Machine Learning HW #2

Andrei-Gabriel Mitran

andreigabrielmitran@gmail.com

Faculty of Automatic Control and Computer Science, Polyethnic University of Bucharest

This report talks about additional methods and models used for classifying two very popular datasets: Fashion-MNIST (items of clothing) and Fruits-360 (fruits and nuts). It will go through how the best models performed for both datasets, their architecture and learning speed.

1. MLP for features

The features were extracted in the exact same way as in the prior report, namely a combination of PCA and HOG.

The Multi-Layer Perceptron (MLP) architecture for features extracted in HW#1 is designed with a balance between model complexity and efficiency, considering the reduced feature space provided by the feature selection process. Below, there is a detailed breakdown of its components and design rationale:

A. Architecture & Training

1. Input Layer

- **Purpose**: The input layer corresponds to the number of features extracted and selected in Step 1.
  - *Fashion-MNIST*: Maximum of 64 features.
  - *Fruits-360*: Maximum of 128 features.
- **Reasoning**: By reducing the feature space, we ensure faster training and reduced risk of overfitting, while maintaining relevant information.

2. Hidden Layers

- **Number of Layers**:
  - *Fashion-MNIST*: Two hidden layers with 64 and 32 neurons, respectively.
  - *Fruits-360*: Two hidden layers with 128 and 64 neurons, respectively.
- **Activation Function**:
  - **ReLU (Rectified Linear Unit)** is used for all hidden layers. It introduces non-linearity, which allows the model to learn complex patterns in the data. ReLU is computationally efficient and helps mitigate the vanishing gradient problem.

- **Rationale**:
  - The first hidden layer has a higher number of neurons to learn more general patterns.
  - The second hidden layer reduces the dimensionality progressively, helping the model focus on the most critical features while minimizing overfitting.

3. Output Layer

- **Number of Neurons**:
  - *Fashion-MNIST*: 10 neurons (one for each class).
  - *Fruits-360*: 141 neurons (one for each class).
- **Activation Function**:
  - **Softmax** is used for the output layer. It converts raw output scores into probabilities, enabling multi-class classification.
- **Reasoning**: The number of neurons in the output layer directly corresponds to the number of classes in the dataset.

4. Dropout Layers

- **Purpose**: Dropout is used after each hidden layer to regularize the model and prevent overfitting.
- **Values**:
  - *Fashion-MNIST*: 30% dropout rate.
  - *Fruits-360*: 40% dropout rate.
- **Rationale**: Since the Fruits-360 dataset is more complex and larger, a higher dropout rate is used to ensure better generalization.

5. Loss Function

- **Categorical Cross-Entropy**:
  - Suitable for multi-class classification problems.
  - Measures the distance between the predicted probability distribution and the true class labels.
- **Reasoning**: This loss function encourages the model to output high probabilities for the correct class while penalizing incorrect predictions.

6. Optimizer

- **Adam Optimizer**:

- Combines the benefits of both RMSprop (adaptive learning rates) and momentum (faster convergence).
- Efficient and works well with sparse gradients.

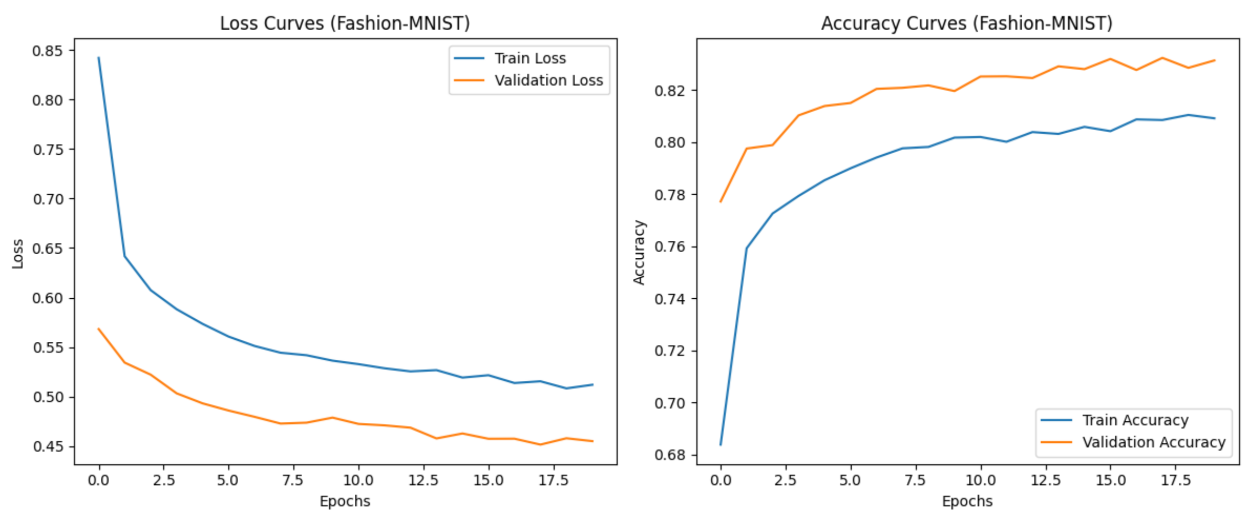
- **Learning Rate**: Set to 0.001, which is a commonly used value for stable training without overshooting.

7. Training Details

- **Epochs**: 20 epochs to ensure sufficient learning without overfitting.
- **Batch Size**:
  - Fashion-MNIST: 32 samples per batch.
  - Fruits-360: 64 samples per batch.
- **Rationale**: A smaller batch size for Fashion-MNIST helps in better gradient approximation given its smaller feature space. The larger batch size for Fruits-360 ensures computational efficiency due to the higher number of features.

B. Loss and Accuracy Curves

a. Fashion-MNIST

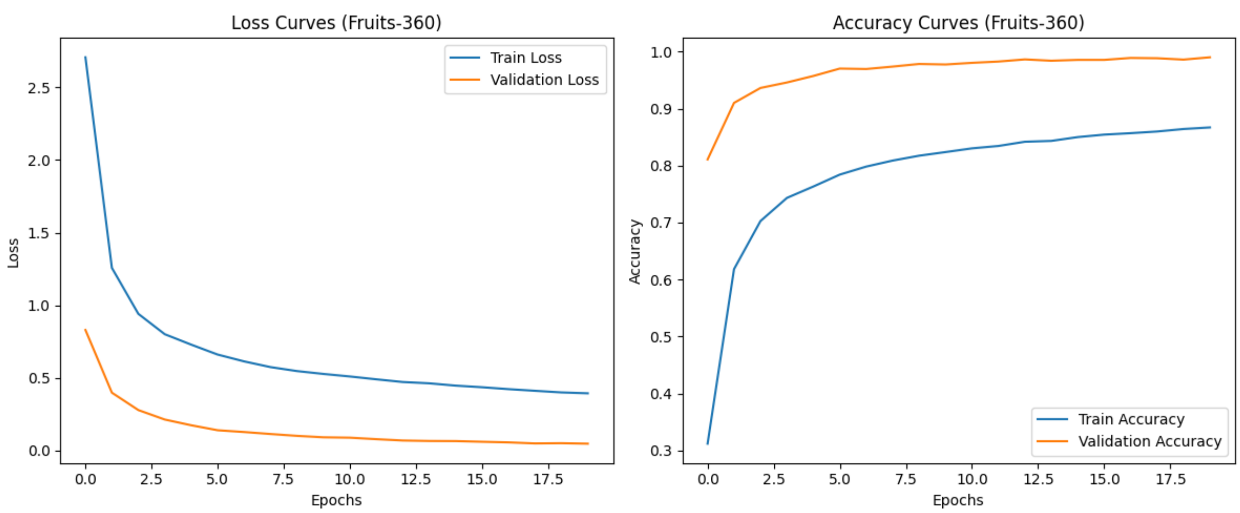


1. **Loss Curves:**
   - The training loss decreases steadily as the epochs progress, which indicates that the model is learning from the training data.
   - The validation loss decreases initially but stabilizes around epoch 10, with only slight fluctuations afterward. This suggests that the model is not overfitting significantly and is generalizing well on unseen data.

2. **Accuracy Curves:**
    - The training accuracy improves consistently, which aligns with the decreasing training loss.
    - The validation accuracy reaches a plateau around epoch 10-15, stabilizing at approximately 82%. This is a strong indication that the model has converged and is performing reliably on the validation set.

The model achieves stable validation performance, indicating that the MLP architecture is appropriate for the Fashion-MNIST dataset. The gap between training and validation accuracy is minimal, suggesting good generalization.

b. Fruits-360



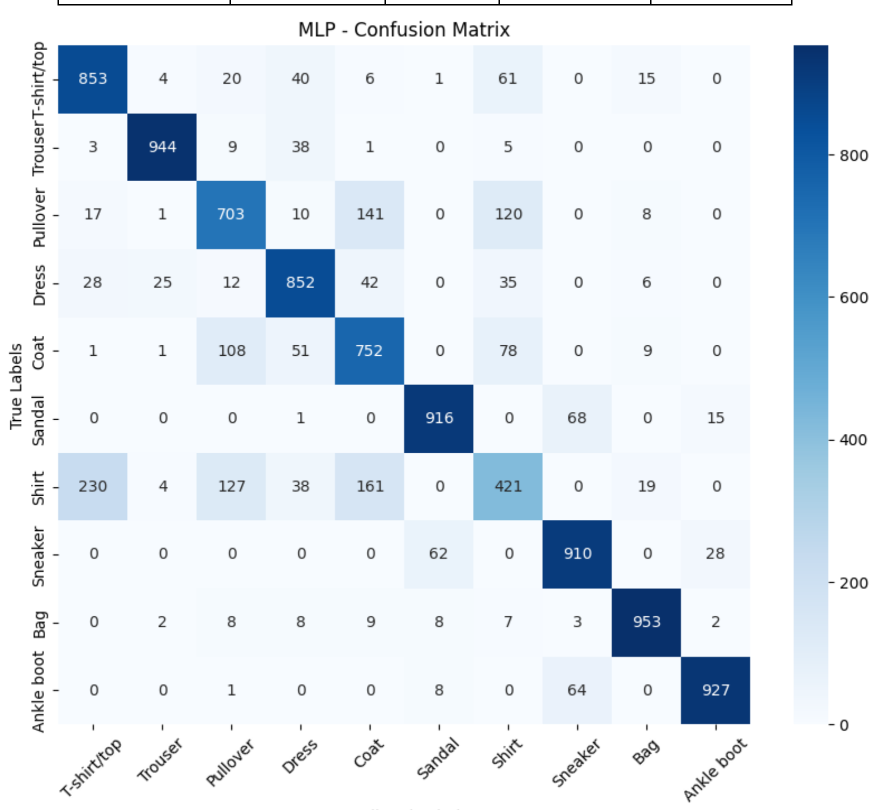
1. **Loss Curves:**
    - The training loss decreases steadily over the epochs, indicating that the model is effectively learning from the training data.
    - The validation loss also decreases significantly in the initial epochs and continues to decline, stabilizing after around epoch 15. This shows that the model generalizes well to the validation set, with minimal risk of overfitting.
2. **Accuracy Curves**:
    - The training accuracy improves consistently throughout the epochs, which corresponds to the steadily decreasing training loss.
    - The validation accuracy increases rapidly in the first few epochs and stabilizes around 98-99% from epoch 10 onward. This suggests that the model has converged and performs reliably on the validation data.

