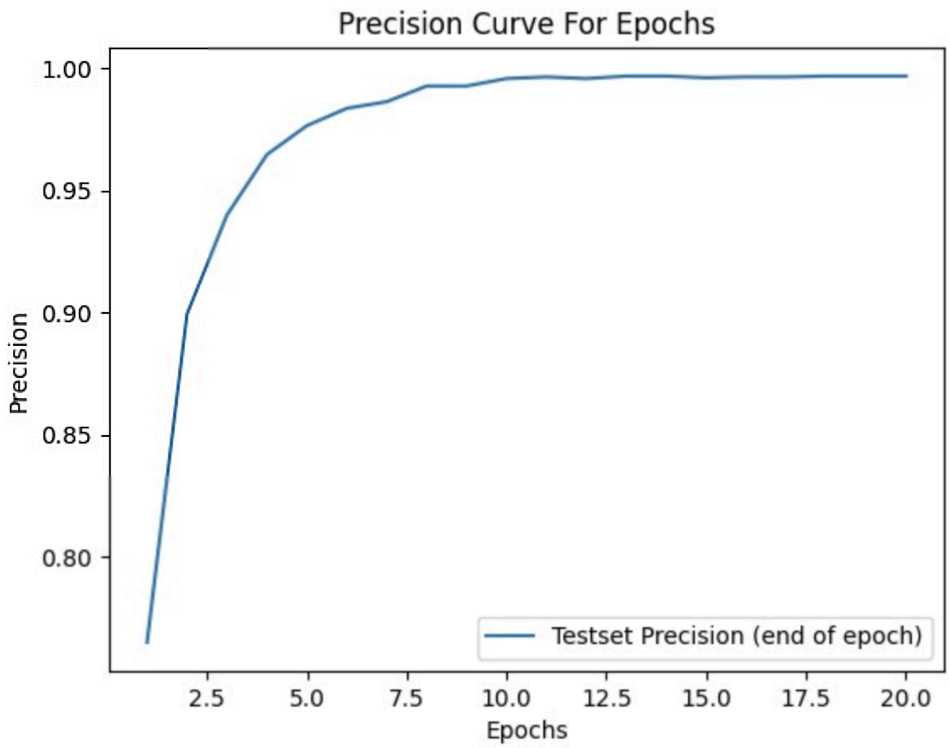
Appendix

Group 14 worked together to finish the project, with Zeeshan Ansari (510370813) and Syed Hamza Kaliyadan (500585454) contributing equally to every facet of the project. Each member was in charge of particular portions of the code and report, and they equally distributed the responsibilities associated with code development and report writing. In addition to studying, developing, and testing the model, Zeeshan and Syed also put in equal amounts of work to describe the technique, experimental assessment, and analysis. Consequently, every team member made around 50% of the total project, guaranteeing equal participation in the technical and documentation aspects.

