Analysis of Various Classification Methods on the Fashion MNIST Dataset

Shaashvat Shetty[1]

[1]University of Maryland, College Park

Introduction

In this project, various different classification methods on the Fashion MNIST dataset were analyzed. The Fashion-MNIST data is a collection of 60,000 grayscale fashion items with 10,000 testing images. However, in this project the models will be trained on 1,000 training samples (to keep computations simple), but the entire 10,000 testing dataset is still used. The different classification methods include KNN, Linear SVM, Nonlinear SVM, Random Forest, LDA and QDA, and CNN. This paper will be describing the methods used as well as explain the different approaches. The data the models will be trained and tested on are either the flattened data and/or the PCA data. The flattened data is 784 dimension flattened data. Whereas the PCA data is 69 dimensions. This was enough dimensions to explain 90% of the variance. The reason why PCA data is included is because it helps in reducing the computational costs by using dimension reduction on the original flattened dataset. This can improve the performance of the model. In most of the classification methods I decided to go with the PCA data because it is less computationally intensive. Note: for the upcoming sections you may refer to figures 6 and 7 on per class and total accuracies for each model. However, Part 6 of this paper will discuss the tables in more depth.

PART 1: KNN Classification

In KNN classification, the model works by finding the k closest data points in the training data for an input. It then classifies the point based on the majority class/average near it among the neighbors. For the KNN classification, two different models were trained, one on the flattened dataset and another on the PCA data. NOTE: For BOTH approaches, the hyperparameter k is tuned using 5-fold cross validation. The following values of k are: [1,3,5,7,9,11]. By using CV, the optimal k value leading to the highest testing accuracy can be obtained.

The results of the Flattened Data are as follows: The best parameter was k = 1 which led to a total accuracy of 75.06%. Certain classes like Trouser, Sneaker, Bag, and Ankle Boot did considerably well with accuracies 93.20%, 92%, 90.6%, and 94.1% respectively. The class with the lowest accuracy was Shirt, being 49.5%.

The results of the PCA Data are as follows: The best parameter was k = 1 which led to a total accuracy of 75.59%. Similar to the flattened dataset, it also performed the best on Trouser, Sneaker, Bag, and Ankle Boot with accuracies 93.3%, 90.4%, 92.7% and 94.2% respectively. Shirt had the worst accuracy at 47.7%.

Overall, in terms of performance, both models seemed to have very similar results; refer to Fig.1. Both of the models performed about equally the same with PCA model performing slightly better as seen in how the bars are relatively the same height with PCA accuracy being slightly higher in most classes. The x axis which goes from 0 - 9 are the 10 different classes. In the following order: ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']. It is recommended to use the PCA data over the flattened data because it is more computationally efficient to train on and the total accuracy is slightly better (75.59% vs 75.06%). Note: that despite both models performing the best with k = 1, it is advisable to not use such a low parameter because of the risk of overfitting the model to the training data. The reason for a small k value being optimal has to do with the fact that only 1000 training samples are used. When the models are instead trained on 3000 training samples, the optimal k values are 5 and 7 for the flattened and PCA data respectively.

