Image Classification Analysis
MLP, LeNet-5, and ResNet

Mîrza Ana-Maria, 341C1

Faculty of Automation and Computer Science

Keywords: Machine Learning, Mutltilayer Perceptron, Convolutional Neuron Network, LeNet-5, ResNet

1 General Description

In this article, we will explore the results obtained on training a multilayered perceptron and a deep convolutional network on the datasets Fruits-360 and Fashion-MNIST. We will compare the results with models such as logistic regression, SVM and gradient boosting trees, as well as a pre-trained ResNet model that is finetuned on the datasets.

The general hyperparameters used for the models include:

- Learning rate: 0.001
- Batch size: 256
- Number of epochs: 200 (20 for ResNet18)
- Optimization algorithm: SGD with momentum (Adam for DeepConvNet)
- Error function: Cross Entropy

2 Models

As mentioned earlier, the analysis was performed on three models - a multilayer perceptron, a deep convolutional network, and a ResNet18. The architectures and parameters used are detailed in the following sections, as they have a big impact on the model's performance and the results obtained.

2.1 MLP

First model used for training in the project is a multilayered perceptron containing 3 layers for input, hidden, and output. The general architecture has the following structure:

Layer	Activation Function	Number of Neurons
Input	ReLU	Input Size
Hidden	ReLU	256
Output	—	Number of Classes

Table 1: Neural Network Architecture

The number of classes depends on the number of classes in the datasets. For the latter case, there are 10 classes for the Fashion-MNIST dataset and 70 for Fruits-360.

2.2 DeepConvNet - LeNet-5

For the deep convolutional network, I chose to implement a well-known architecture and compare its performance with MLP and ResNet. Several modifications to the architecture were made to increase accuracy and reduce underfitting and overfitting. Since the model showed signs of underfitting on the full images, by having an accuracy of 10%, the optimizer was first changed from SGD to Adam. This drastically changed the accuracy of the model on the training set to 80%, but the scores on the test set were still low 50%. This case of overfitting was solved by adding dropout layers and removing a linear layer at the end, leading to accuracies of 80% on testing set as well.

2.2.1 Architecture

The architecture of the model is described in detail below.

Layer	Operation	Image Size / Output Shape	Additional Details
1	Conv2D(3, 6, k=5, padding=2) + LeakyReLU	28x28	negative_slope=0.01
2	AvgPool2D(2, 2)	14x14	—
3	Dropout2D(0.2)	14x14	dropout_rate=0.2
4	Conv2D(6, 16, k=5) + ReLU	10x10	—
5	AvgPool2D(2, 2)	5x5	—
6	Dropout2D(0.2)	5x5	dropout_rate=0.2
7	Linearize(16, 5, 5)	16 * 5 * 5	—
8	Fully Connected (FC): 16 * 5 * 5 → 120 + LeakyReLU	120	negative_slope=0.01
9	Fully Connected (FC): 120 → Number of Classes	Number of Classes	—

Table 2: DeepConvNet Model Architecture

2.2.2 Data Augmentation

For the sake of the experiment, we tested the model on the augmented dataset and observed the changes in the results. Here are the augmentation methods used:

- **Rescaling**:
  - `transforms.Resize((32, 32))` – Rescales all input images to a fixed size of 32x32 pixels.
- **Random Horizontal Flip**:
  - `transforms.RandomHorizontalFlip()` – Applies a random horizontal flip to the input image.
- **Random Rotation**:
  - `transforms.RandomRotation(10)` – Rotates the input image randomly within a range of ±10 degrees.
- **Random Crop with Padding**:
  - `transforms.RandomCrop(size=(32, 32), padding=(4, 4), padding_mode="reflect")` – Applies a random crop of size 32x32 pixels after padding the image with 4 pixels on each side using the `reflect` mode.

2.3 Pre-trained Model - ResNet18

Last model used is a pre-trained model on the CIFAR-10 data set, ResNet18. The model can be found and imported from the CIFAR-10 github repository. This model was used for finetuning with images from the two data sets, rescaled to 32x32.

3 Datasets

3.1 Fruits-360

3.1.1 Data Analysis

The Fruits-360 dataset is comprised of 70 classes of fruits, vegetables and nuts, with approximately 71k trainig and 21 test data set samples.

