Parker Williamson
2/14/2018
Springboard Data Science Career Track

Capstone Project 1 Final Report – Clothing Categorization

Problem

Organizing dishes and clothes are some of the most time intensive tasks that are still done manually in the home. I hope to take a step in the direction of alleviating those tasks by creating an automated system for clothes categorization. New items of clothing must also be manually categorized when being added to online stores and clothing categorization can save time for that as well.

Client

Washer and dryer makers and indirectly home owners can benefit from some of this dynamic classification. It has the potential to move towards saving millions of people hours per week. As a result a folding machine should be made if the categorization is right a high percentage of the time. Another segment of customers which will be reached first is online clothes sales.

Dataset

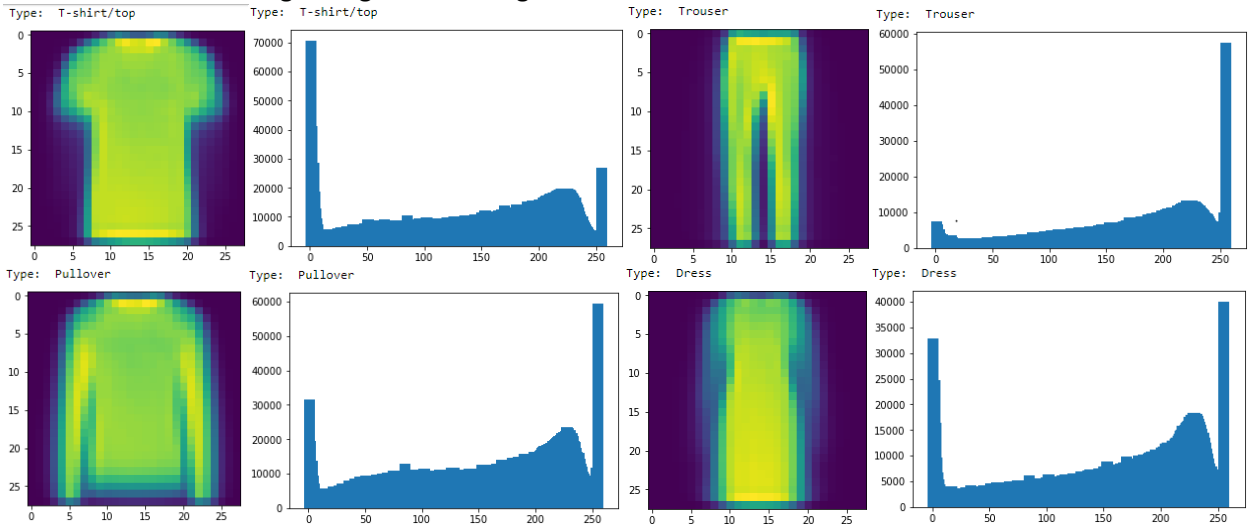
Fashion.mnist (https://www.kaggle.com/zalando-research/fashionmnist/data) will be the main data I use to explore image processing further and train on.