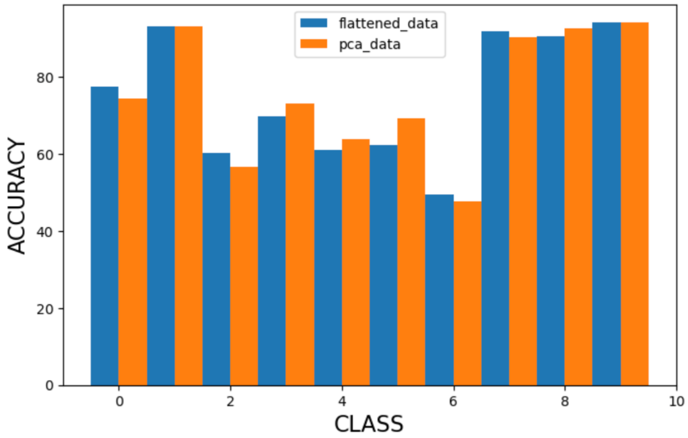


Fig. 1. Similar accuracies between Flattened Data and PCA

PART 2: Linear SVM Classification on PCA data

The next approach was to use a Linear SVM classification. SVMs are used to find the optimal separating hyperplane for distinguishing between different groups for classification. The reason for training a Linear SVM only on the PCA data is because it is only 69 dimensions as opposed to 784. Attempting to create an optimal separating hyperplane for 784 dimension data would be computationally heavy so it is important to use dimension reduction. The process is as follows: A linear kernel is used, this assumes that the data is somewhat linearly separable. By choosing a linear kernel over a nonlinear kernel we also reduce the risk of overfitting the model to noise. The hyperparameter is C which is used to control how much misclassifications occurs in the training data. A larger value of C indicates a hyperplane with a smaller margin, so an error on the training data is penalized more. A smaller value of C indicates a hyperplane with a larger margin, so an error on the training data isn't penalized as harshly. 5 fold cross validation was used with C values of [0.001,0.01,0.1,10.0,100.0] in order to obtain the optimal C that led to the highest testing accuracy. The model that performed the best was with C being 0.01. Resulting in a total accuracy of 79.58%. However, Shirts get misclassified often (50.1%). To understand why shirts get misclassified often, refer to the confusion matrix in Fig. 2. Shirts get confused with T-Shirts 20% of the time. Pullovers are classified correctly 62% of the time but also get confused with Shirts 15% of the time. However, they don't get misclassified as T-shirts that often (1.7% of the time). These results seem to make sense because the three classes Shirts, T-Shirts, and Pullovers look visually similar to each other.

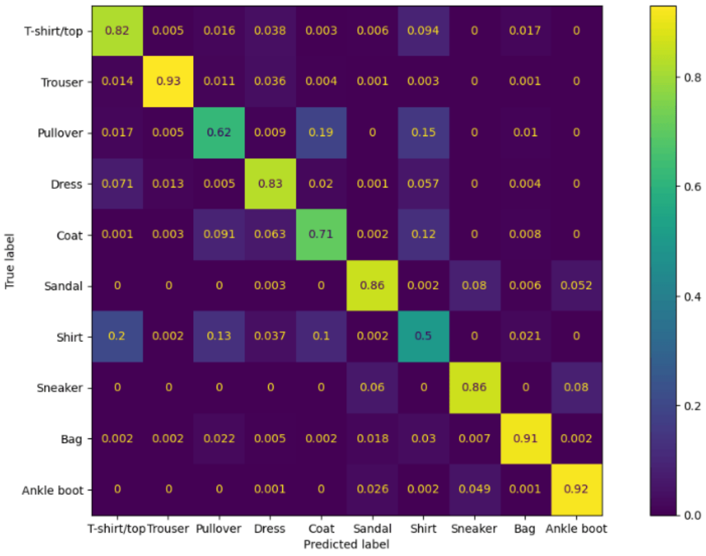


Fig. 2. Confusion Matrix for Linear SVM on PCA data

PART 3: Nonlinear SVM classification on PCA data

The next approach was to use a Nonlinear SVM Classification on the PCA data. The Nonlinear approach differs itself from the Linear approach in that it uses a kernel function to transform the data into a higher dimensional space where the data becomes more linearly separable. In doing so it is possible that the model becomes more accurate at classification. But this also means that the model is more prone to overfitting than the Linear SVM approach. The kernel function that was chosen is the RBF kernel which is also the most commonly used kernel function. The equation for the RBF kernel is as follows:

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \qquad \textbf{(1)}$$

Note that in the equation above $\frac{1}{2\sigma^2}$ is the same as γ

The different values for the hyper parameter C are the same as what was used for the linear SVM approach. However, since the RBF kernel is being used, the hyperparameter γ as mentioned above needs to be included. Larger values of γ leads to more flexible boundaries, thus possible overfitting. Whereas smaller values of γ mean rigid boundaries so possible underfitting. The following values of γ were tested on: [0.0001,0.001,0.01,0.1,1,1.0]. 5 fold cross validation was used over the C values (same as with the linear SVM approach) and γ values in order to find the optimal parameters for both that would lead to highest testing accuracy. The results are as follows: Optimal C and γ are 10.0 and 0.001 respectively. The total accuracy is 80.74% with certain categories such as Trousers, Bags, and Ankle Boots performing well with scores 94.2%, 92.7%, and 92.7% respectively. Shirt still continues to get misclassified often, with an accuracy of 50.5%. These results can be seen in Fig.3. (trained on optimal gamma of 0.001) Classes with high levels of accuracy seem to have a clear decision boundary with the exception being Bags. It is possible that Bags are more distinguishable in higher dimensions however, Fig.3. only looks at the first 2 PCA components. What we can reason from the figure is that classes that are the most visually similar seem to be clustered near each other. For example, Pullovers (green) are situated in the red region belonging to Coat. Or Bags (yellow) are often classified as Ankle boots (brown).