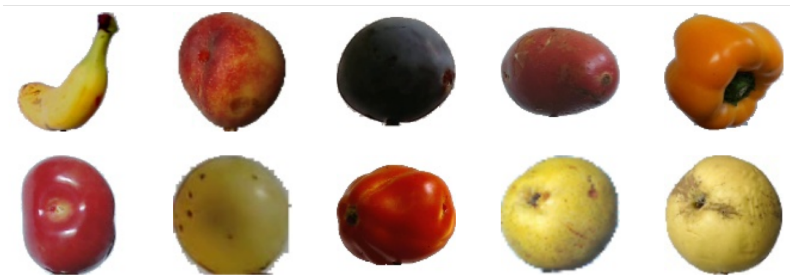

Figure 1: Top 10 fruits

This figure shows samples of images from the dataset with the 10 most frequent fruits. The images were resized to 32x32 in order to fit in the memory.

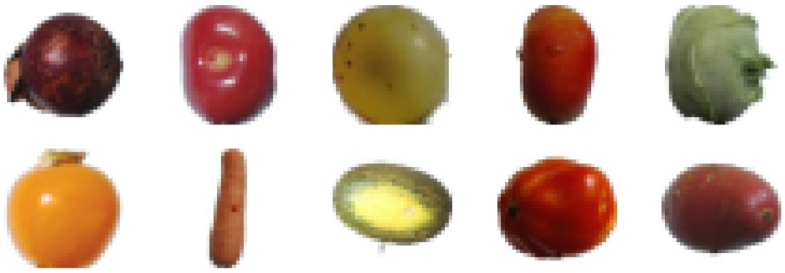

Figure 2: Top 10 fruits resized 32x32

The frequency of the data set can be observed that it is not uniform, which might have inhibited models to learn the data set to it's full potential.

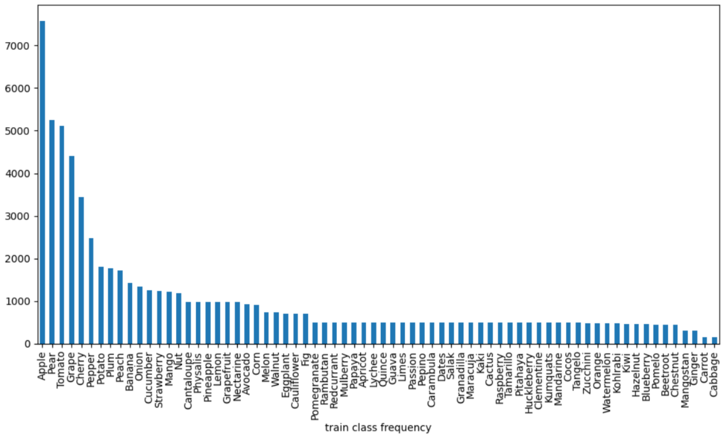

Figure 3: Class

3.1.2 Extracted Features

In a previous iteration, the dataset was used for training other models and several feature extraction algorithms were used in order to down sample the dataset. We will use the best one to train the MLP and compare the results with the models trained on the whole images. The feature extraction algorithm used is PCA with 70 principal components.

3.1.3 Model Results

The results obtained on the MLP for the features extracted are displaying a little overfit on the test data set, but overall the results were good.