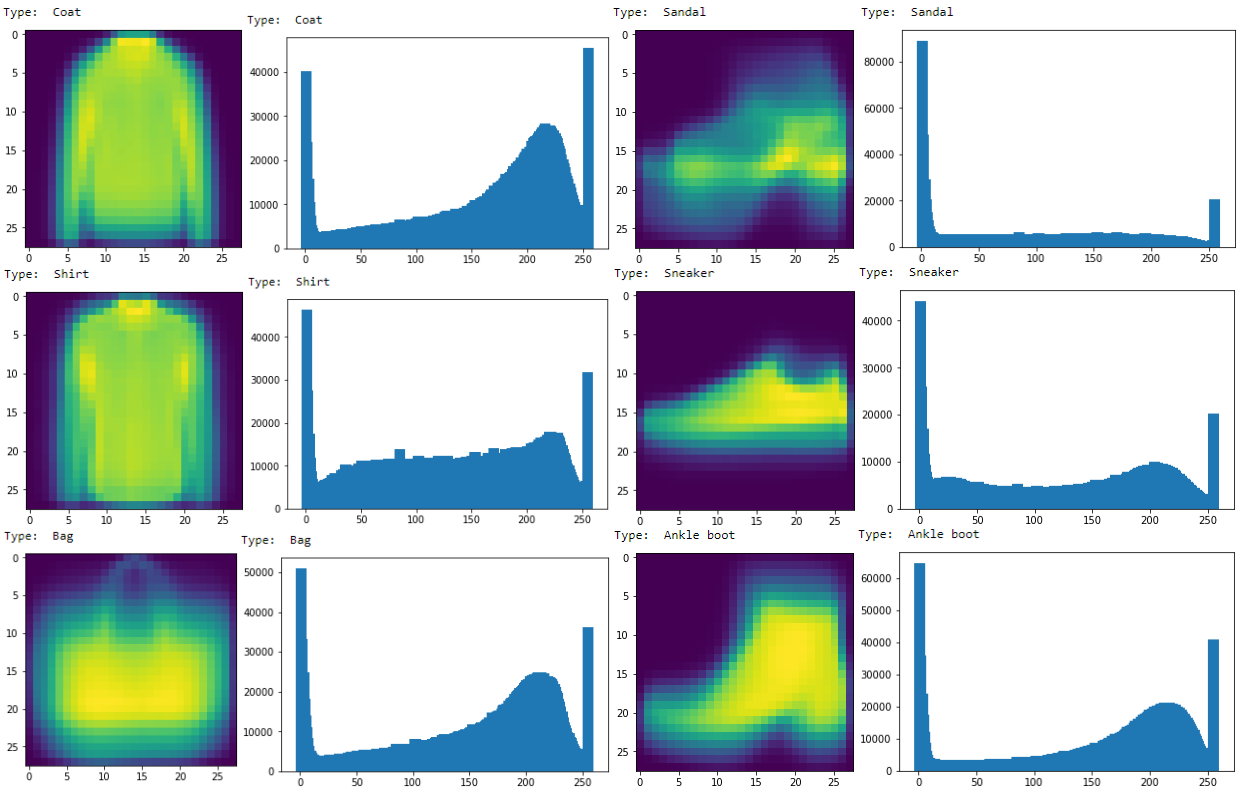
Alternative datasets

I also test against two cleaned photos I took myself, although not statistically significant it gave a general idea of the model robustness.

Analysis

The clothing MNIST dataset has 10 different clothing types: T-Shirt, Trousers, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. To get a general understanding of each type I took the average of each type by pixel and made one average image. I also took histograms of all the images for each class. These average images and histograms are shown below.





Printing out averages images and their histograms shows how similar the T-shirt/top category is to shirt and how similar pullovers are to coats. All the other categories have pretty distinct features compared to each other. Coats tend to be a little larger/brighter than pullovers and there's a little bit of a button line. Shirts sometimes have sleeves, where T-shirts do not and also have a little bit of a button line.

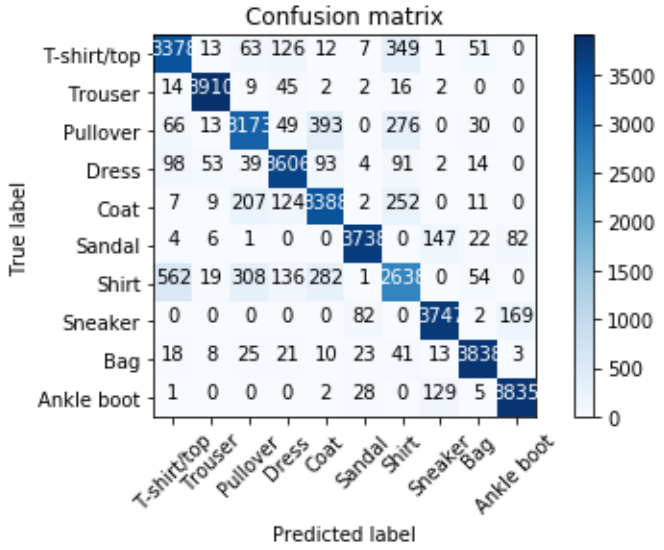
Preprocessing was the next part of the categorization that I explored. The technique was different depending on if you were running CNN or one of the other categorization techniques. For Multi-layer Perceptron (MLP), SVC and Logistic regression it was more beneficial to normalize and standardize to a 0-1 range. When using CNN standardizing and normalizing to a 0-255 range improves the results, the images were initially normalized to a 0-255 range. For CNN standardization improves results by ~1%.

In addition to analyzing the models, the image categories were explored as well. In order to compare how correlated the categories of clothing are I compared the means of the data. The most similar means would be next to each other, so I ordered the means from smallest to largest and then did a z-test to determine if the different categories have the same means. A z-test assumes normal distribution and whether two different samples have the same mean. For a z-test the sample sizes should be larger than 30 (here it is 6000). Z-tests use the mean and standard deviation. The ordered means of the photos have no statistically significant correlation, because all of the p-scores of the closest means are below .05. A p-score shows the results of the null hypothesis test determining where or not the means are correlated. It is not a probability, but you can see how far away the results are from being above .05.