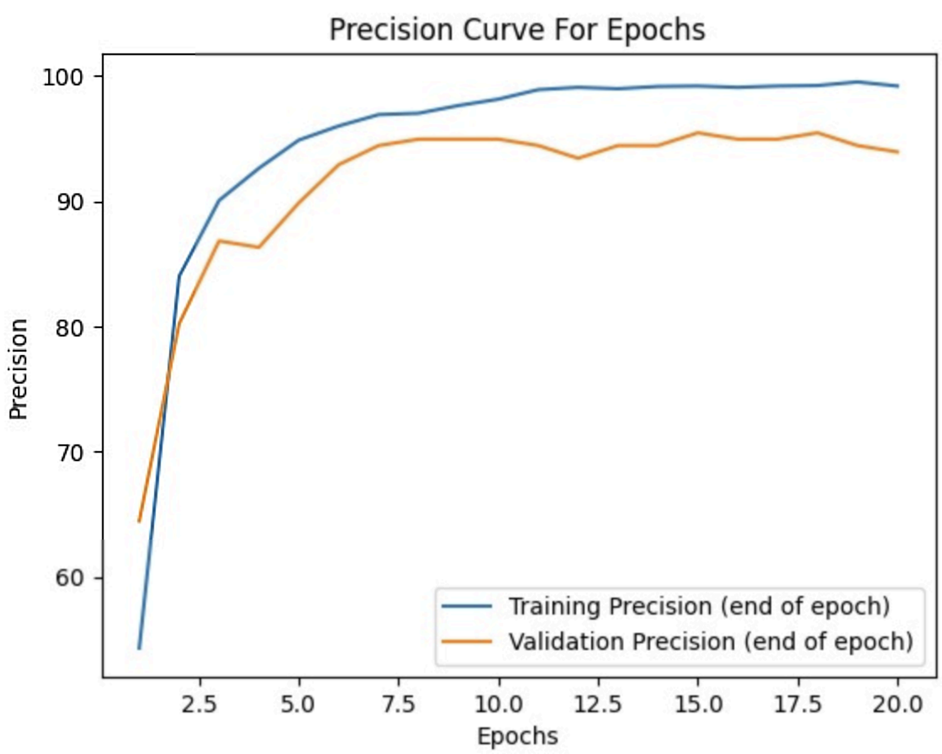
1. **Figure 1: Loss Curve for Epochs**



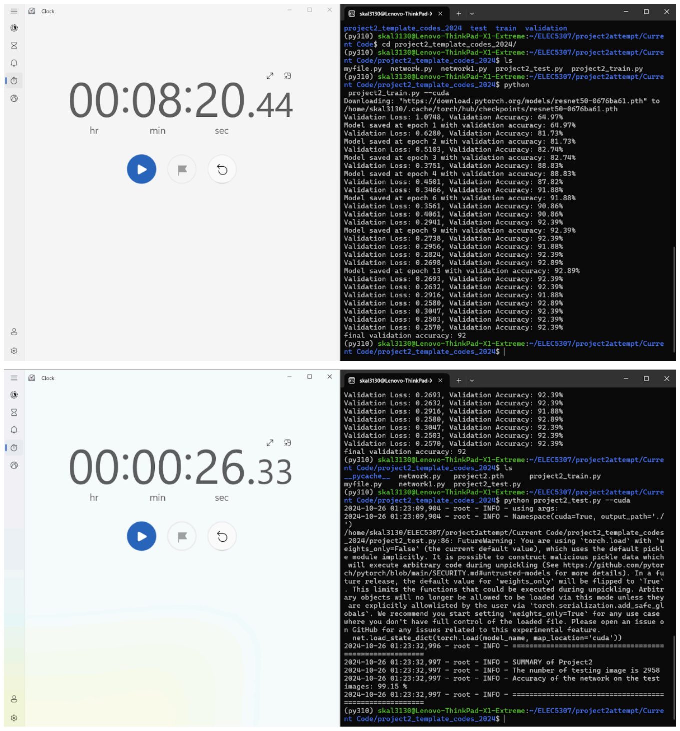
2. **Figure 2: Precision Curve for Epochs**



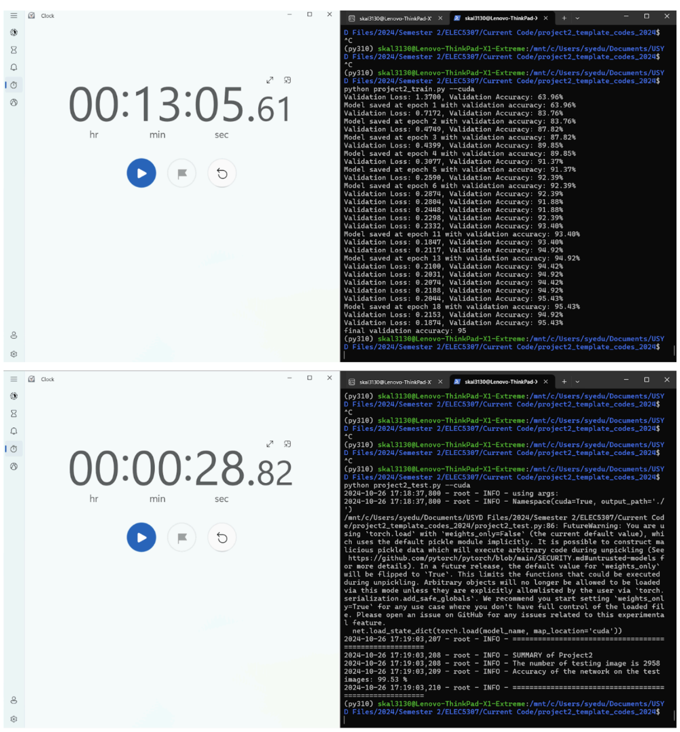
3. **Figure 3: Training and Validation Precision Curve**