The model achieves excellent validation performance, demonstrating that the MLP architecture is highly effective for the Fruits-360 dataset. The minimal gap between training and validation accuracy confirms robust generalization.

C. Evaluation

1. Fashion-MNIST

Class	Precision	Recall	F1-Score	Support
T-shirt/top	0.75	0.85	0.80	1000
Trouser	0.96	0.94	0.95	1000
Pullover	0.71	0.70	0.71	1000
Dress	0.82	0.85	0.84	1000
Coat	0.68	0.75	0.71	1000
Sandal	0.92	0.92	0.92	1000
Shirt	0.58	0.42	0.49	1000
Sneaker	0.87	0.91	0.89	1000
Bag	0.94	0.95	0.95	1000
Ankle boot	0.95	0.93	0.94	1000



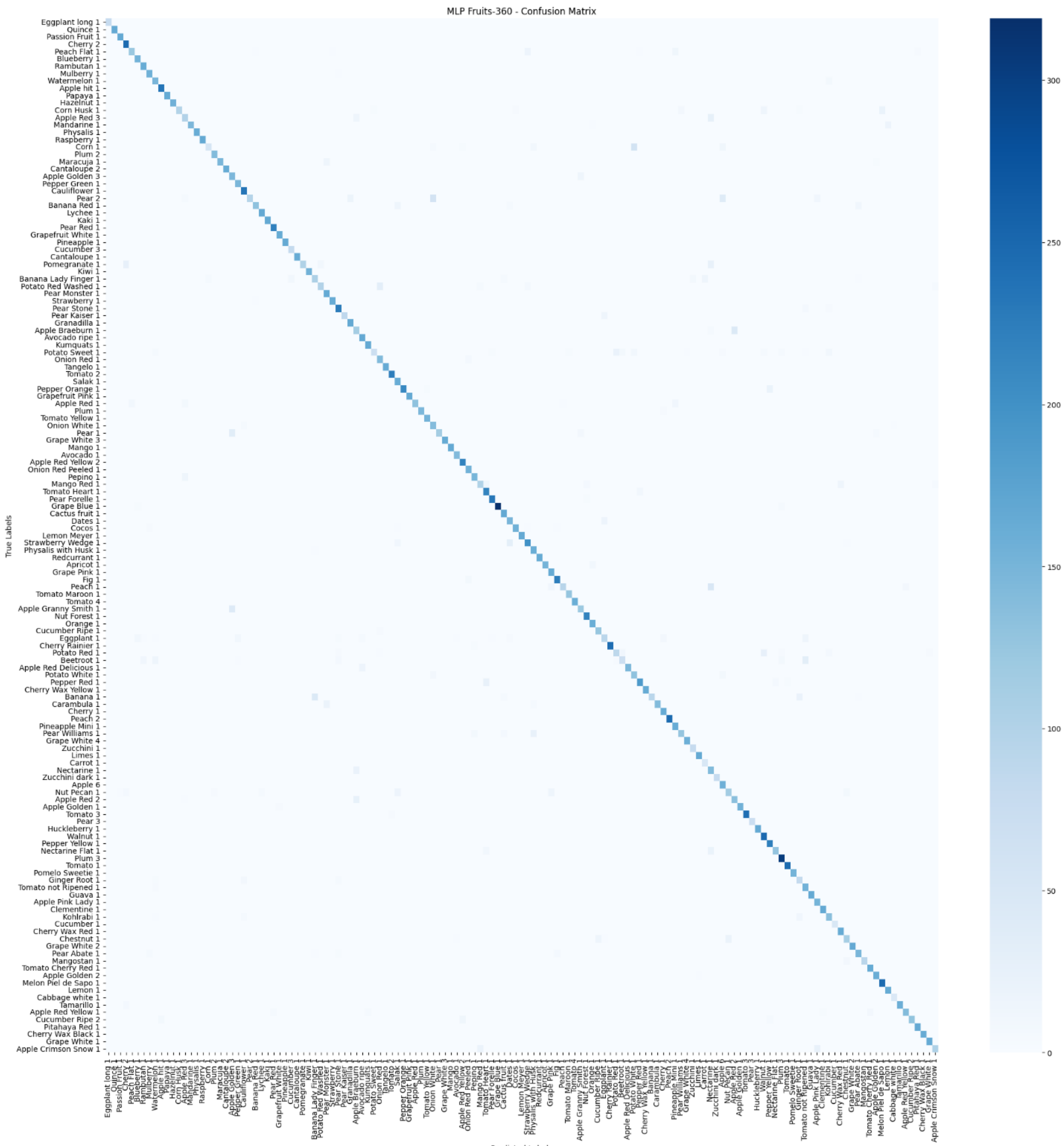
In summary:

- **Accuracy**: 0.82
- **Macro avg (Precision, Recall, F1)**: 0.82, 0.82, 0.82
- **Weighted avg (Precision, Recall, F1)**: 0.82, 0.82, 0.82
- **Speed**: 1 min

The MLP's inferior performance compared to Random Forest, XGBoost, SVM, and Logistic Regression (all of which achieved over 0.85 accuracy) can be attributed to its sensitivity to the quality and distribution of input features. Models like SVM and Logistic Regression excel with small, linearly separable feature sets due to their simpler architectures, while Random Forest and XGBoost handle feature importance and interactions effectively. In contrast, MLPs require more data and careful tuning to learn complex patterns. Without sufficient regularization or with suboptimal feature extraction, the MLP is more prone to overfitting or failing to generalize, leading to weaker results than these simpler, more robust models.

2. Fruits-360

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Eggplant long 1	1.00	1.00	1.00	80
Quince 1	1.00	1.00	1.00	166
Passion Fruit 1	0.99	1.00	0.99	166
Pineapple 1	1.00	1.00	1.00	166
Grapefruit Pink 1	1.00	1.00	1.00	166
Corn 1	0.92	0.37	0.53	150
Potato Sweet 1	0.93	0.47	0.62	150
Beetroot 1	0.77	0.46	0.57	150
Eggplant 1	0.72	0.59	0.65	156
Potato Red 1	0.77	0.59	0.67	150



In summary:

- **Accuracy**: 0.91
- **Macro avg** (Precision, Recall, F1): 0.92, 0.91, 0.91
- **Weighted avg** (Precision, Recall, F1): 0.92, 0.91, 0.91
- **Speed**: 55 s

The MLP achieved an impressive accuracy of **0.91** on the Fruits-360 dataset, almost matching the best results from previous models like Random Forest and XGBoost, and it did so in under one minute of training, making it the most efficient approach. This superior performance can be attributed to the richer and more discriminative features available in Fruits-360, such as color, texture, and shape, compared to the simpler grayscale features in Fashion-MNIST. Additionally, the higher dimensionality of the extracted features (128 for Fruits-360 vs. 64 for Fashion-MNIST) allowed the MLP to better capture complex patterns. Furthermore, the distinct class separations in Fruits-360 provided an advantage, enabling the MLP to learn class boundaries effectively. Combined with dropout regularization and the Adam optimizer, the MLP demonstrated excellent generalization, performing as well as the top models but with significantly less computational time.

2. MLP for flattened images

The MLP architecture for Fashion-MNIST and Fruits-360 is designed to handle pixel-level data directly after flattening the images. This setup balances computational efficiency with the capacity to capture complex data patterns. Below, the architecture and rationale are detailed:

A. Architecture & Training

1. Input Layer

- **Purpose:** The input layer corresponds to the total number of pixels in the flattened images.
  - *Fashion-MNIST:* 28×28=784 features.
  - *Fruits-360:* 28×28×3=2352 features.
- **Reasoning:** Flattening the images allows the MLP to process raw pixel data directly, preserving all information while maintaining compatibility with fully connected layers.

2. Hidden Layers

- **Number of Layers:**
  - *Fashion-MNIST:* Two hidden layers with 256 and 128 neurons, respectively.
  - *Fruits-360:* Two hidden layers with 512 and 256 neurons, respectively.
- **Activation Function:** ReLU (Rectified Linear Unit) is used for all hidden layers.
- **Rationale:**
  - ReLU introduces non-linearity, enabling the model to learn complex patterns.
  - The first hidden layer captures general patterns with a higher number of neurons, while the second hidden layer progressively reduces dimensionality to focus on critical features.

3. Output Layer

- **Number of Neurons:**
  - *Fashion-MNIST:* 10 neurons (one per class).
  - *Fruits-360:* 141 neurons (one per class).
- **Activation Function:** Softmax is used to convert raw outputs into probabilities for multi-class classification.
- **Reasoning:** The number of output neurons matches the number of classes, ensuring proper classification for each dataset.

4. Dropout Layers

- **Purpose:** Dropout is used after each hidden layer to regularize the model and prevent overfitting.
- **Values:**
  - *Fashion-MNIST:* 30% dropout rate.
  - *Fruits-360:* 40% dropout rate.
- **Rationale:** A higher dropout rate is applied for Fruits-360 due to its greater complexity, ensuring better generalization.

5. Loss Function

- **Categorical Cross-Entropy:**
  - Suitable for multi-class classification tasks.
  - Measures the distance between predicted probabilities and true class labels.
- **Reasoning:** This loss function encourages the model to confidently predict the correct class while penalizing incorrect predictions.

6. Optimizer

- **Adam Optimizer:**
  - Combines the benefits of adaptive learning rates and momentum for faster and stable convergence.
  - **Learning Rate:** Set to 0.001, providing a balance between training speed and stability.

7. Training Details

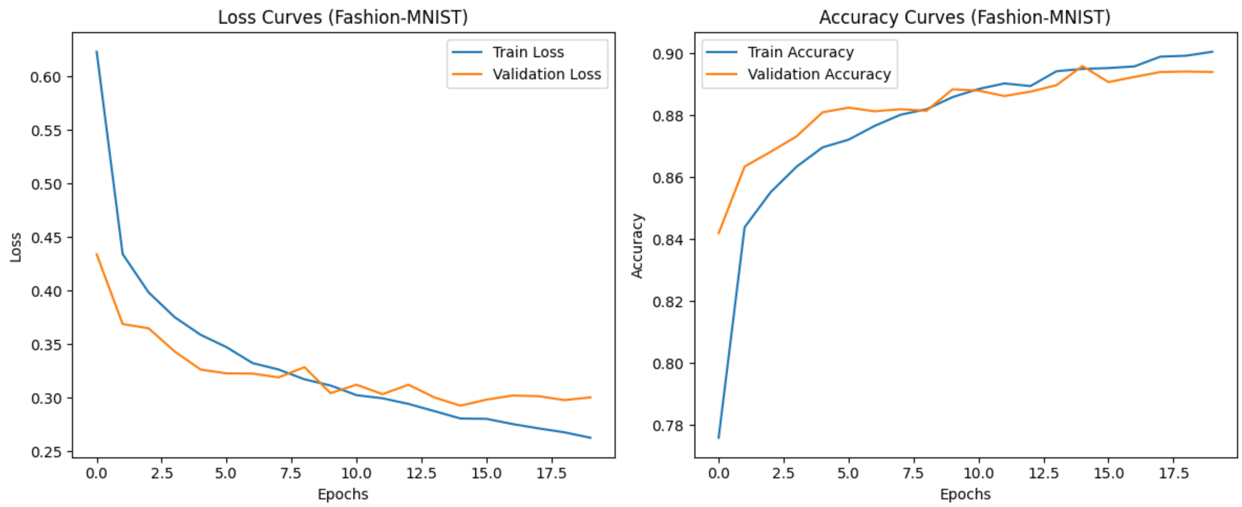
- **Epochs:** 20 epochs to ensure sufficient learning while avoiding overfitting.
- **Batch Size:**

- *Fashion-MNIST:* 64 samples per batch.
- *Fruits-360:* 64 samples per batch.