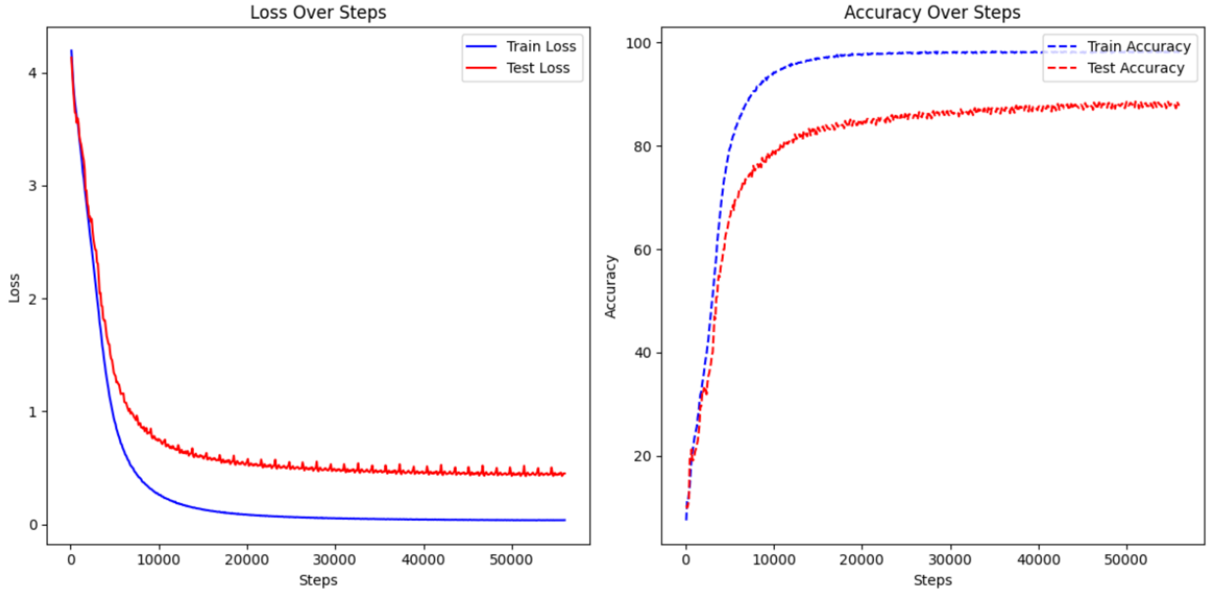

Figure 4: MLP on Fruits-360 Extracted Features

Next, the same mlp model, whose architecture we described in the first section, was trained on the full images of the Fruits-360 dataset. The results had lesser overfit, and better accuracy (from 87% to 90%).

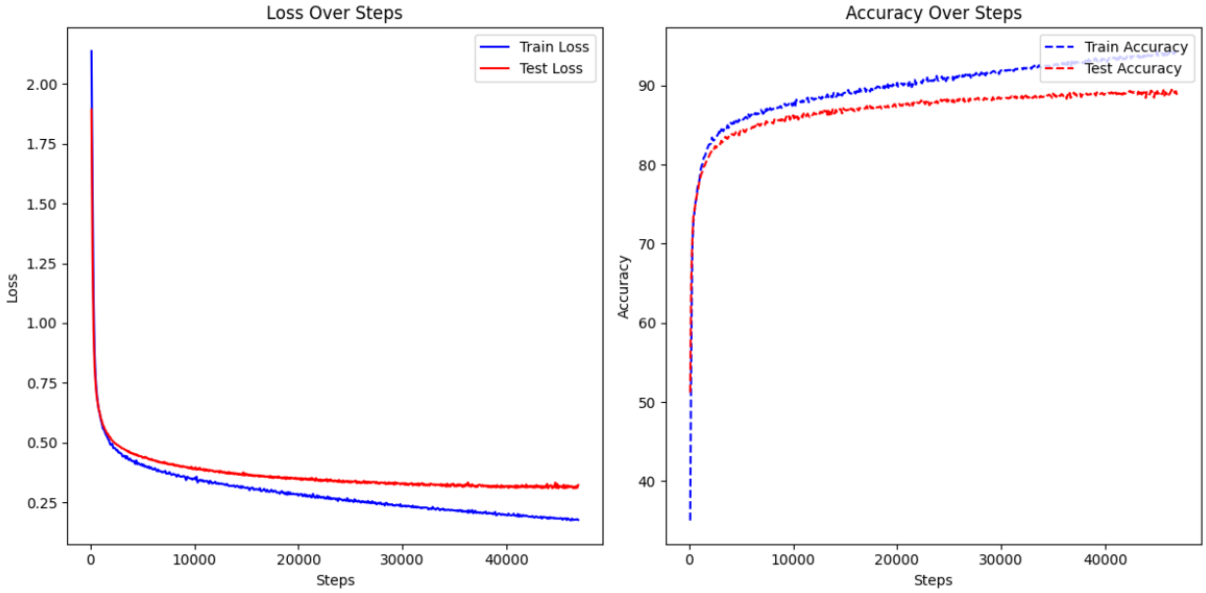

Figure 5: MLP on Fruits-360 Images

The results obtained on the model trained with the extracted features were 24.29% better than the worst model (logistic regression) and worse by 5.43% compared to best model (SCM). Similarly, the results obtained on the mlp model trained with the linearized images were 28.57% better than logistic regression, and 2.17% worse than SCM.

Training the LeNet-5 model on the linearized images was the most time-consuming out of all models, taking 9h and 12h (on data without augmentation and with augmentation respectively).