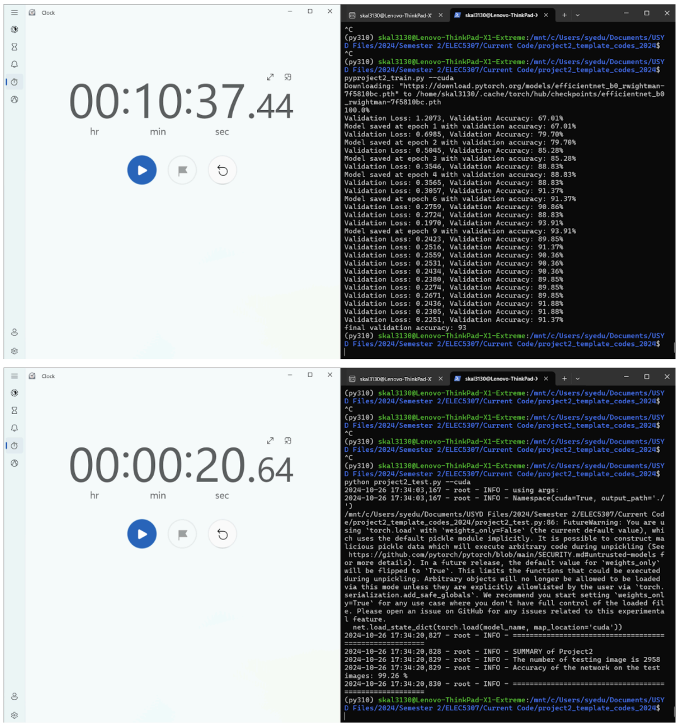
4. **Figure 4: Training and Inference Times for ResNet50 Network with Validation Accuracy**



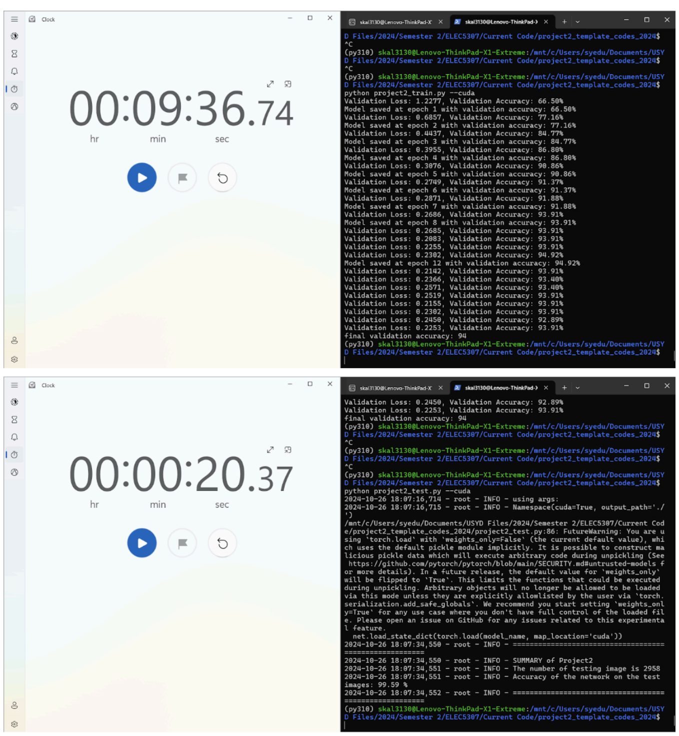
5. **Figure 5: Training and Inference Times for DenseNet Network with Validation Accuracy**