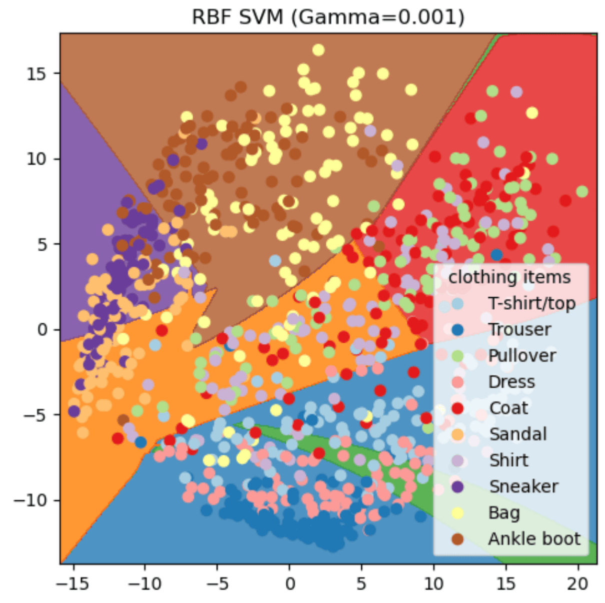


Fig. 3. Decision Boundaries for RBF SVM

PART 4: Random Forrest on PCA data

The next method is to train a Random Forrest model on the PCA data. The model was trained with 500 estimators. This means that there are 500 trees in the forest. Increasing this hyperparameter can lead to a higher training accuracy but it is also more computationally heavy and more prone to overfitting thus performing poorly on the test data. It was trained with a maximum of 8 features to consider when looking for the best split. The reason for choosing 8 features is because it is approximately the sqrt of the number of features in the PCA dataset (69). This is a common approach to reduce overfitting. The results are as follows: The Out Of Bag (OOB) test accuracy estimate was 82.2%.This is an estimate of how the model may do on unseen data by essentially treating the training data as if it were test data. This OOB accuracy is close to the total accuracy of 78.73%. Since both values are close together this means that the model has done a good job at generalizing the training data. If the OOB test accuracy was significantly higher then this is a sign that the model may have over fitted the training data. The model performed well on classes like Trousers, Bags, and Ankle Boots (91.7%, 91.5%, and 92% respectively). However this model performed poorly on classifying shirts (47%). According to the confusion matrix in Fig 4, shirts get misclassified with T-shirts often (23% of the time). However, the converse is not true, T-shirts are correctly identified 82% of the time and gets confused with shirts 9% of the time. It is possible than T-shirts are included more often in the splits that shirts which would make sense as to why T-shirts are classified correctly often but shirts get misclassified as T-Shirts.