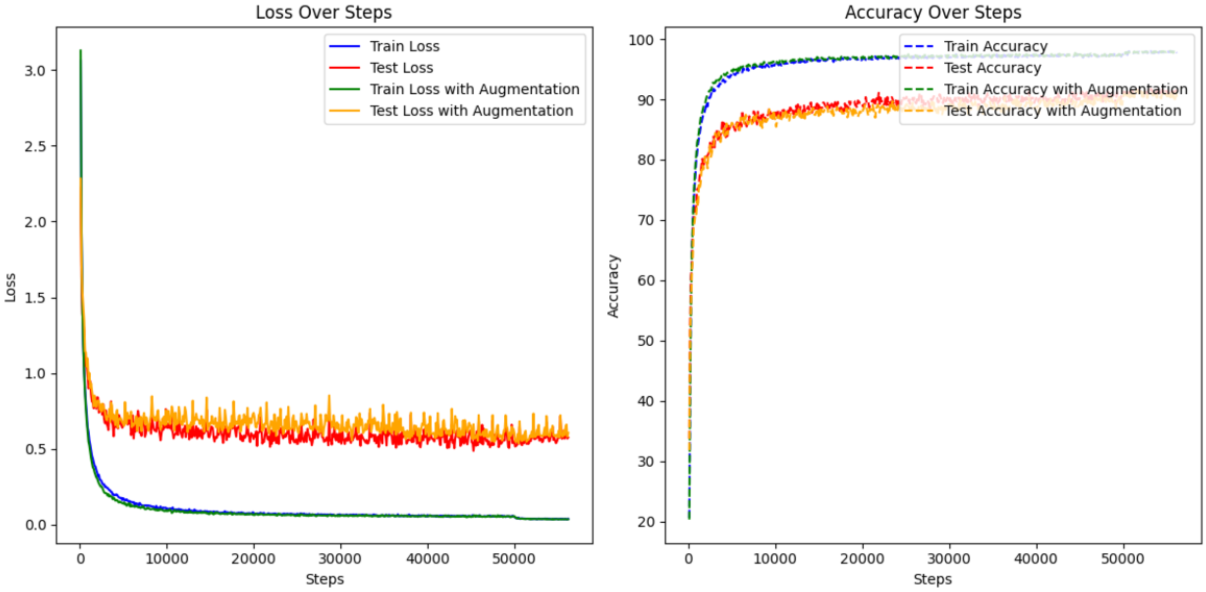

Figure 6: LeNet-5 on Fruits-360 Images

The results obtained indicate almost no overfitting, but a distinct impact caused by the data augmentation.

On figure 6, the results obtained on the finetuning of the ReNnet18 model are presented. Having taken only 2-3 hours for 20 epochs and having the best performance on the data set of 97%, which is 5.43% better than the best results obtained on svm.