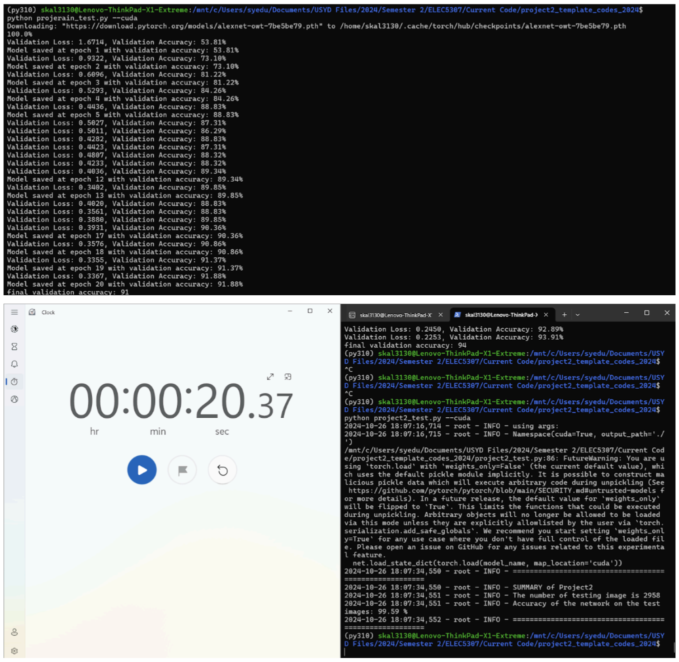
6. **Figure 6: Training and Inference Times for EfficientNet Network with Validation Accuracy**



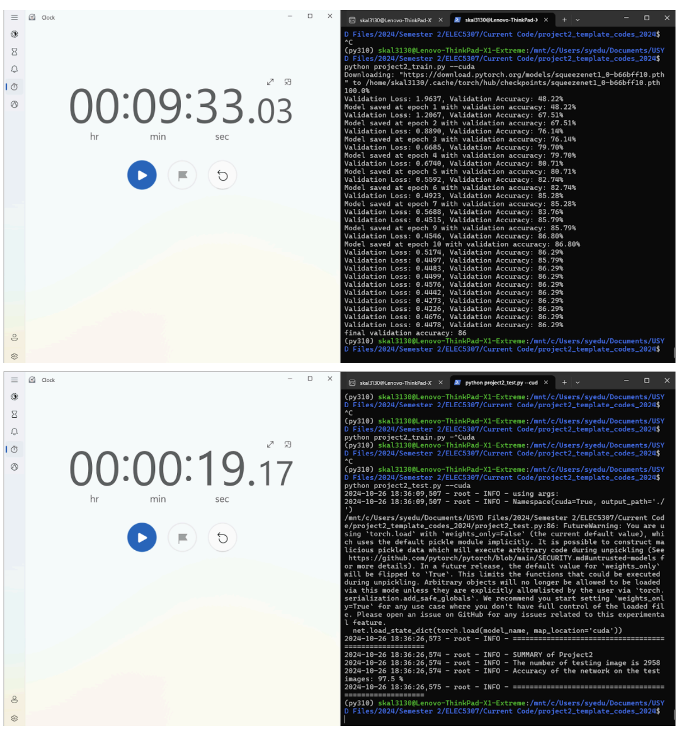
7. **Figure 7: Training and Inference Times for MobileNetV3 Network with Validation Accuracy**



8. **Figure 8: Training and Inference Times for AlexNet Network with Validation Accuracy**