- **Rationale:** The chosen batch size balances computational efficiency and stable gradient updates, tailored for the complexity of each dataset.

B. Loss and Accuracy Curves

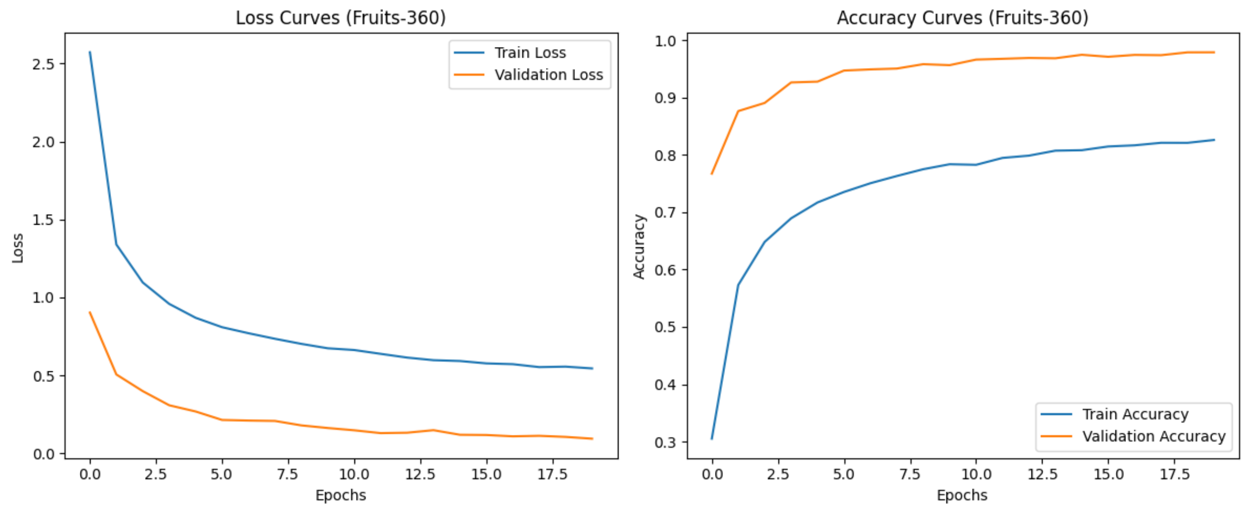
a. Fashion-MNIST



1. **Loss Curves:**
   - The training loss exhibits a consistent decrease over the epochs, indicating that the model is effectively learning the underlying patterns in the training data.
   - The validation loss initially decreases sharply, then stabilizes after epoch 8 with minimal fluctuations, suggesting that the model is not overfitting and maintains a good balance between bias and variance.
2. **Accuracy Curves:**
   - Training accuracy improves steadily, reflecting the model's ability to correctly classify more training samples as it learns.
   - Validation accuracy plateaus between epochs 8-15, reaching a stable level of approximately 89%, which confirms that the model is well-suited for the Fashion-MNIST dataset and performs reliably on unseen data.

The MLP architecture demonstrates strong generalization capabilities, as evidenced by the minimal gap between training and validation curves for both loss and accuracy. This balance indicates the chosen model is both efficient and effective for this classification task.

b. Fruits-360



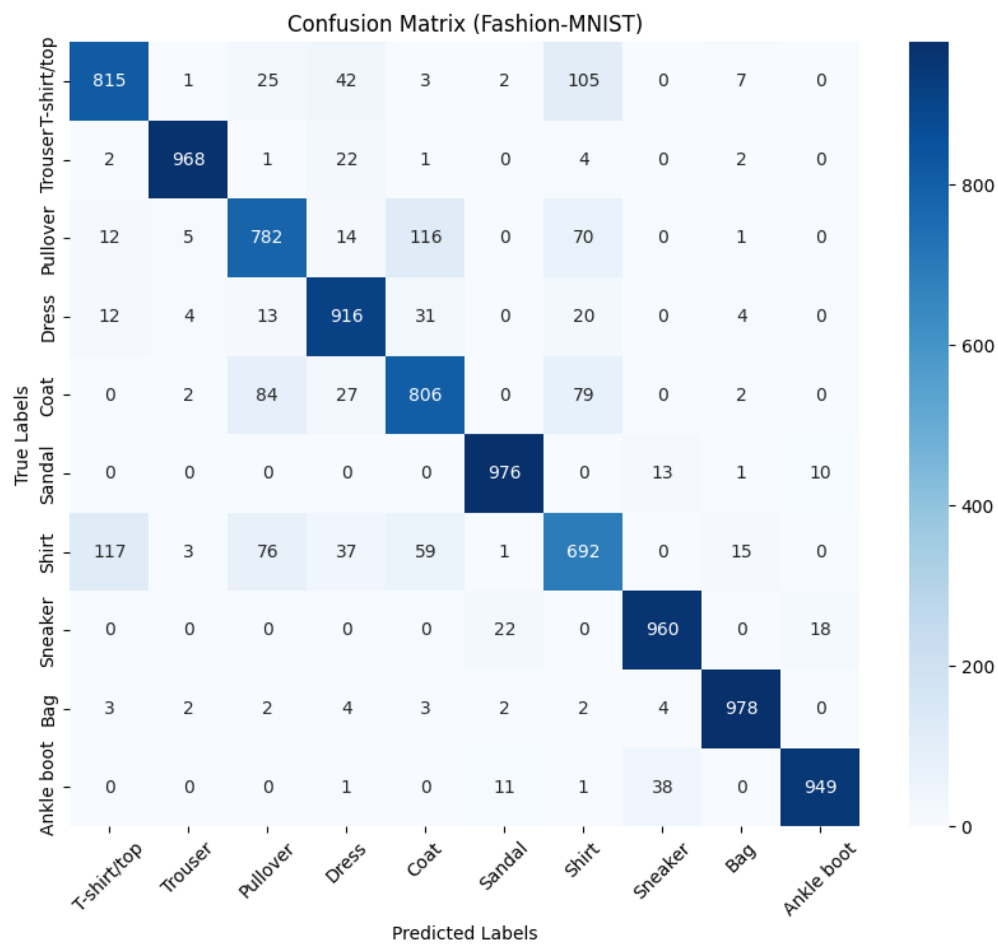
1. **Loss Curves:**
   - The training loss decreases consistently across epochs, indicating that the model effectively captures patterns in the Fruits-360 dataset.
   - The validation loss decreases sharply in the initial epochs and stabilizes after epoch 8, with minimal fluctuations, suggesting robust generalization without signs of overfitting.
2. **Accuracy Curves:**
   - Training accuracy improves steadily, aligning with the reduction in training loss, showcasing the model's growing proficiency in classifying the data correctly.
   - Validation accuracy reaches a high plateau around epoch 8-12, stabilizing at approximately 97%. This demonstrates the model's strong performance and reliability on the validation set.

The MLP architecture proves to be highly effective for the Fruits-360 dataset, achieving near-perfect generalization as evidenced by the minimal gap between the training and validation accuracy curves. This reflects the model's ability to handle the complexity of the dataset while maintaining computational efficiency.

C. Evaluation

1. Fashion-MNIST

Class	Precision	Recall	F1-Score	Support
T-shirt/top	0.85	0.81	0.83	1000
Trouser	0.98	0.97	0.98	1000
Pullover	0.80	0.78	0.79	1000
Dress	0.86	0.92	0.89	1000
Coat	0.79	0.81	0.80	1000
Sandal	0.96	0.98	0.97	1000
Shirt	0.71	0.69	0.70	1000
Sneaker	0.95	0.96	0.95	1000
Bag	0.97	0.98	0.97	1000
Ankle boot	0.97	0.95	0.96	1000



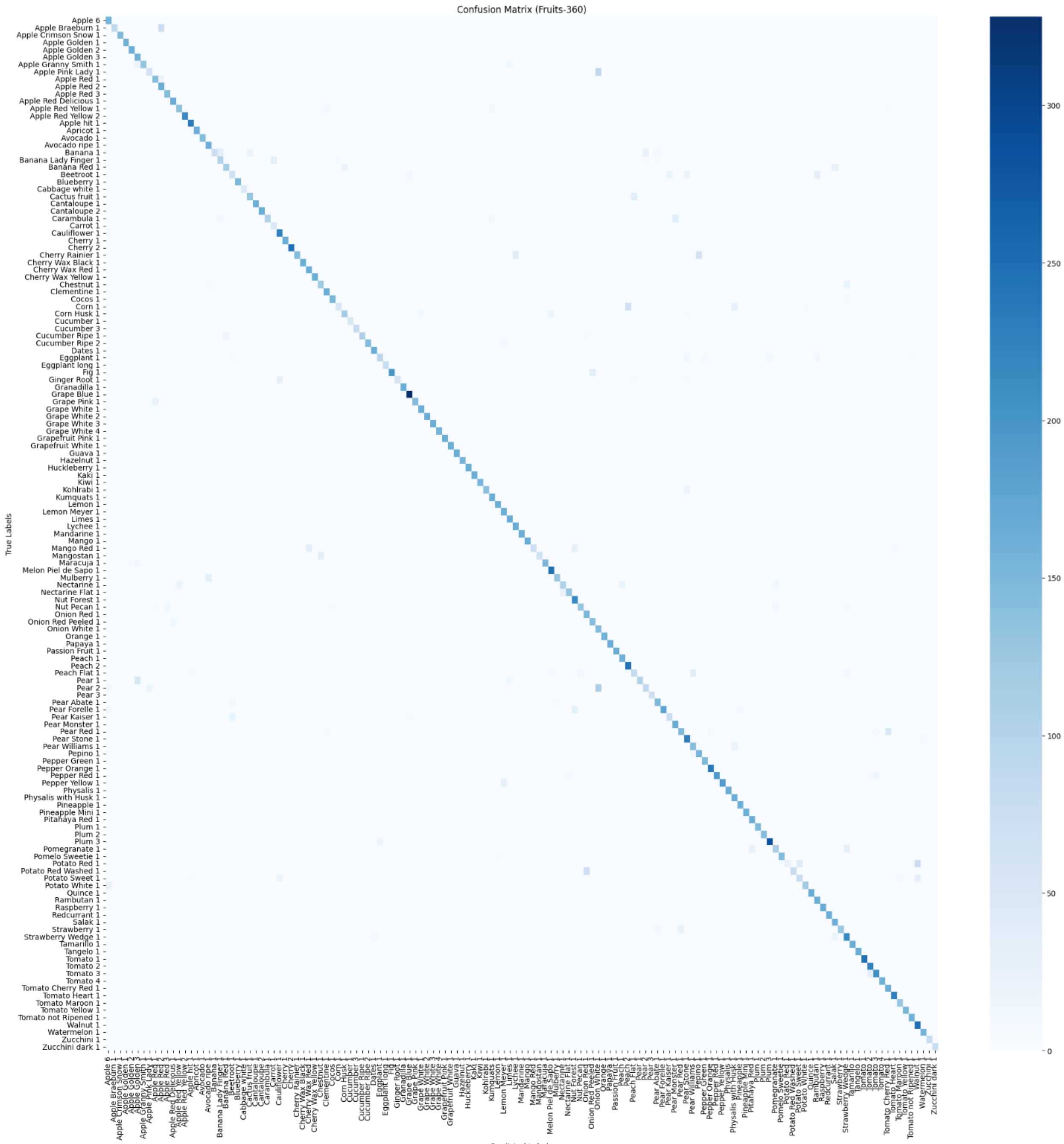
In summary:

- **Accuracy:** 0.88
- **Macro avg (Precision, Recall, F1):** 0.88, 0.88, 0.88
- **Weighted avg (Precision, Recall, F1):** 0.88, 0.88, 0.88
- **Speed:** 47 s

The MLP trained directly on pixel data for Fashion-MNIST outperforms the MLP trained on extracted features, achieving an accuracy of 0.88 compared to 0.82. This improvement can be attributed to the richer representation provided by the raw pixel data, which allows the model to learn more nuanced and detailed patterns. In contrast, the feature-extracted MLP relies heavily on the quality of the selected features, which may not fully capture the intricate relationships present in the data. Additionally, the larger network capacity of the pixel-based MLP, with its higher number of neurons, enables it to model complex patterns more effectively. However, the increased data dimensionality necessitates careful regularization, such as dropout, to prevent overfitting, which was less critical for the simpler feature-based MLP.