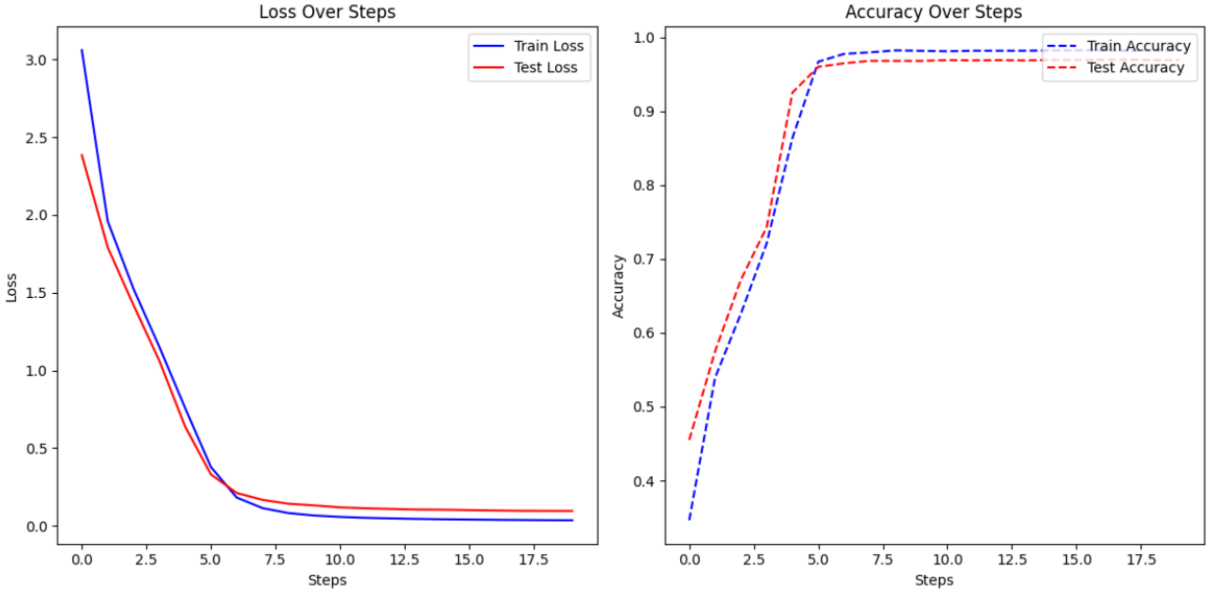

Figure 7: ResNet18 on Fruits-360 Images

3.2 Fashion-MNIST

Fahsion-MNIST data set contains 28x28 images of 10 type of clothing items. The trainig set contains 60k images, and the test set contains 10k.

3.2.1 Data Analysis

A subset of the images can be seen in the following features.

Figure 8: Fashion-MNIST

3.2.2 Extracted Features

The algorithm for features extracted from the last iteration, that gave the best results were HOG + PCA with 20 principle components. The best results obtained on these attributes were 87.54% on SVM and 80.98% worst results on logistic regression. More details on this Image-Classifier Github repository.

3.2.3 Model Results

The first results obtained for this data set were on the MLP architecture, on features extracted with the methodology described above, having 2.47% better accuracy than worst results obtained on logistic regression, and 4.60% worse results compared to SVM, but much faster (10 min).

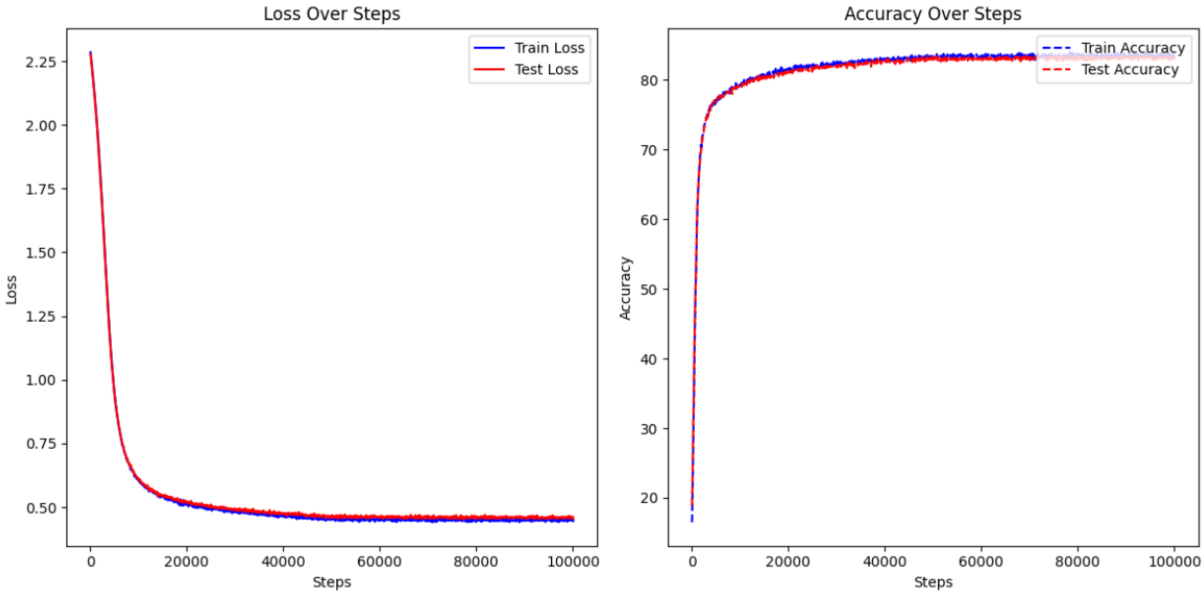

Figure 9: MLP on Fashion-MNIST Extracted Features

Training the MLP model on the full images (linearized) led to better results by 9.88% and 2.30% than logistic regression and SVM respectively.