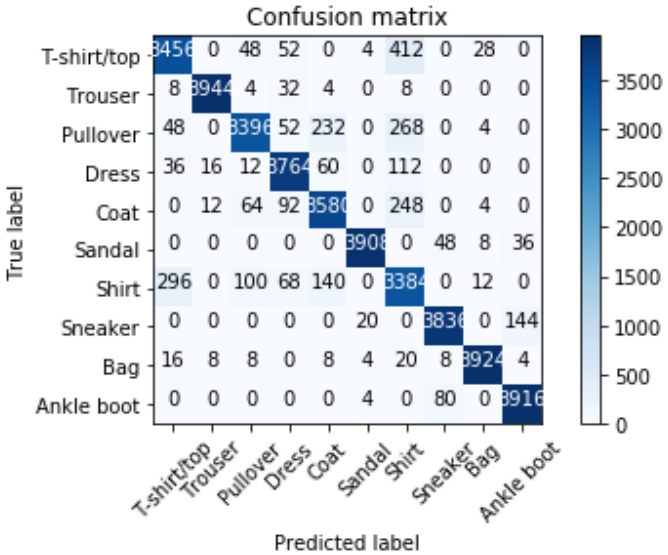
```
p-score 0-1(Z): 0.0
p-score 1-2(Z): 0.0
p-score 2-3(Z): 0.0
p-score 3-4(Z): 0.0
p-score 4-5(Z): 0.0
p-score 5-6(Z): 0.0001
p-score 6-7(Z): 0.0
p-score 7-8(Z): 0.0
p-score 8-9(Z): 0.0
```

The predicted (using standardized MLP) versus actual results were compared and the results are shown below the diagonal large values on the diagonal clearly show that large majority were correctly classified. The rest of the values (not correct values) were analyzed using percentile to look for 95% outliers. The first confusion matrix is of all 4 of the main different model results summed and the second one is of just 4 times the matrix of the CNN model. Multiplying by 4 allows for direct comparison between the sum and CNN confusion matrixes.

Combined

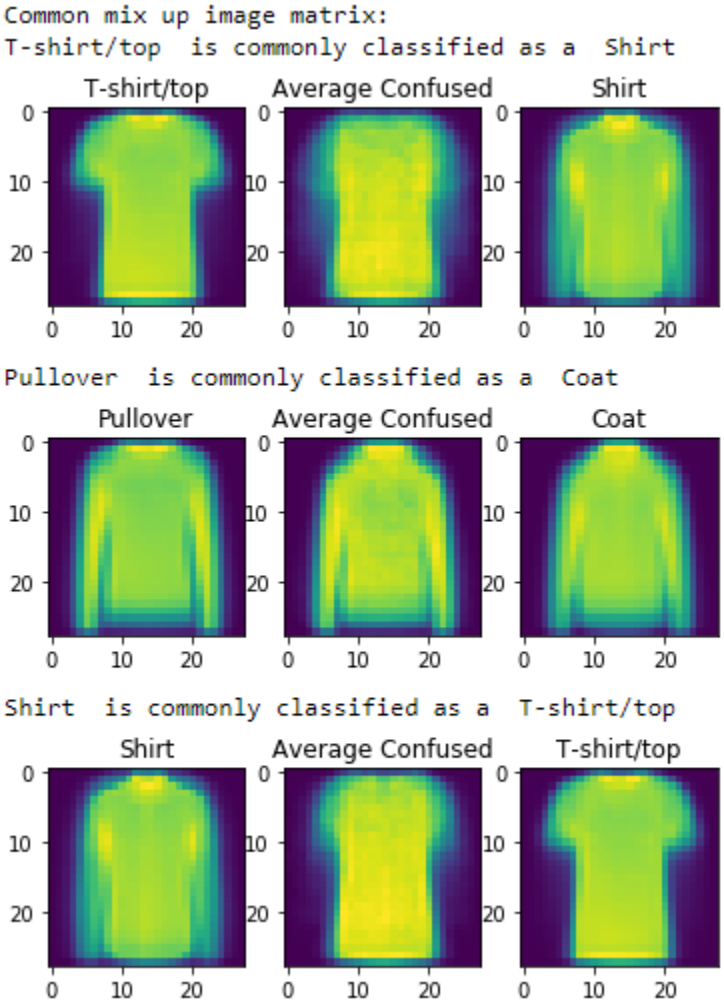


CNN*4



As you can see the CNN model has lower mistakes and didn't hit some miss classifications of the summed confusion matrix. Only is the points with one mistake per classifier dos the CNN model occasionally have 1 or 2 more of that type of misclassification.