2. Fruits-360

Class	Precision	Recall	F1-Score	Support
Apple Crimson Snow 1	0.99	1.00	1.00	148
Apple Golden 1	1.00	1.00	1.00	160
Grapefruit Pink 1	1.00	1.00	1.00	166
Raspberry 1	1.00	1.00	1.00	166
Tomato Yellow 1	1.00	1.00	1.00	153
Potato Red 1	0.96	0.17	0.28	150
Corn 1	0.88	0.35	0.50	150
Pear 2	0.75	0.38	0.50	232
Beetroot 1	0.64	0.45	0.53	150
Apple Pink Lady 1	0.76	0.40	0.53	152



Summary:

- **Accuracy:** 0.90
- **Macro avg (Precision, Recall, F1):** 0.91, 0.90, 0.89
- **Weighted avg (Precision, Recall, F1):** 0.91, 0.90, 0.89
- **Speed:** 1 min

The MLP trained on extracted features outperformed the MLP trained on flattened images in both accuracy and speed. The feature-based MLP achieved an accuracy of 0.91, compared to 0.90 for the flattened image MLP, while also being faster, with a training time of just 55 seconds versus 1 minute for the flattened image MLP. This improved performance can be attributed to the reduced and refined feature set, which eliminates irrelevant or redundant data, enabling the model to focus on the most discriminative attributes. The flattened image MLP, despite having access to richer raw data, likely suffered from the inclusion of noise and redundant information, slightly hindering its performance. The results demonstrate the effectiveness of feature selection in enhancing both model efficiency and accuracy, particularly when working with high-dimensional data such as Fruits-360.

3. CNN

The Convolutional Neural Network (CNN) architecture for Fashion-MNIST and Fruits-360 datasets is designed to leverage spatial hierarchies in image data, enabling effective feature extraction and classification. Below is a detailed breakdown of its components and design rationale:

A. Architecture

1. **Input Layer**
   - **Purpose**: The input layer processes the raw image data directly, with the channel dimension accounting for grayscale or RGB data.
     - *Fashion-MNIST*: Input shape is (1, 28, 28) for grayscale images.
     - *Fruits-360*: Input shape is (3, 32, 32) for RGB images.
   - **Reasoning**: Images are fed directly without flattening to retain spatial relationships, critical for convolutional operations.
2. **Convolutional Layers**
   - **Number of Layers**:
     - *Fashion-MNIST*: Two convolutional layers followed by max-pooling and batch normalization.
     - *Fruits-360*: Three convolutional layers, each followed by max-pooling and batch normalization.
   - **Activation Function**:
     - **ReLU (Rectified Linear Unit)** is applied after each convolutional operation to introduce non-linearity and mitigate the vanishing gradient problem.

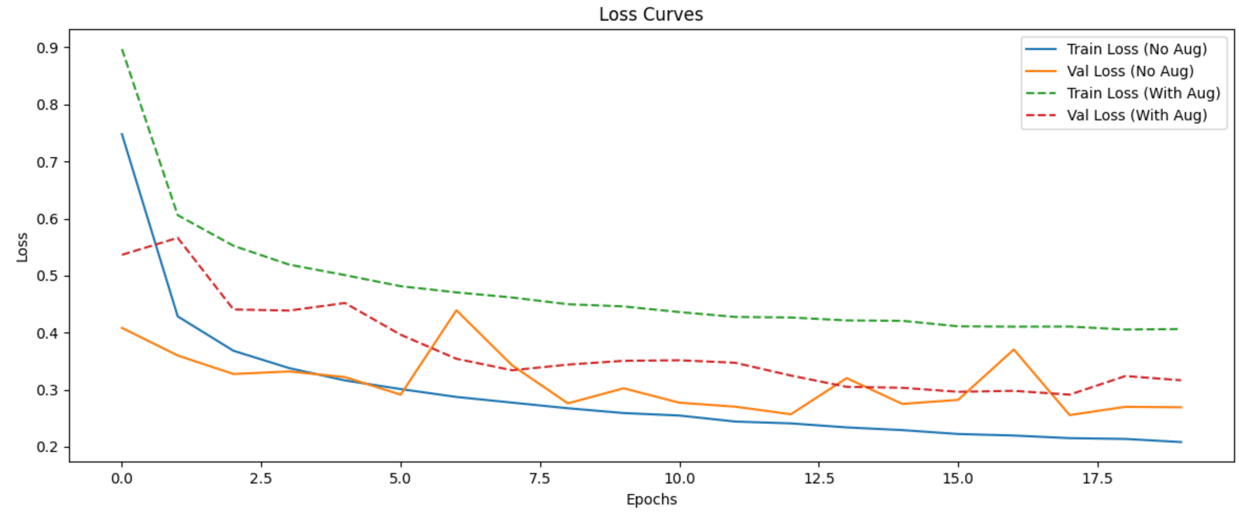
- **Rationale**:
    - The first convolutional layer extracts basic features like edges and textures.
    - Subsequent layers learn more complex patterns, enabling hierarchical feature extraction.
    - Batch normalization stabilizes and accelerates training by normalizing activations, while max-pooling reduces spatial dimensions, improving computational efficiency and reducing overfitting.

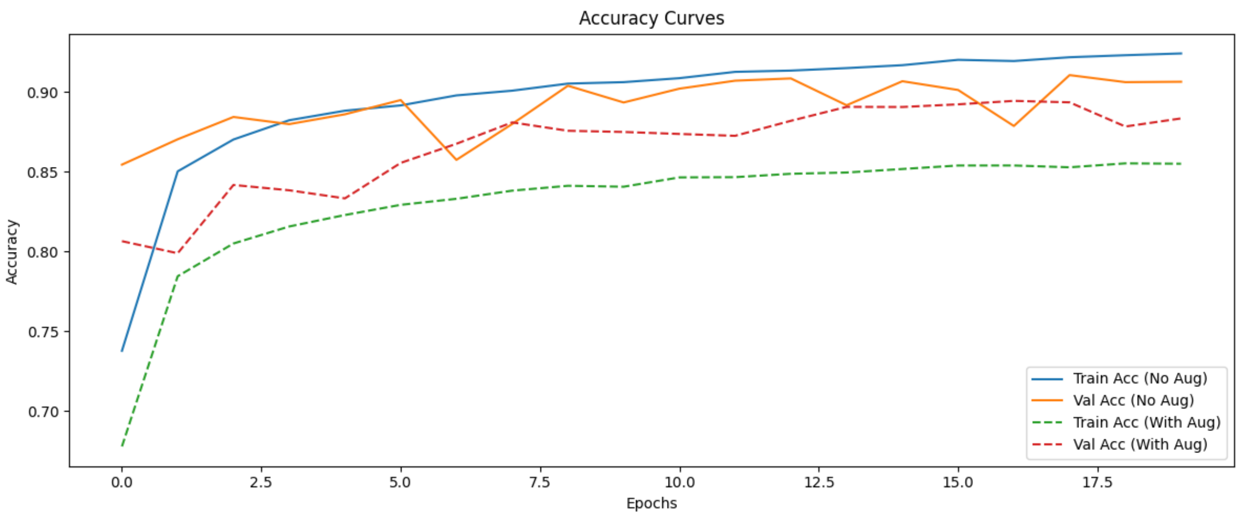
3. **Global Pooling Layer**
    - **Purpose**: An adaptive average pooling layer reduces each feature map to a size of 1x1, summarizing spatial information.
    - **Rationale**: Reduces overfitting by significantly decreasing the number of parameters before the fully connected layers.
4. **Fully Connected Layers**
    - **Number of Layers**:
        - *Fashion-MNIST*: Two fully connected layers—one with 128 neurons and a final output layer with 10 neurons.
        - *Fruits-360*: Two fully connected layers—one with 256 neurons and a final output layer with 141 neurons.
    - **Activation Function**:
        - **ReLU** for the intermediate fully connected layer to learn non-linear patterns.
        - **Softmax** for the output layer to produce class probabilities.
    - **Rationale**:
        - The intermediate layer compresses the features learned by the CNN into a dense representation.
        - The output layer provides probabilities for each class, allowing multi-class classification.
5. **Dropout Layers**
    - **Purpose**: Dropout is applied to the fully connected layers to regularize the model and prevent overfitting.
    - **Values**:

        - *Fashion-MNIST*: 50% dropout rate.
        - *Fruits-360*: 50% dropout rate.
    - **Rationale**: Dropout introduces stochasticity during training, improving the model's ability to generalize.
6. **Loss Function**
    - **Categorical Cross-Entropy**:
        - Suitable for multi-class classification tasks.
        - Measures the divergence between predicted probabilities and true class labels.
    - **Rationale**: Encourages the model to output high probabilities for the correct class and penalizes incorrect predictions.
7. **Optimizer**
    - **Adam Optimizer**:
        - Combines momentum and adaptive learning rates for efficient and stable convergence.
    - **Learning Rate**: 0.001, which balances stability and convergence speed.
    - **Rationale**: Adam's adaptive nature makes it well-suited for deep networks with non-stationary objectives.
8. **Training Details**
    - **Epochs**: 20, ensuring sufficient learning while avoiding overfitting.
    - **Batch Size**: 64 for both Fashion-MNIST and Fruits-360.
    - **Rationale**: Batch size of 64 provides a good trade-off between gradient approximation and computational efficiency.