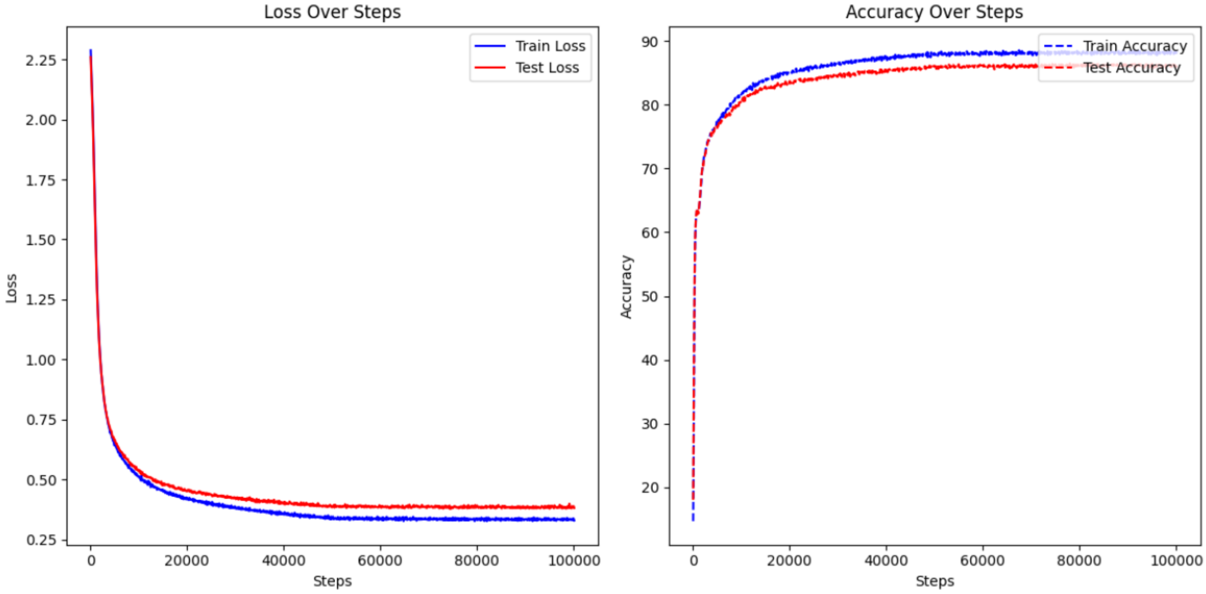

Figure 10: MLP on Fashion-MNIST Images

Results obtained on the LeNet-5 architecture were interesting, having test accuracies and loss scores better than the training set on data with augmentation. Some overfitting can also be observed on the datasets.