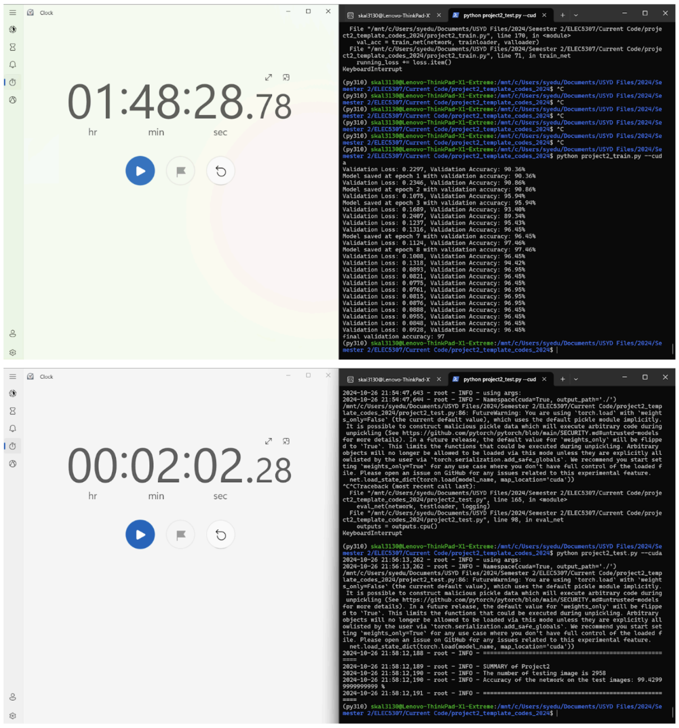


9. **Figure 9: Training and Inference Times for SqueezeNet Network with Validation Accuracy**

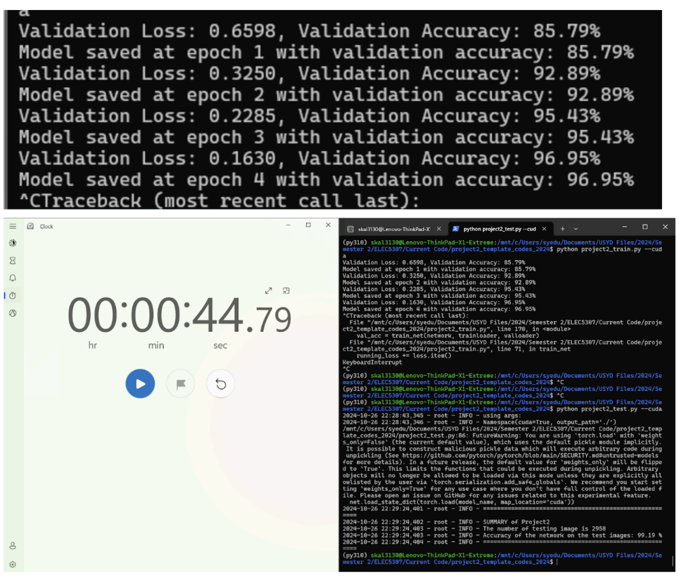


10. **Figure 10: Training and Inference Times for ViT Transformer Network with Validation**