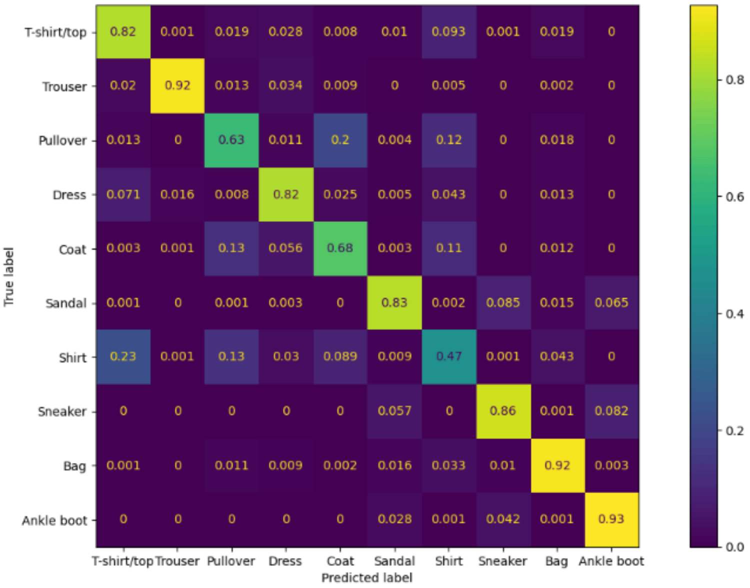


Fig. 4. Confusion Matrix Random Forrest PCA data

PART 5: LDA and QDA on PCA data

The next method is to do Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). In LDA the goal is to separate the various classes with a linear decision boundary. It works best when the number of classes is small or the classes are well separated. LDA uses LDA components (which are the axes) in order to explain which direction to travel in to best separate the data. The maximum number of LDA components is the number of classes -1. In the PCA data there are 9 LDA components (because 10 classes), their explained variances are as follows [0.44811484 0.2101604 0.08402945 0.07624884 0.06689878 0.04970227 0.03934738 0.01766795 0.0078301]. This tells you how much between class variance is explained by each of the components/axes. In this case the first 2 components explain most of the variance. Refer to Fig.5. which is a visualization of the first 2 LDA components. Note the X-axis is LDA 1 and the y-axis is LDA 2. According to the scatter plot, classes like Trouser and Bags were separated from the other classes well, with few overlap. This coincides with the results, 91.20% accuracy for Trouser and 88.70% for Bags. However, T-shirts which other models did a good job at separating isn't well separated from other classes like Coat and Shirt and this explains why the T-shirt classification accuracy is low at 79%. QDA differs from LDA in it creates a covariance matrix for each class (69x69 matrix for each of the 10 classes) unlike LDA which assumes that all classes have equal covariances (one 69x69 matrix). By doing so it creates quadratic boundaries rather than linear ones. The QDA model performed poorly with a total accuracy of 71.89%. Classes such as Bag and Ankle boot had high accuracies at 95.80% and 92.60% respectively. Whereas, classes like Pullover and T shirt performed the worst at 22.2% and 62.9% respectively. However the model didn't perform as bad in certain classes like Shirt and Bag indicating that these classes may be more quadratically separable. The poor performance could be an indication that the models could have covariance matrices similar to each other. QDA also tends to do worse when training on smaller datasets because with larger datasets the covariance matrix for each class becomes less precise. In this case, it could be performing poorly because the training size is only 1000.