B. Loss and Accuracy Curves

a. Fashion-MNIST





1. **Loss Curves**:
    - The training loss for the non-augmented model (No Aug) decreases steadily, showing consistent learning, whereas the augmented model (With Aug) starts with a higher loss and decreases more gradually. This difference is due to the variability introduced by the augmented training samples.
    - The validation loss for the No Aug model stabilizes rather late and presents some spikes throughout. However, the With Aug model exhibits slightly higher validation loss, as the augmented data increases the complexity of the learning process.
2. **Accuracy Curves**:
    - The training accuracy for the No Aug model improves faster and stabilizes around 92%, while the With Aug model lags slightly, reaching a lower plateau of approximately 89%. This reflects that the augmentation introduces more

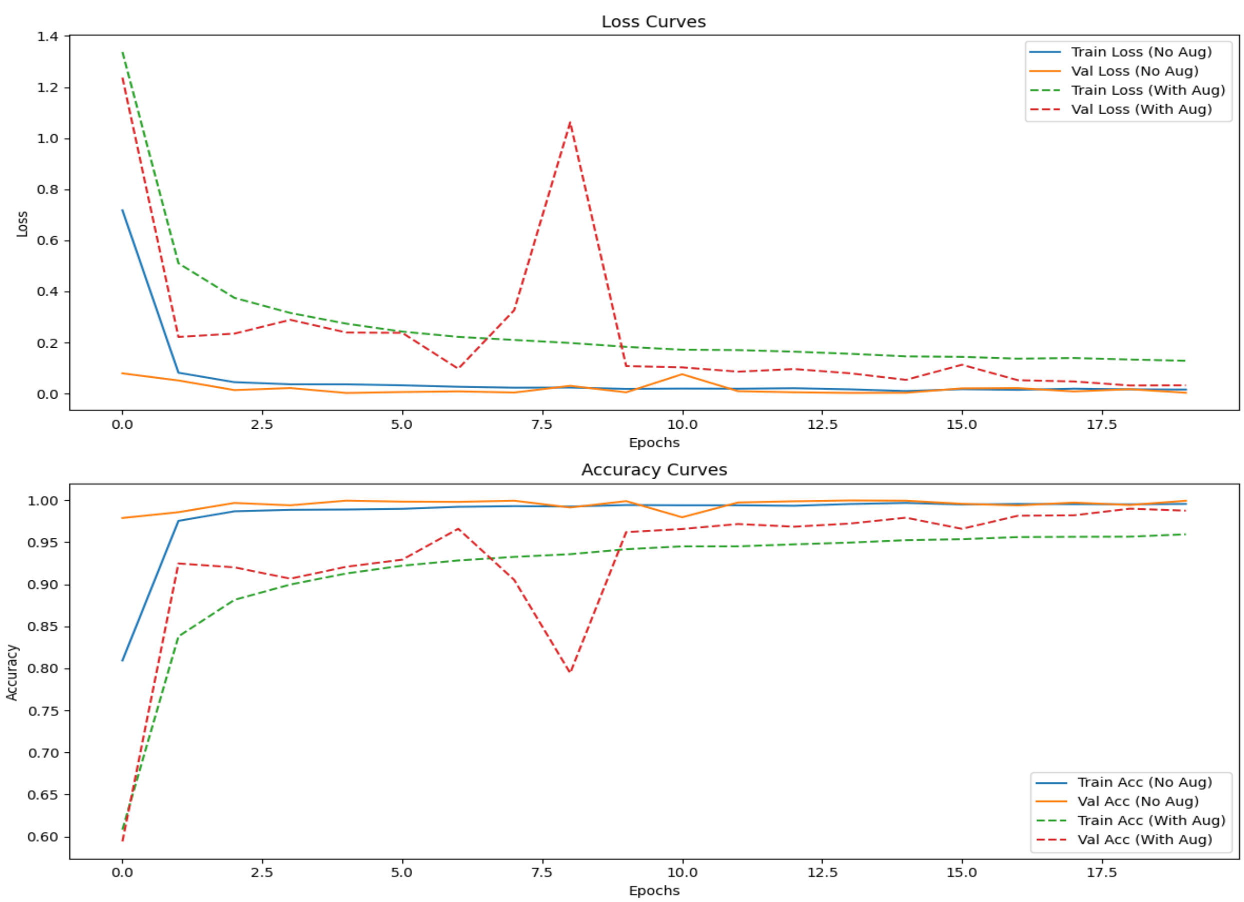
challenging variations in the training data, making it harder for the model to achieve high training accuracy.

- Validation accuracy for both models stabilizes, with the No Aug model consistently outperforming the With Aug model by about 2%. This suggests that while augmentation diversifies the dataset and improves robustness, it might slightly hinder performance on this specific validation set.

Comparison and Explanation:
The No Aug model benefits from a simpler learning process due to the consistency of the original training data. In contrast, the With Aug model was trained on data augmented with random horizontal flips, rotations, and random cropping, which added diversity and increased robustness at the cost of slightly reduced accuracy. However, the augmented model is expected to generalize better to unseen, real-world data where similar variations may occur.

The discrepancy in performance highlights the trade-off between achieving high validation accuracy and building a model robust to variations not present in the validation set. While the No Aug model performed slightly better in this controlled setting, the With Aug model is likely better suited for applications involving more varied or noisy data.

b. Fruits-360



1. **Loss Curves**:
    - The training loss for the non-augmented model (No Aug) consistently decreases and stabilizes at a very low level, indicating effective learning. Conversely, the With Aug model shows a more erratic loss pattern, with spikes around epoch 8. This behavior is likely due to the variations introduced by augmentation, which increase the difficulty of training.
    - Validation loss for the No Aug model remains stable at a low value, while for the With Aug model, the validation loss initially drops but then fluctuates significantly. This suggests that the augmented model faces challenges in maintaining consistency, potentially due to the diversity of the augmented dataset.
2. **Accuracy Curves**:
    - The training accuracy for the No Aug model rapidly reaches near-perfect levels, reflecting its ability to fit the relatively uniform training data. For the With Aug model, training accuracy improves steadily but experiences dips, aligning with the observed loss spikes. This indicates the model struggles more to adapt to the augmented data.
    - Validation accuracy is higher for the No Aug model, stabilizing above 97%. The With Aug model, while reaching comparable levels later, displays fluctuations, showing that augmentation introduces more variability, requiring the model to generalize more effectively.

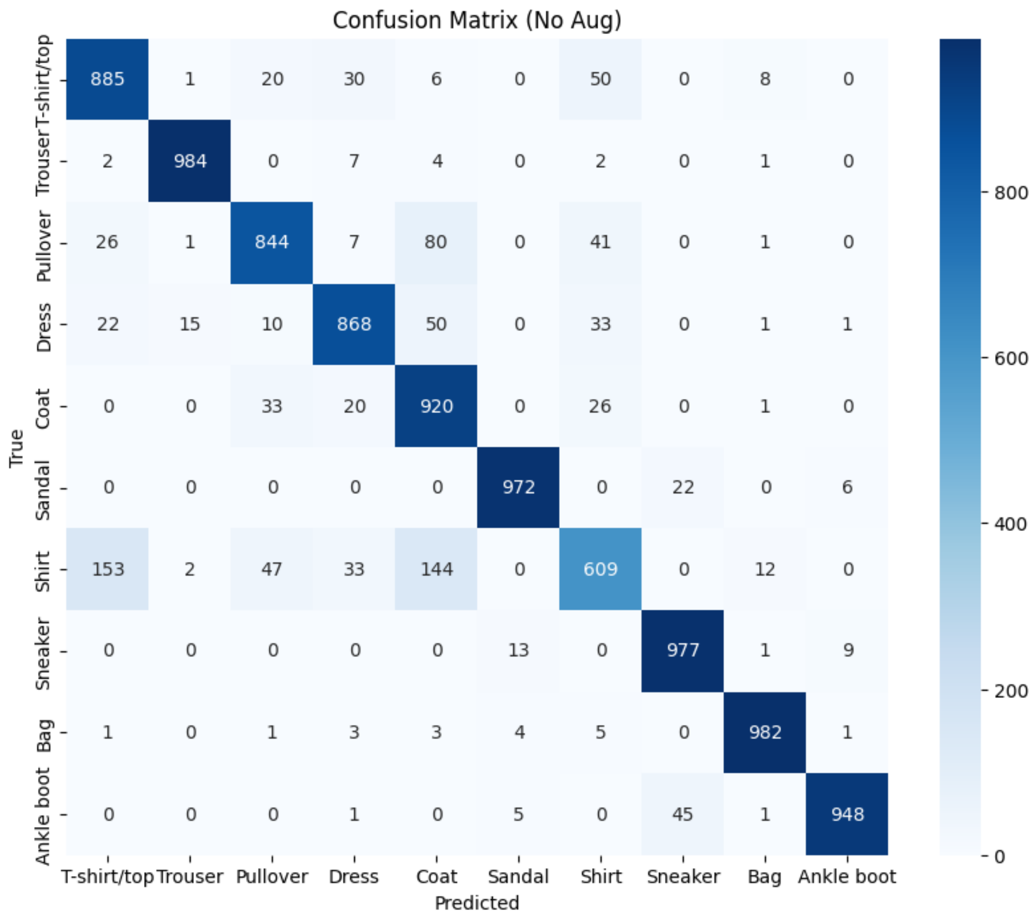
Comparison and Explanation:
The No Aug model benefits from the simpler, unmodified training data, leading to faster convergence and more stable validation performance. The With Aug model, trained on data subjected to random horizontal flips, rotations, and resized cropping, faces a more complex learning task. This augmentation increases robustness by exposing the model to variations likely to appear in real-world scenarios but also introduces challenges that result in less stable performance. Despite this, the With Aug model is likely better equipped to generalize beyond the test set, making it advantageous for real-world deployment where input data may include similar transformations.

C. Evaluation

1. Fashion-MNIST

a) No Aug

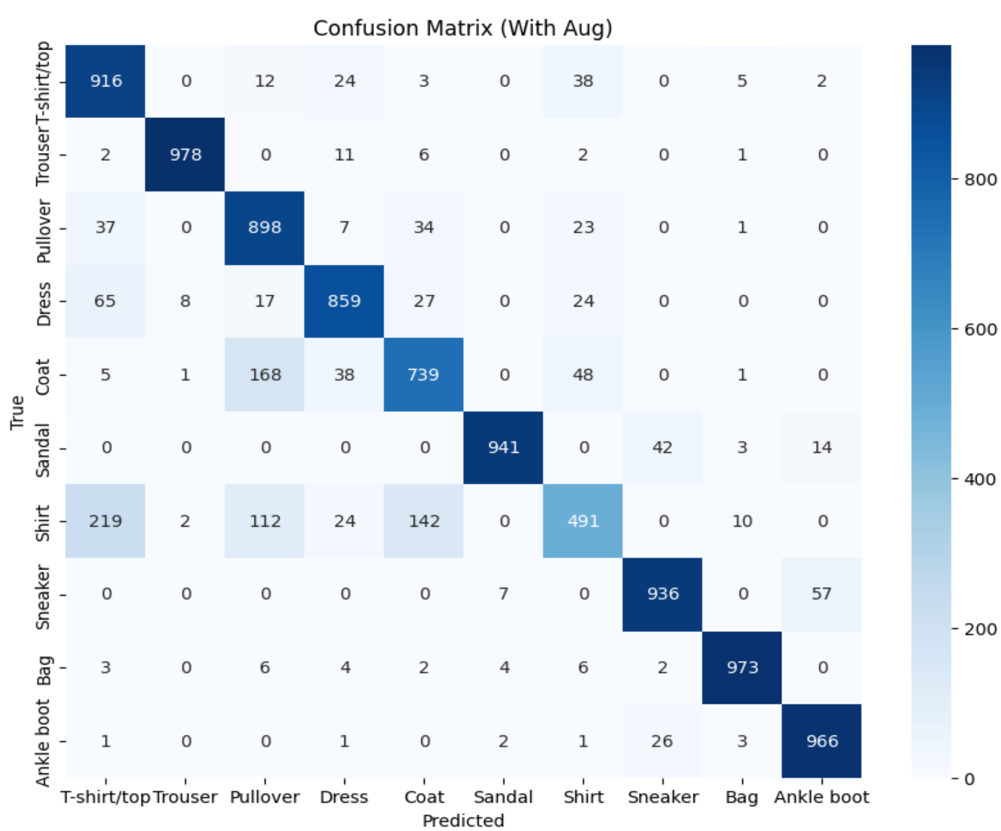
Class	Precision	Recall	F1-Score	Support
T-shirt/top	0.81	0.89	0.85	1000
Trouser	0.98	0.98	0.98	1000
Pullover	0.88	0.84	0.86	1000
Dress	0.90	0.87	0.88	1000
Coat	0.76	0.92	0.83	1000
Sandal	0.98	0.97	0.97	1000
Shirt	0.80	0.61	0.69	1000
Sneaker	0.94	0.98	0.96	1000
Bag	0.97	0.98	0.98	1000
Ankle boot	0.98	0.95	0.96	1000