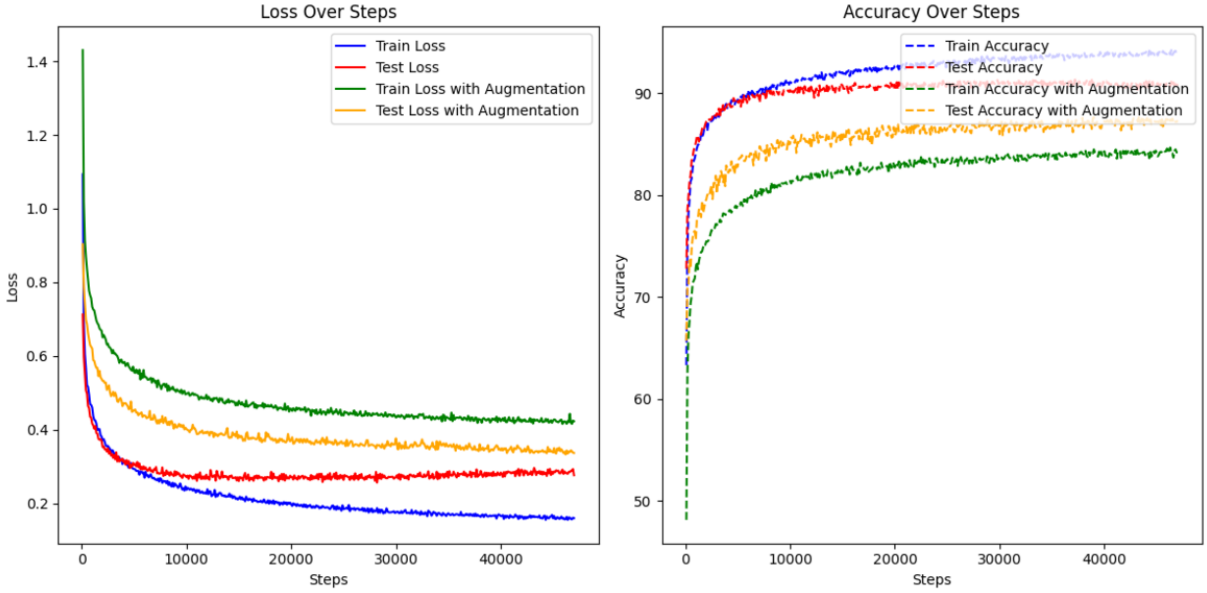

Figure 11: LeNet-5 on Fashion-MNIST Images

Several factors that could have contributed to this situation might have been unequal distribution of classes in the test and training sets or too much regularization, according to this source on stackoverflow: Test-accuracy-is-grater-than-train-accuracy.

Lastly, results obtained on the finetuning of the resnet18 model have an accuracy of 94% on the test set, which was 16.05% better than logistic regression and 8.05% better than svm.

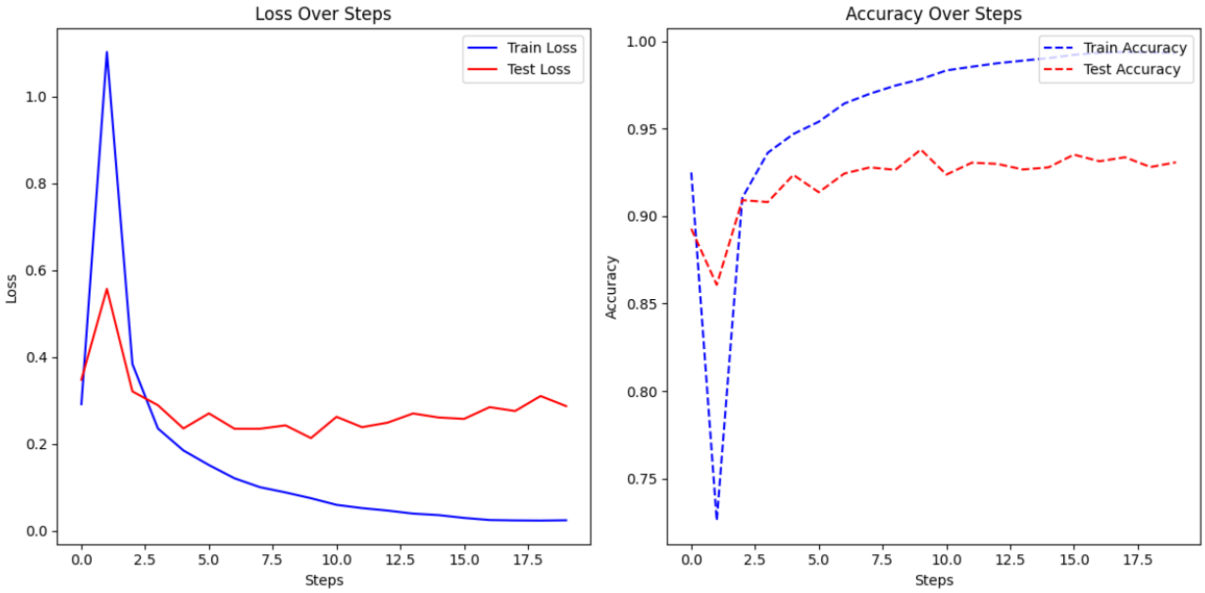

Figure 12: ResNet18 on Fashion-MNIST Images

The graph of the accuracies and loss show some irregular slopes. This might be because of running the training for only 20 epochs, but since the training took about 3 hours, we didn't run it again with more epochs.

4 Results Summary

A summary of the results obtained on the described models for this analysis, as well as models used for the previous iteration can be seen more clearly in the tables below.

Model	**Dataset**	**Input**	**Accuracy on Test**
MLP	Fashion-MNIST	Attributes: 16	83.10%
MLP	Fashion-MNIST	Image Size: 28x28	89.05%
LeNet-5	Fashion-MNIST	Image Size: 28x28	90.63%
LeNet-5	Fashion-MNIST	Image Size: 28x28 + Aug.	87.55%
ResNet-18	Fashion-MNIST	Image Size: 28x28	93.80%
MLP	Fruits-360	Attributes: 70	87.98%
MLP	Fruits-360	Image Size: 32x32	90.93%
LeNet-5	Fruits-360	Image Size: 32x32	91.02%
LeNet-5	Fruits-360	Image Size: 32x32 + Aug.	90.42%
ResNet-18	Fruits-360	Image Size: 32x32	96.97%

Table 3: Model performance on different datasets and inputs.