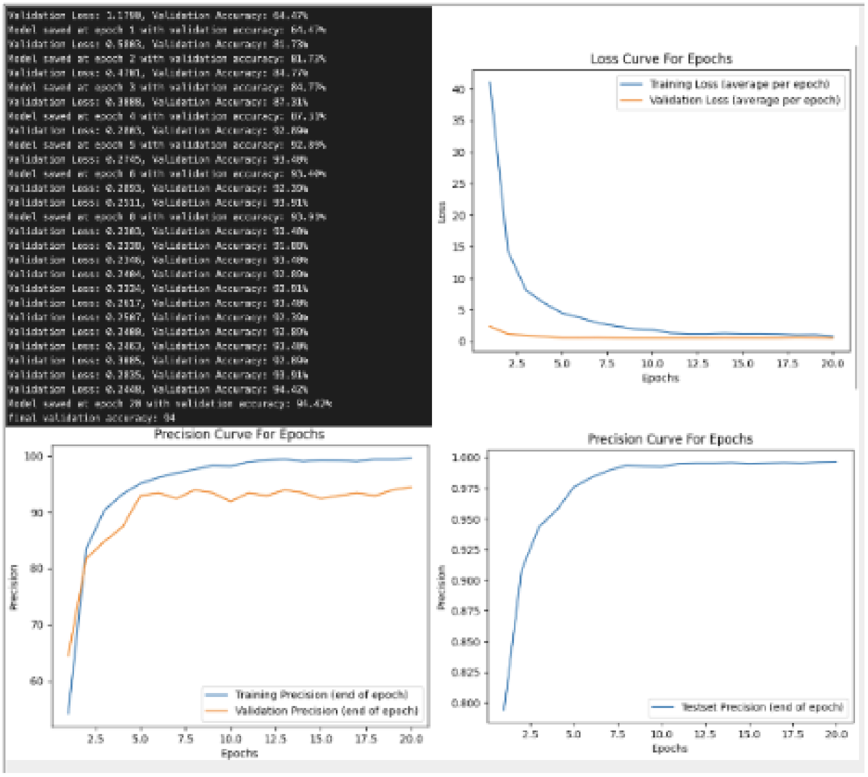
Accuracy



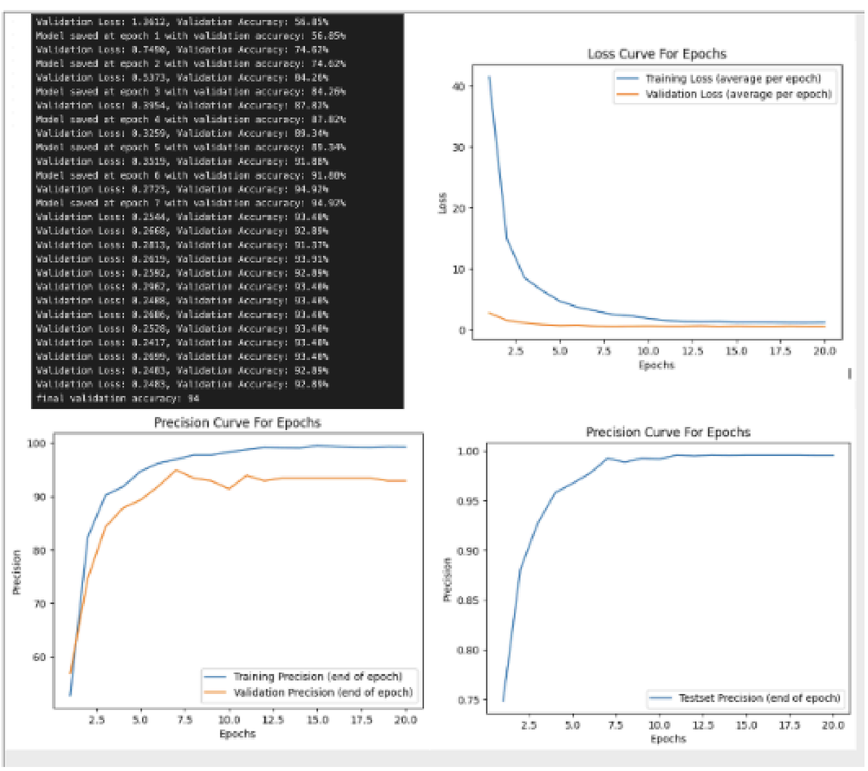
11. **Figure 11: Training and Inference Times for ConvNeXt Network with Validation Accuracy**



12. **Figure 12: Comparison of Step Learning Rate Scheduler on Validation and Test Accuracy**

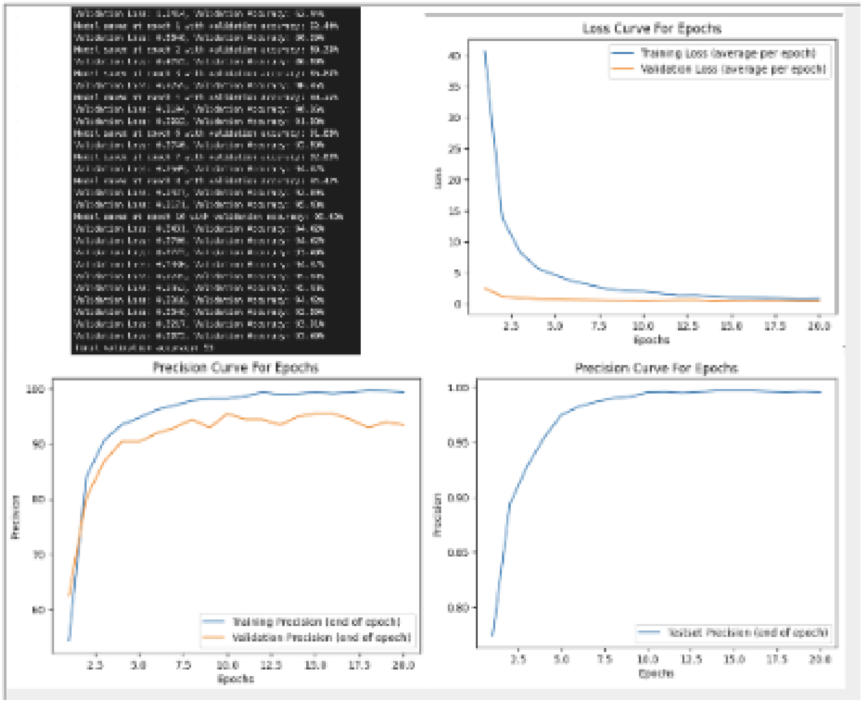


13. **Figure 13: Comparison of MultiStep Learning Rate Scheduler on Validation and Test Accuracy**