- **Accuracy**: 0.90
- **Macro avg (Precision, Recall, F1)**: 0.90, 0.90, 0.90
- **Weighted avg (Precision, Recall, F1)**: 0.90, 0.90, 0.90
- **Speed**: 1 min

b) With Aug

Class	Precision	Recall	F1-Score	Support
T-shirt/top	0.73	0.92	0.81	1000
Trouser	0.99	0.98	0.98	1000
Pullover	0.74	0.90	0.81	1000
Dress	0.89	0.86	0.87	1000
Coat	0.78	0.74	0.76	1000
Sandal	0.99	0.94	0.96	1000
Shirt	0.78	0.49	0.60	1000
Sneaker	0.93	0.94	0.93	1000
Bag	0.98	0.97	0.97	1000
Ankle boot	0.93	0.97	0.95	1000



- **Accuracy**: 0.87
- **Macro avg (Precision, Recall, F1)**: 0.87, 0.87, 0.87
- **Weighted avg (Precision, Recall, F1)**: 0.87, 0.87, 0.87
- **Speed**: 8 min

The CNN models demonstrated superior performance compared to the MLPs on the Fashion-MNIST dataset, particularly when trained on raw images. The No Augmentation CNN achieved a higher accuracy of 90% in just 1 minute, while the With Augmentation CNN reached 87% in 8 minutes. In contrast, the MLP trained on images achieved 88% accuracy in 47 seconds, while the MLP trained on extracted features reached only 82% in 1 minute. These results highlight the strength of CNNs in processing spatial information directly from image data.

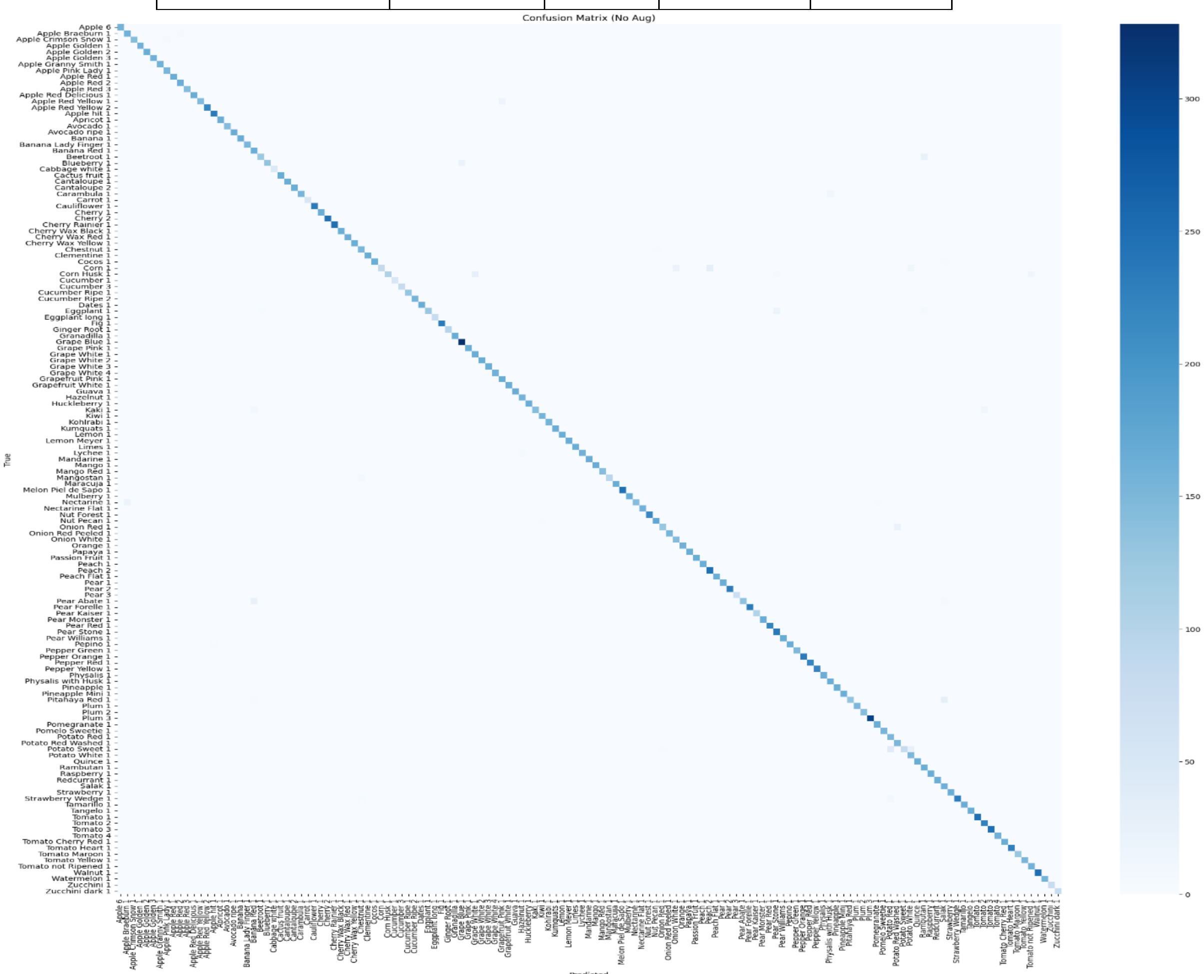
The discrepancy in performance stems from the ability of CNNs to automatically learn hierarchical features through convolutional layers, which are particularly effective for image data. MLPs, on the other hand, lack this spatial awareness and rely solely on global features, which limits their ability to capture finer details. Additionally, the CNN's use of batch normalization and adaptive pooling helps stabilize and optimize the learning process, contributing to its superior accuracy.

In terms of efficiency, while the No Augmentation CNN required the same training time as the MLP on features, it achieved significantly better accuracy. This demonstrates that CNNs are not only more effective for raw image classification but also computationally feasible. However, the With Augmentation CNN showed a drop in accuracy and longer training time, likely due to the increased variability introduced by augmentation, which may require more epochs or better tuning to yield consistent improvements. Overall, the CNNs proved to be the most effective models for Fashion-MNIST, surpassing MLPs in terms of accuracy and suitability for image data.

2. Fruits-360

a) No Aug

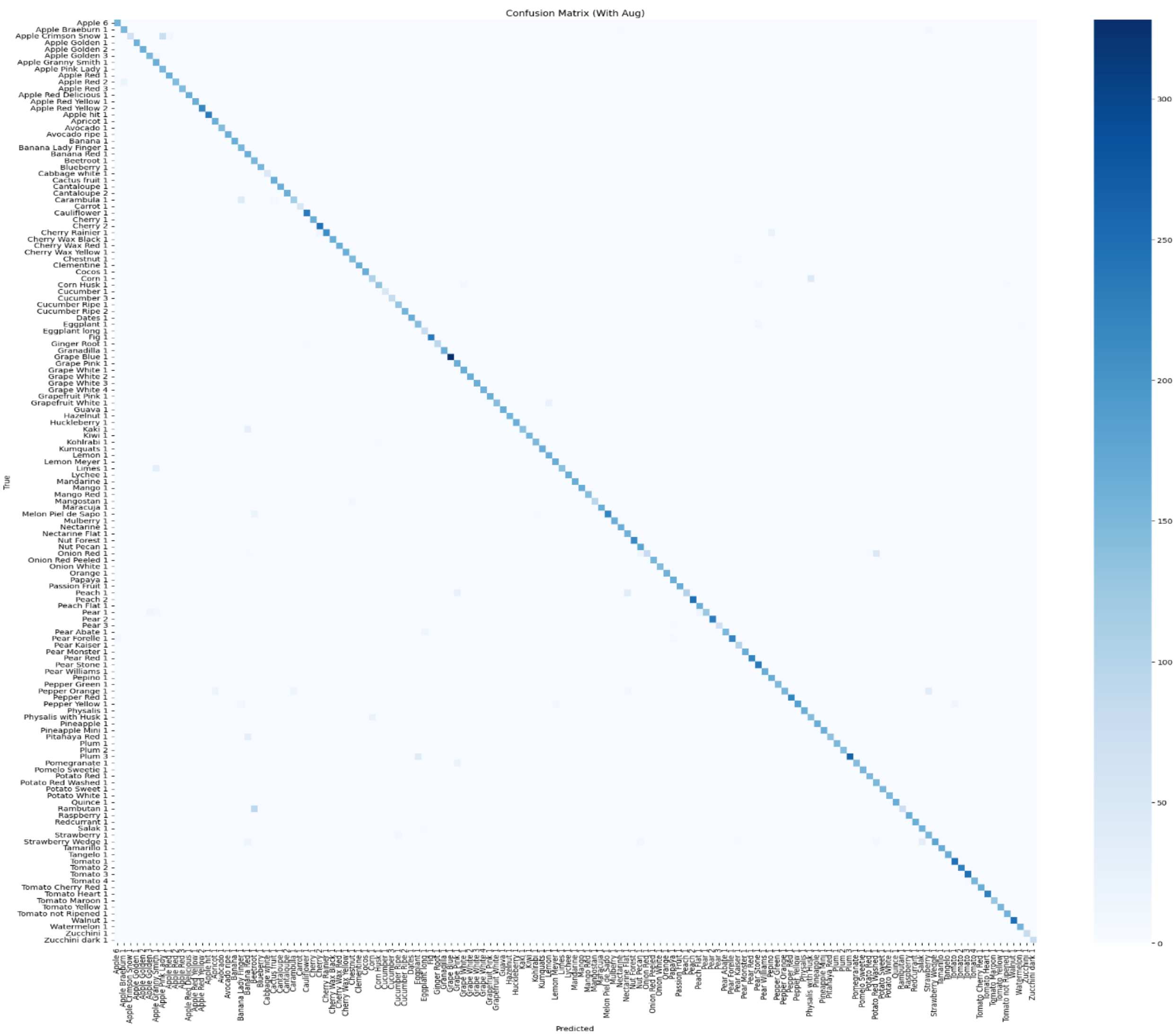
Class	Precision	Recall	F1-Score	Support
Apple 6	1.00	1.00	1.00	157
Apple Golden 1	1.00	1.00	1.00	160
Apple Golden 2	1.00	1.00	1.00	164
Apple Golden 3	0.96	1.00	0.98	161
Avocado ripe	1.00	1.00	1.00	166
Corn	1.00	0.56	0.72	150
Corn Husk	0.99	0.68	0.80	154
Eggplant	1.00	0.81	0.90	156
Potato Sweet	1.00	0.55	0.71	150
Potato Red	0.75	1.00	0.85	150