Model	**Attribute Selection Algorithm**	**Fruits-360**	**Fashion**
Logistic Regression	HIST	61.78%	-
	PCA (70 PC)	71.78%	-
	HOG - PCA (20 PC)	-	80.98%
SVM	HIST	70.89%	-
	PCA (70 PC)	92.35%	-
	HOG - PCA	-	87.54%
Random Forest	HIST	59.46%	-
	PCA (70 PC)	87.31%	-
	HOG - PCA (20 PC)	-	85.29%
Gradient Boosted Trees	HIST	59.38%	-
	PCA (70 PC)	81.16%	-
	HOG - PCA (20 PC)	-	85.83%

Table 4: Results for Different Models and Attribute Selection Algorithms

Comparing the different results on a time basis, the LeNet-5 model took the longest due to much more layers in the model - taking up to 12 hours on Fruits-360 data set. If we would have ran the ResNet18 model for 200 epochs as well, it would probably have beaten LeNet-5 on time, since it too has a large number of layers and it took around 2-3 hours for only 20 epochs. Nevertheless, the resuls obtained are better on more complex architectures such as ResNet18. For a more visible compararison, below is a table with precision, recall, F1 score, accuracy of models.

Dataset	Input Type	Model	Data Split	Precision	Recall	F1 Score	Accuracy	Time (min)
Fashion-MNIST	HOG-PCA20 Attributes	SVM	Test	0.87	0.88	0.87	0.8754	30
Fashion-MNIST	HOG-PCA20 Attributes	MLP	Test	0.8303	0.831	0.8303	0.831	8
Fashion-MNIST	Linearized Images	MLP	Test	0.8900	0.8905	0.8898	0.8905	96
Fashion-MNIST	Linearized Images	LeNet-5 w/o Aug.	Test	0.9066	0.9063	0.9061	0.9063	147
Fashion-MNIST	Linearized Images	LeNet-5 w/ Aug.	Test	0.8755	0.8755	0.8751	0.8755	224
Fashion-MNIST	Linearized Images	Finetuned ResNet18	Test	0.9378	0.938	0.9378	0.938	200
Fruits-360	PCA70 Extracted Attributes	SVM	Test	0.93	0.92	0.92	0.9235	30
Fruits-360	PCA70 Extracted Attributes	MLP	Test	0.8846	0.8799	0.8791	0.8799	350
Fruits-360	Linearized Images	MLP	Test	0.9061	0.9094	0.9011	0.9094	435
Fruits-360	Linearized Images w/o Aug.	LeNet-5	Test	0.9082	0.9102	0.9033	0.9102	567
Fruits-360	Linearized Images w/ Aug.	LeNet-5	Test	0.9046	0.9043	0.8994	0.9043	738
Fruits-360	Linearized Images	Finetuned ResNet18	Test	0.9754	0.9698	0.9692	0.9698	230

Table 5: Detailed Results for Different Datasets, Input Types, and Models

5 Conclusion

Based on the models trained with the datasets, we were able to extract several conclusions resulted to the impact of the input, data augmentation, and model architecture.

1. Best Model: ResNet18 is the clear winner for both datasets and input types.

- **ResNet18 vs. LeNet-5:**
  - +3.5% (Fashion-MNIST)
  - +6.5% (Fruits-360)

2. Impact of Input: Using image-based inputs consistently yields better performance.

- **Image Input vs. Attributes:**
  - +7.2% (Fashion-MNIST)
  - +3.4% (Fruits-360)

3. Future Improvements: Experimenting with different augmentation techniques or fine-tuning hyperparameters could further improve performance, especially for LeNet-5 on Fashion-MNIST.

- **Augmentation Impact on Fashion-MNIST:** -3.4% (LeNet-5)
- **Augmentation Impact on Fruits-360:** -1.11% (LeNet-5)

In conclusion, based on the individual need, one can chose the appropiate architecture. SVM is overall the best model to use when resources are scarce, having the capacity of giving very good results with as little as 16 attributes extracted from an image. If higher accuracy and precision is desired, one can use a pre-trained model, as it can lead to better results, but with higher computational power.