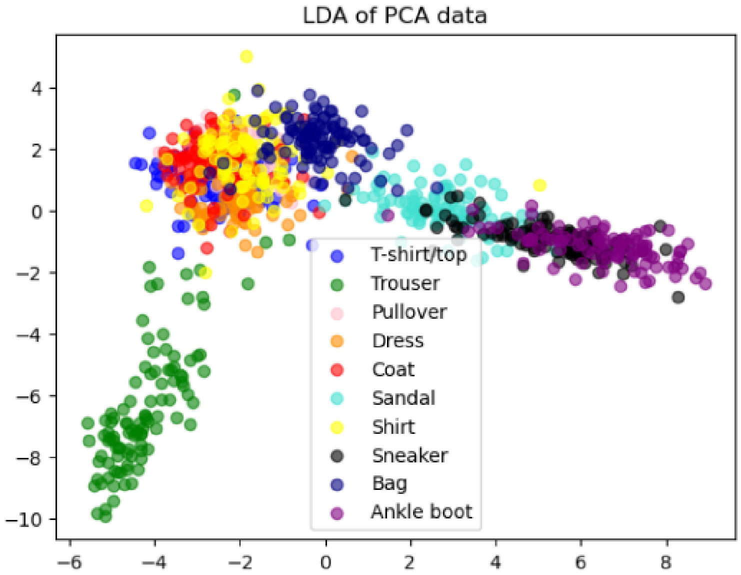


Fig. 5. LDA on first 2 LDA components

PART 6: Comparison of non-DL models

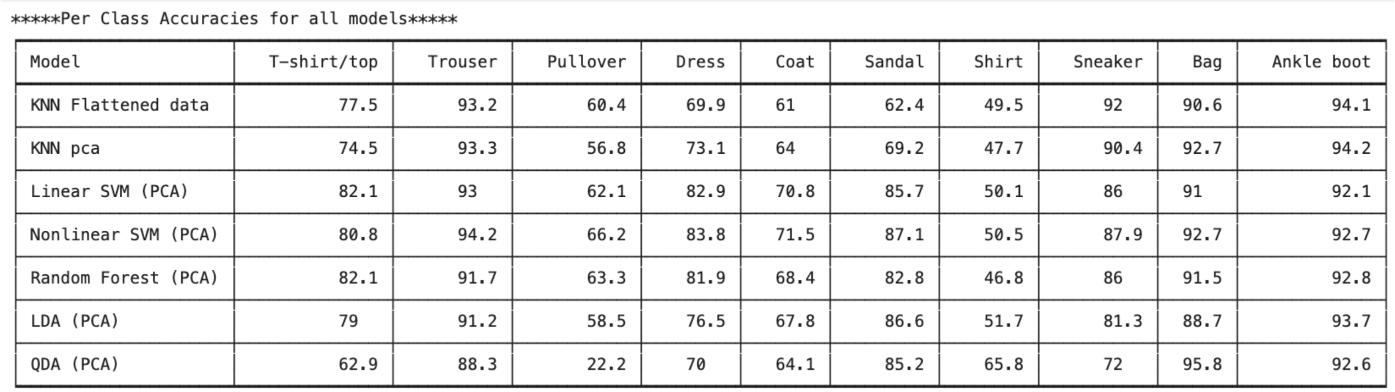
Refer to Figures 6 and 7. Overall, Classes such as Ankle Boot, Bag, and Trouser had the highest classification accuracies in the range of 88.3% to 95.8%. However, classes such as Shirts, Coats, and Pullovers having the most misclassifications within the approximate range of 50% to 60% with the exception of QDA classifying Pullovers at an accuracy of 22.2% which might mean the covariance matrix for Pullovers is not optimal/corrupted. The model I would choose is the Nonlinear SVM model which uses the RBF kernel. The reason is because it works well at classifying data in higher dimensions like the PCA data. It also had the highest total accuracy at 80.74% and performed the best in 5 out of the 10 classes. The total accuracy for each model follows the following order, Nonlinear SVM, Linear SVM, Random Forest, LDA, KNN (pca), KNN (flattened data), and QDA. Ranking by class accuracies: T-Shirts: Linear SVM | Trouser, Pullover, Dress, Coat, and Sandal: Nonlinear SVM | Shirts: QDA | Sneaker: KNN (pca data) | Bag: QDA | Ankle Boot: KNN (pca data). Despite QDA having the lowest total accuracy, it performed the best on identifying shirts (which other models struggled a lot with) and Bags. This could indicate that Shirts and Bags are more separable by a quadratic decision boundary.

PART 7: DNN on full image data

A Deep Neural Network (DNN) is a neural network consisting of input layers, hidden layers, and output layers. It excels in learning complex relationships and patterns from data. The Last method is to use a Convolutional Neural Network (CNN) which is a type of DNN. Two models were trained both using the full non-flattened training data but the first approach only trained on the first 1000 samples whereas the second approach trained on all 60000 samples. For the first approach, a batch size of 100 was used. The batch size is the number of samples that will processed through the network before updating the weights at each iteration. The benefit of using a batch size is that the model can train faster and it uses less memory. However, the smaller the batch size the less accurate the model could become. The CNN network was trained using Adam optimizer for 50 epochs. The number of epochs is how many many passes through the entire training dataset during a cycle of training. Too many epochs can lead to overfitting and can be slow to train on. Cross-entropy loss was used to measure the performance of the model. Refer to Fig 8. The DNN trained on 1000 samples performed much better than the non-DL methods. It performed better than the other models in classifying T-shirts, Trousers, Dress, Coats, Sandals, Sneakers, Bags, and Ankle Boots. The most improvement seems to be in classifying Sandals however it also struggles with classifying Shirts. Regarding the CNN trained on all samples, the main change was to train on all 60000 samples with 10 epochs rather than 50 for speed. (Refer to Fig 9) It performed even better than the previous CNN model which makes sense because it is trained on more samples. For all classes it had higher accuracies. Classes such as Sandals and Bags had the highest level of accuracies whereas this model also struggled with Shirts. (Refer to Fig 10), the increase in sample sizes led to a total accuracy of 91.05% rather than 83.03% as in the first CNN model. What we can conclude from this is that increasing the sample sizes leads to considerable increase in performance. But the downside with this is that the training process is also much slower which is why the number of epochs need to be reduced to improve speed.

PART 8: Conclusions

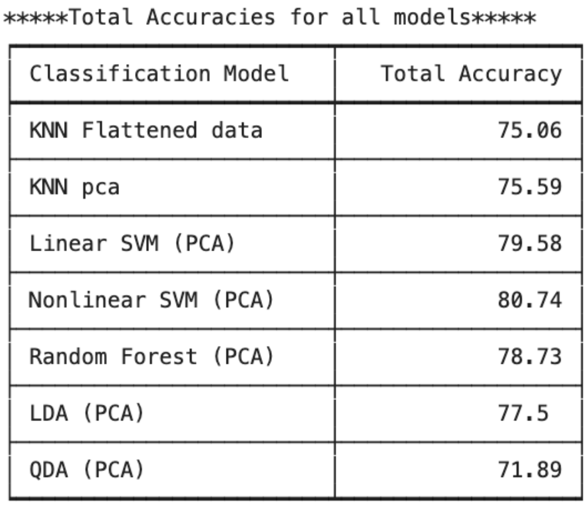
Overall regarding the non-DL methods, the best models were the Nonlinear SVM model which used the RBF kernel and the Linear SVM model. The worst model being QDA which did perform better in few categories such as Shirts and Bags indicating that these classes might be best separated with a quadratic decision boundary. The total accuracies for non-DL models hovered between the 70%s and low 80%s. The DNN models performed much better than the non-DL methods with total accuracies in the high 80%s and low 90%s. This means that CNNs perform much better when it comes to image classification tasks. The second CNN model which was trained on the entire training set of 60000 samples performed the best overall which shows the importance of having large amounts of training samples. Throughout all approaches the classes that seemed to be the easiest to classify are Ankle Boots, Bags, Sneakers, and Trousers. Indicating that these classes are more separable from the rest. Classes that all models struggled the most on are Shirts and Pullovers indicating that they are less separable from other classes.