- **Accuracy**: 0.98
- **Macro avg (Precision, Recall, F1)**: 0.98, 0.98, 0.98
- **Weighted avg (Precision, Recall, F1)**: 0.98, 0.98, 0.98
- **Speed**: 1 min

b) With Aug

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Apple Red Yellow 2	1.00	1.00	1.00	219
Apple hit	1.00	1.00	1.00	234
Avocado ripe	1.00	1.00	1.00	166
Banana	1.00	1.00	1.00	166
Cherry Wax Yellow	1.00	1.00	1.00	164
Apple Crimson Snow	1.00	0.45	0.62	148
Rambutan	1.00	0.45	0.62	164
Corn	0.83	0.69	0.76	150
Carambula	0.89	0.71	0.79	166
Beetroot	0.57	1.00	0.73	150



- **Accuracy**: 0.96
- **Macro avg (Precision, Recall, F1)**: 0.96, 0.96, 0.96
- **Weighted avg (Precision, Recall, F1)**: 0.97, 0.96, 0.96
- **Speed**: 23 min

The CNN implementations on Fruits-360 outperformed both MLPs in accuracy while matching or exceeding their training speed. The CNN without augmentation achieved an accuracy of **0.98**, significantly higher than the **0.91** accuracy of the feature-based MLP and the **0.90** accuracy of the flattened image MLP. Even the augmented CNN, with an accuracy of **0.96**, outperformed both MLPs. In terms of speed, the CNN (No Aug) matched the feature-based MLP, completing training in **1 minute**, while the augmented CNN required **23 minutes** due to the overhead of data augmentation. The superior performance of the CNNs can be attributed to their ability to learn spatial hierarchies and extract intricate patterns, which MLPs, even with refined features, cannot exploit. The CNNs' robustness to raw image data and ability to capture spatial structures give them a decisive edge in performance for image classification tasks like Fruits-360.

4. **The ResNet-18 Pretrained Model**

The ResNet-18 model in this project was pretrained on the **CIFAR-10** dataset, which contains 60,000 32x32 color images across 10 classes. This choice of pretraining aligns well with the target datasets, as both **Fashion-MNIST** and **Fruits-360** have resolutions smaller than the 224x224 standard of **ImageNet**, making CIFAR-10 a better match in terms of image size and characteristics.

A. **Adjustments**

1. **Dataset-Specific**
    a) **Fashion-MNIST**
    Fashion-MNIST consists of grayscale images with a resolution of 32x32. Since ResNet-18 expects 3-channel color images, the single grayscale channel was duplicated to create a 3-channel image. This adjustment ensured compatibility with the model while preserving the dataset's original visual information.
    b) **Fruits-360**
    Fruits-360 images, initially at a resolution of 100x100, were resized to 32x32 to match the input size of the pretrained ResNet-18. This resizing allowed for seamless integration into the model while retaining essential visual features for accurate classification.
2. **Fine-Tuning Procedure**

To adapt ResNet-18 for the specific tasks of classifying Fashion-MNIST and Fruits-360 images, the model's final fully connected (fc) layer was replaced with task-specific layers:

- *Fashion-MNIST*: 10 output neurons (one per clothing category).
- *Fruits-360*: 141 output neurons (one per fruit class).

In addition to replacing the fc layer, **fine-tuning was applied to all layers** of the network. This allowed the model to recalibrate its learned features to better suit the unique characteristics of each dataset. This step was especially important given the stark contrast between the datasets: grayscale images in Fashion-MNIST and colorful, high-detail images in Fruits-360.

3. **Choice of Optimizer: SGD vs. Adam**

The fine-tuning process used **Stochastic Gradient Descent (SGD)** with momentum as the optimizer, as opposed to **Adam**, which was employed in earlier experiments. SGD was chosen for fine-tuning because it provides better generalization and avoids the tendency of Adam to converge too quickly to sharp minima, which can lead to overfitting in fine-tuning scenarios.

- **SGD with momentum** helps maintain a balance between exploring the loss landscape and converging efficiently, allowing the model to adapt more robustly to the new datasets.
- In contrast, **Adam** optimizes faster due to its adaptive learning rates but may prioritize speed over stability, potentially leading to less generalizable solutions on smaller datasets.

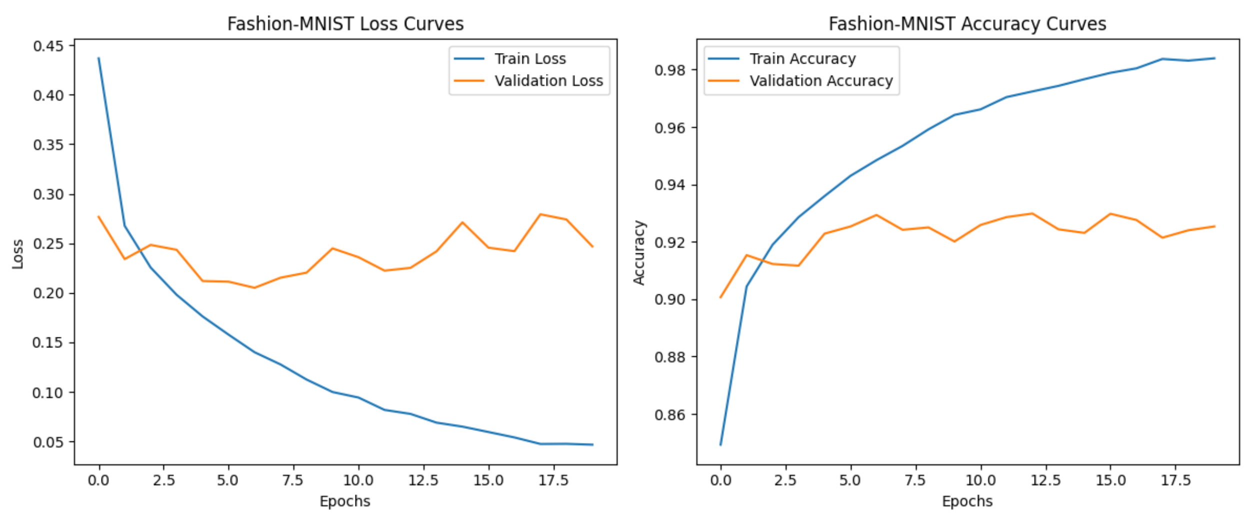
By leveraging SGD, the model achieved smoother convergence and more reliable adaptation, which was particularly beneficial given the distinct characteristics of the Fashion-MNIST and Fruits-360 datasets.

4. **Importance of CIFAR-10 Pretraining**

Pretraining on CIFAR-10 provided ResNet-18 with a strong foundation of general features, such as textures and shapes, at a resolution like the target datasets. This eliminated the need for extensive adaptation and significantly accelerated training. Fine-tuning the model further specialized these features for Fashion-MNIST and Fruits-360, resulting in high classification accuracy. By combining the benefits of pretraining and fine-tuning, the model achieved superior performance while remaining computationally efficient.

B. Loss and Accuracy Curves

a. Fashion-MNIST



1. **Loss Curves:**
   - **Training Loss:** The training loss decreases steadily and consistently, showing the model is effectively learning the patterns in the Fashion-MNIST dataset.

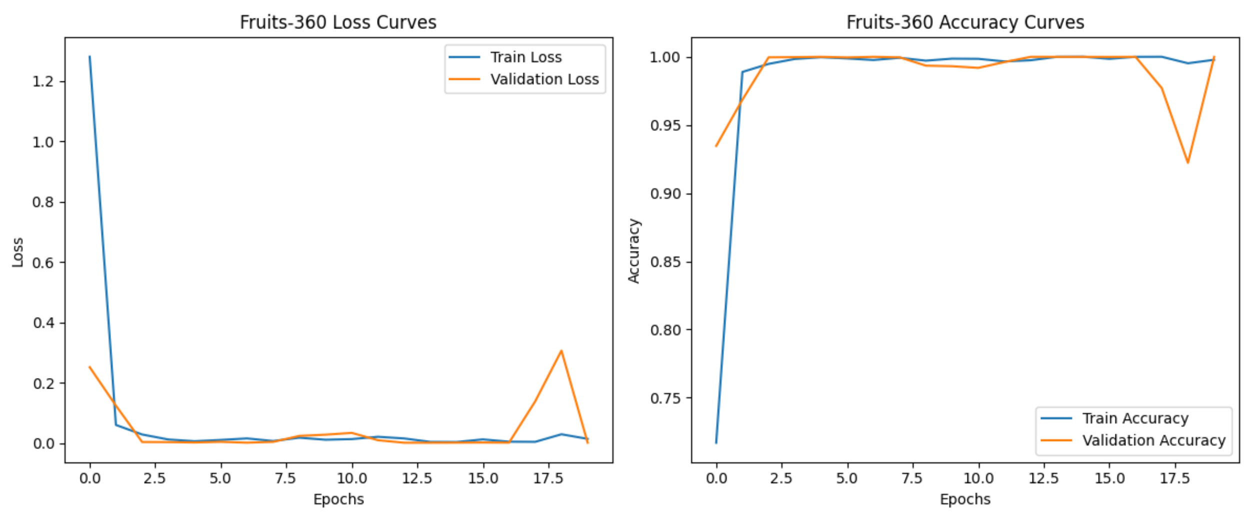
- **Validation Loss:** The validation loss stabilizes after epoch 5, with some fluctuations, suggesting the model generalizes reasonably well but could still improve with fine-tuning of hyperparameters or regularization.

2. **Accuracy Curves:**

   - **Training Accuracy:** The training accuracy increases steadily and reaches nearly 98% by the 20th epoch, indicating the model fits the training data well.
   - **Validation Accuracy:** The validation accuracy plateaus around 92% after epoch 6, suggesting a slight gap between training and validation performance. This indicates some room for improvement in generalization.

Overall, the **ResNet-18 fine-tuned on Fashion-MNIST** achieves excellent performance with strong generalization, as evidenced by the minimal gap between training and validation accuracy curves. However, the model could potentially benefit from additional regularization techniques, such as dropout or data augmentation, to reduce the slight overfitting observed.

b. Fruits-360



1. **Loss Curves:**

   - **Training Loss:** The training loss rapidly decreases and reaches near-zero within the first few epochs, indicating that the model is quickly fitting the training data.
   - **Validation Loss:** The validation loss initially follows a similar trend but shows instability towards the later epochs, likely caused by overfitting or insufficient regularization. The spike at epoch 18 suggests that the model may struggle with certain samples in the validation set.

2. **Accuracy Curves:**

   - **Training Accuracy:** The training accuracy rapidly approaches 100%, confirming the model's strong capability to learn the patterns within the training data.