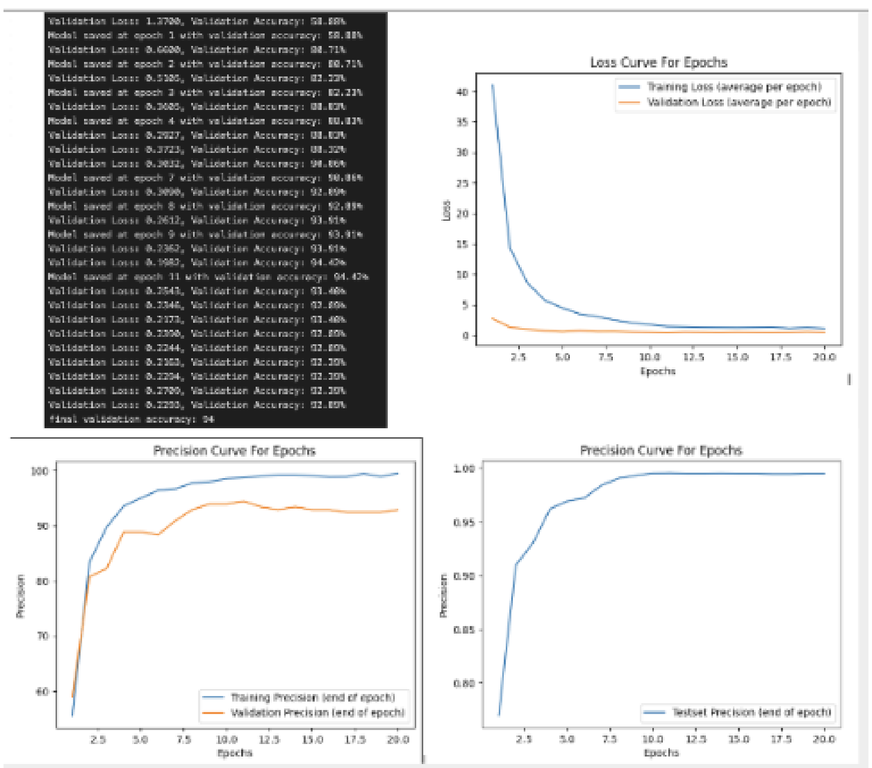


14. **Figure 14: Comparison of Exponential Learning Rate Scheduler on Validation and Test**