*****Per Class Accuracies for all models*****

Model	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
KNN Flattened data	77.5	93.2	60.4	69.9	61	62.4	49.5	92	90.6	94.1
KNN pca	74.5	93.3	56.8	73.1	64	69.2	47.7	90.4	92.7	94.2
Linear SVM (PCA)	82.1	93	62.1	82.9	70.8	85.7	50.1	86	91	92.1
Nonlinear SVM (PCA)	80.8	94.2	66.2	83.8	71.5	87.1	50.5	87.9	92.7	92.7
Random Forest (PCA)	82.1	91.7	63.3	81.9	68.4	82.8	46.8	86	91.5	92.8
LDA (PCA)	79	91.2	58.5	76.5	67.8	86.6	51.7	81.3	88.7	93.7
QDA (PCA)	62.9	88.3	22.2	70	64.1	85.2	65.8	72	95.8	92.6

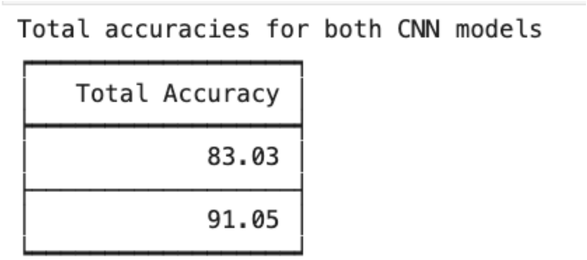
Fig. 6. Per-class accuracies for all models



*****Total Accuracies for all models*****

Classification Model	Total Accuracy
KNN Flattened data	75.06
KNN pca	75.59
Linear SVM (PCA)	79.58
Nonlinear SVM (PCA)	80.74
Random Forest (PCA)	78.73
LDA (PCA)	77.5
QDA (PCA)	71.89

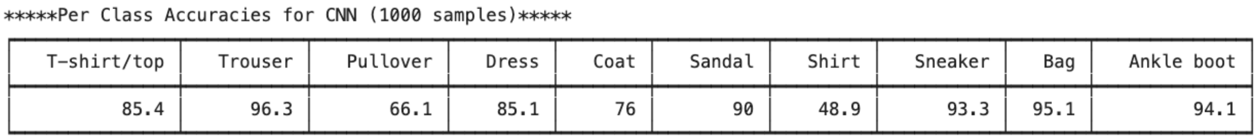
Fig. 7. Total Accuracies for all models



Total accuracies for both CNN models

Total Accuracy
83.03
91.05

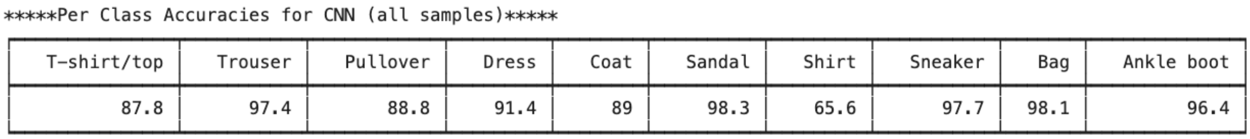
Fig. 10. CNN total accuracies for 1000 samples and all samples respectively



*****Per Class Accuracies for CNN (1000 samples)*****

T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
85.4	96.3	66.1	85.1	76	90	48.9	93.3	95.1	94.1

Fig. 8. Per Class Accuracy CNN (1000 samples)



*****Per Class Accuracies for CNN (all samples)*****

T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
87.8	97.4	88.8	91.4	89	98.3	65.6	97.7	98.1	96.4

Fig. 9. Per Class Accuracy CNN (all samples)