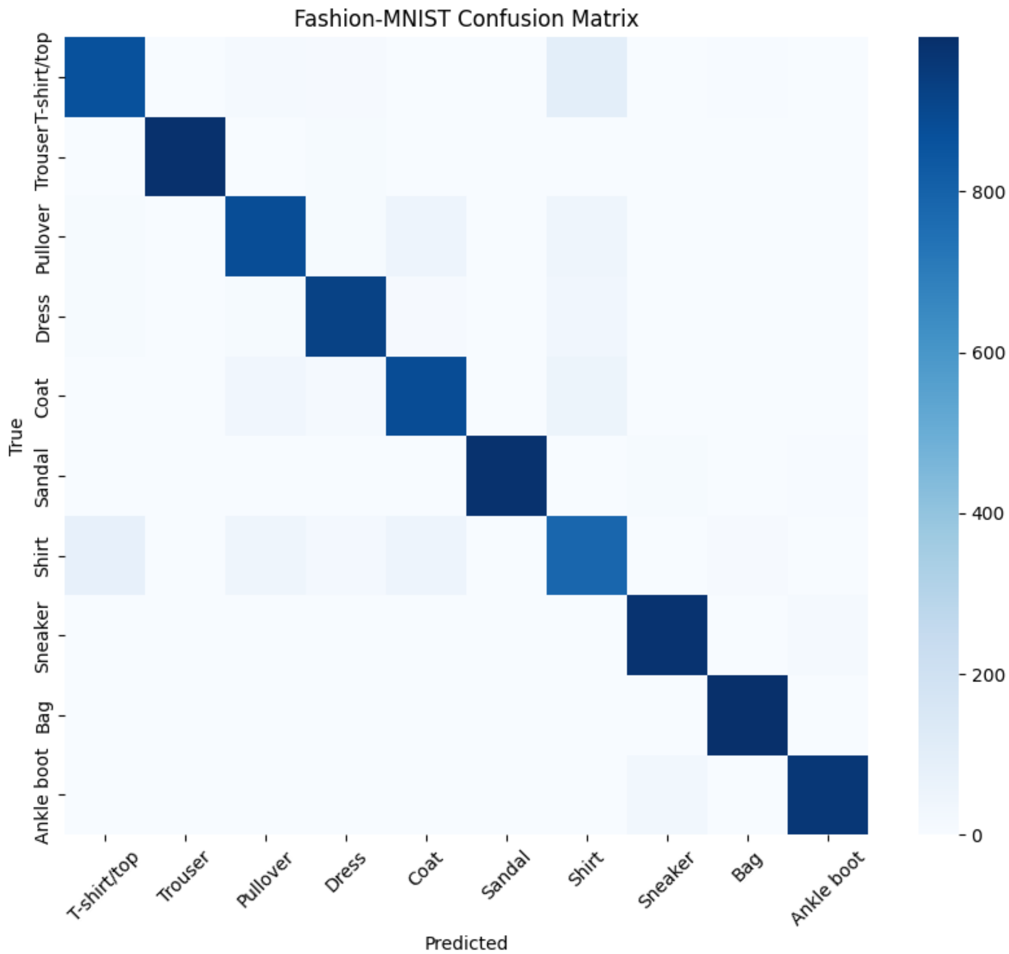
- **Validation Accuracy:** The validation accuracy closely mirrors the training accuracy for most of the epochs but exhibits noticeable fluctuations in later epochs, dropping significantly at times, which indicates potential overfitting or sensitivity to validation data variations.

The **ResNet-18 fine-tuned on Fruits-360** demonstrates excellent initial performance, with high training and validation accuracies achieved early on. However, the instability in validation loss and accuracy towards the later epochs indicates a risk of overfitting. To address this, additional regularization (e.g., increased dropout or data augmentation) or early stopping could improve the model's generalization performance on unseen data.

C. Evaluation

1. Fashion-MNIST

Class	Precision	Recall	F1-Score	Support
T-shirt/top	0.89	0.86	0.88	1000
Trouser	0.99	0.99	0.99	1000
Pullover	0.88	0.88	0.88	1000
Dress	0.93	0.92	0.93	1000
Coat	0.88	0.89	0.88	1000
Sandal	0.99	0.98	0.99	1000
Shirt	0.76	0.78	0.77	1000
Sneaker	0.96	0.98	0.97	1000
Bag	0.98	0.99	0.98	1000
Ankle boot	0.97	0.96	0.97	1000



- **Accuracy**: 0.92
- **Macro avg (Precision, Recall, F1-Score)**: 0.92, 0.92, 0.92
- **Weighted avg (Precision, Recall, F1-Score)**: 0.92, 0.92, 0.92
- **Speed**: 7 min

The ResNet-18 model outperformed both the MLPs and CNNs in terms of accuracy on the Fashion-MNIST dataset, achieving a final accuracy of 0.92. This is higher than the CNN without augmentation (0.90), the CNN with augmentation (0.87), the MLP on flattened images (0.88), and the MLP on extracted features (0.82). The ResNet-18's superior performance can be attributed to its pretraining on the CIFAR-10 dataset, which allowed it to learn powerful and transferable features, further refined during fine-tuning for Fashion-MNIST.

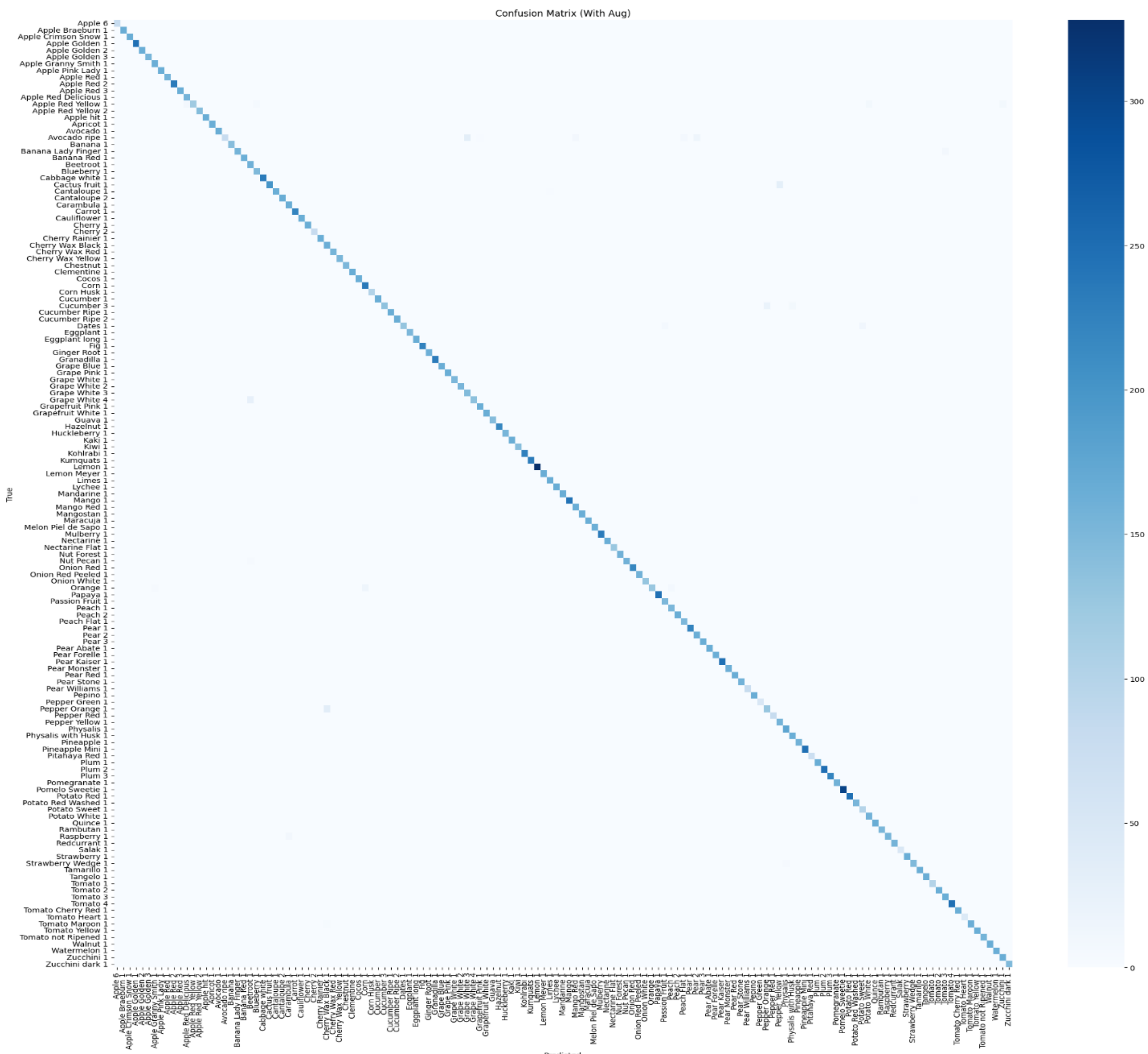
In terms of generalization, ResNet-18 demonstrated minimal overfitting, as seen by the tight alignment between training and validation accuracy, a result of both the use of fine-tuning and the SGD optimizer. The MLPs and CNNs had slightly larger gaps between their training and validation accuracies, suggesting they had less effective feature extraction mechanisms compared to the pre-trained ResNet-18.

When considering training speed, the MLP on flattened images was the fastest, completing training in just 47 seconds, followed by the feature-based MLP at 1 minute. The CNN without augmentation also trained quickly, completing in 1 minute, while the ResNet-18 required approximately 7 minutes for fine-tuning. The CNN with augmentation had the slowest training time at 8 minutes, due to the additional computational overhead of generating augmented data.

The results show that while MLPs and CNNs provide competitive accuracies and faster training times, the pre-trained and fine-tuned ResNet-18 stands out by achieving the highest accuracy and robust generalization, making it the most effective model for the Fashion-MNIST dataset.

2. Fruits-360

Class	Precision	Recall	F1-Score	Support
Apple 6	1.00	1.00	1.00	80
Apple Braeburn 1	1.00	1.00	1.00	166
Apple Crimson Snow 1	1.00	1.00	1.00	166
Apple Golden 1	1.00	1.00	1.00	246
Apple Golden 2	1.00	1.00	1.00	164
Avocado ripe 1	1.00	0.55	0.71	150
Beetroot 1	0.83	1.00	0.91	161
Cherry Wax Black 1	0.81	1.00	0.90	164
Grape White 3	0.81	1.00	0.90	146
Grape White 4	1.00	0.83	0.91	164



- **Accuracy**: 0.99
- **Macro Avg (Precision, Recall, F1)**: 0.99, 0.99, 0.99
- **Weighted Avg (Precision, Recall, F1)**: 0.99, 0.99, 0.99
- **Speed**: 9 min

The ResNet-18 model demonstrated superior performance on the Fruits-360 dataset, achieving an impressive accuracy of 0.99 in just 9 minutes of training. This outperforms both the MLPs, which achieved accuracies of 0.91 (on features) and 0.90 (on flattened images), and the CNNs, which reached 0.98 and 0.96 accuracy. ResNet-18's pretraining on CIFAR-10 and fine-tuning allowed it to leverage its deep architecture and extract complex features, providing a significant advantage over the simpler MLPs and CNNs.

In terms of training time, ResNet-18 (9 minutes) was faster than the CNN with data augmentation, which required 23 minutes. However, it was slightly slower than the CNN without augmentation, which trained in 8 minutes. Compared to the MLPs (55 seconds for features and 1 minute for flattened images), ResNet-18 required considerably more time due to its depth and complexity.

This highlights ResNet-18's strength in achieving near-perfect accuracy with moderate computational cost, outperforming simpler models while maintaining efficiency relative to augmented CNNs.

5. Conclusion

This project demonstrated the application of various neural network architectures, including MLPs, CNNs, and ResNet-18, for image classification on the Fashion-MNIST and Fruits-360 datasets. The ResNet-18 model, fine-tuned from CIFAR-10 pretraining, achieved the best overall performance, leveraging its deep feature extraction capabilities to deliver near-perfect accuracy on Fruits-360 and high accuracy on Fashion-MNIST, while maintaining a reasonable training time. This underscores the power of transfer learning and fine-tuning in modern machine learning pipelines, highlighting the balance between computational efficiency and predictive accuracy across different tasks.