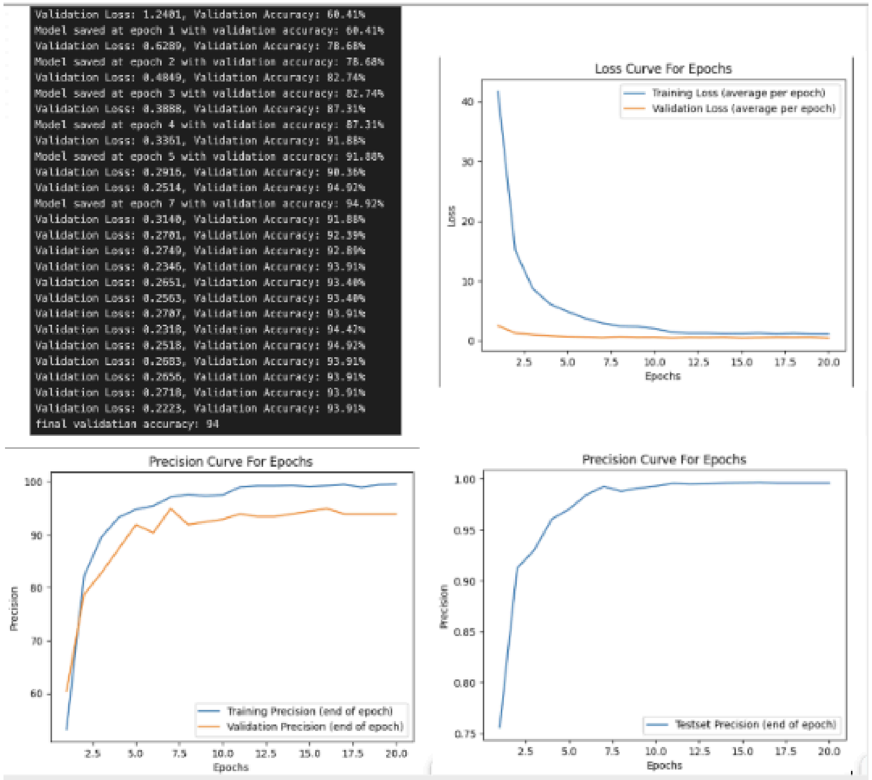
Accuracy



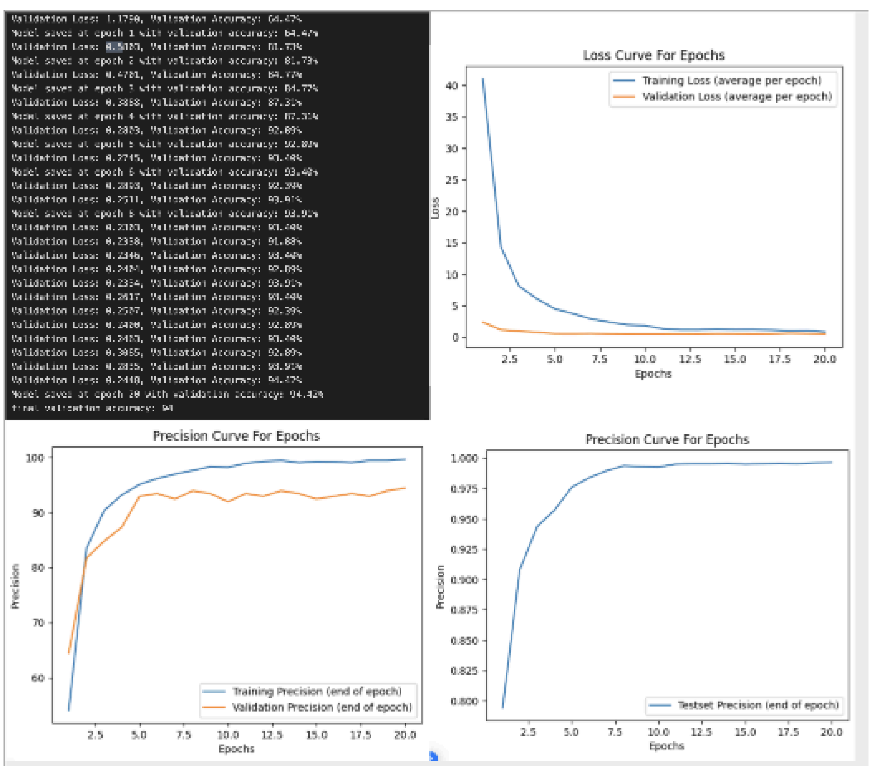
15. **Figure 15: Validation and Test Accuracy with Step Scheduler (Gamma = 0.05)**



16. **Figure 16: Validation and Test Accuracy with Step Scheduler (Gamma = 0.1)**



17. **Figure 17: Validation and Test Accuracy with Step Scheduler (Gamma = 0.2)**



References

[1] Hossain, M.S., Al-Hammadi, M. and Muhammad, G., 2018. Automatic fruit classification using deep learning for industrial applications. *IEEE transactions on industrial informatics*, *15*(2), pp.1027-1034.

[2] Gill, H.S., Khalaf, O.I., Alotaibi, Y., Alghamdi, S. and Alassery, F., 2022. Fruit Image Classification Using Deep Learning. *Computers, Materials & Continua*, *71*(3).

[3] Kathepuri, S., 2020. *Recognition and classification of fruits using deep learning techniques* (Doctoral dissertation, Dublin, National College of Ireland).

[4] Latha, R.S., Sreekanth, G.R., Suganthe, R.C., Geetha, M., Swathi, N., Vaishnavi, S. and Sonasri, P., 2021, January. Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.