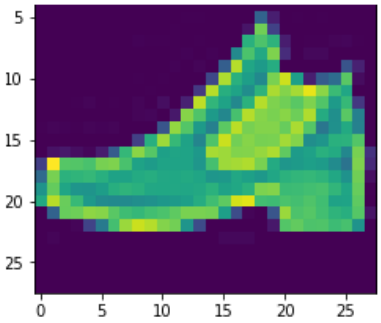
There were three values that stood out, and they were: T-shirts/tops were commonly classified as shirts, Pullovers were commonly classified as coats and shirts were commonly classified as T-shirts/tops. Those are some of the categories that would be hard for a human to classify as well as you can see when looking at the average image of those types. The bias towards classifying pullovers as coats was an interesting feature, because it would make sense for the confusion to go both ways but pullovers may have more than on distinct category one of which overlaps with coat.



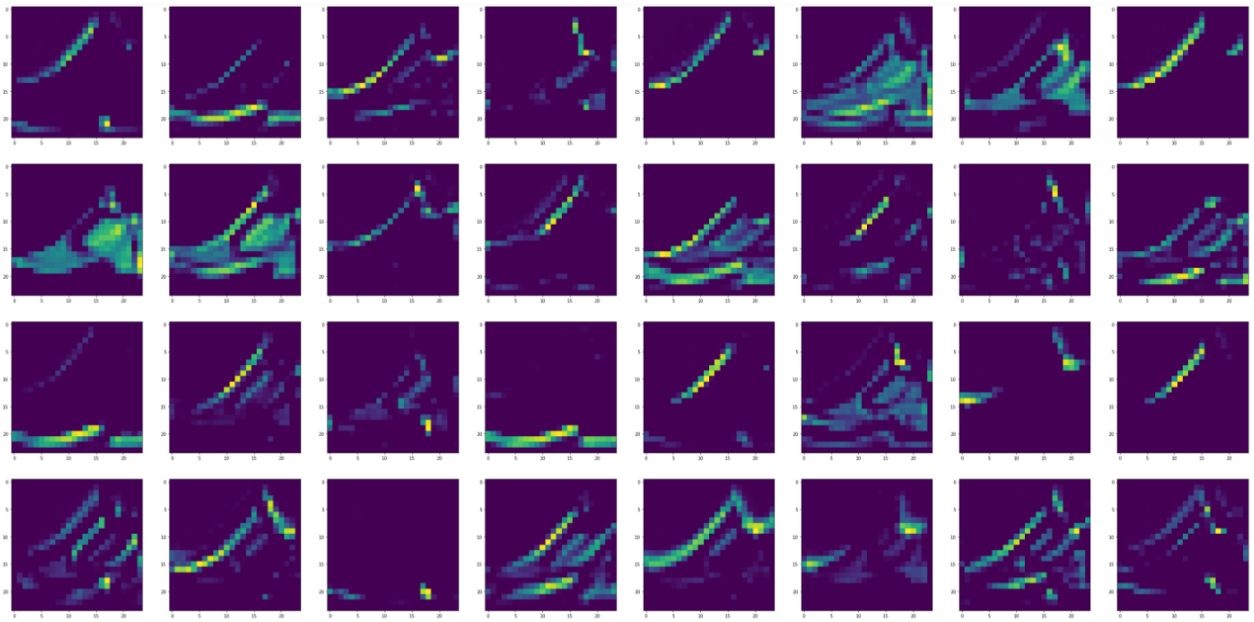
The commonly mixed up image matrix shows the average of the images that were mixed between groups of the outliers. The T-shirts/tops that are classified as shirts seem to have less prominent sleeves and a more patterned texture. The pullovers classified as coats seem very similar so the classifier seems to classify it as a coat if it is not sure between the pullover and coat categories. The Coat group seems ever so slightly brighter and that may be one of the main distinctions it sees. The average confused shirt does appear to me to be more like a T-shirt so it may be hard for ever humans to classify them, though example of all those could be looked at individually to see if the original classification break between shirt and T-shirt is consistent.

Finally there are visualizations of the convolutional neural network (CNN) intermediate layers. The first layer breaks the image into many different pieces and the second layer abstracts them to focus more on the general form. This is of the first CNN architecture and layout. Results may be improved by changing it, but these results are representative of the filtration and abstraction that the neural network does on every image.

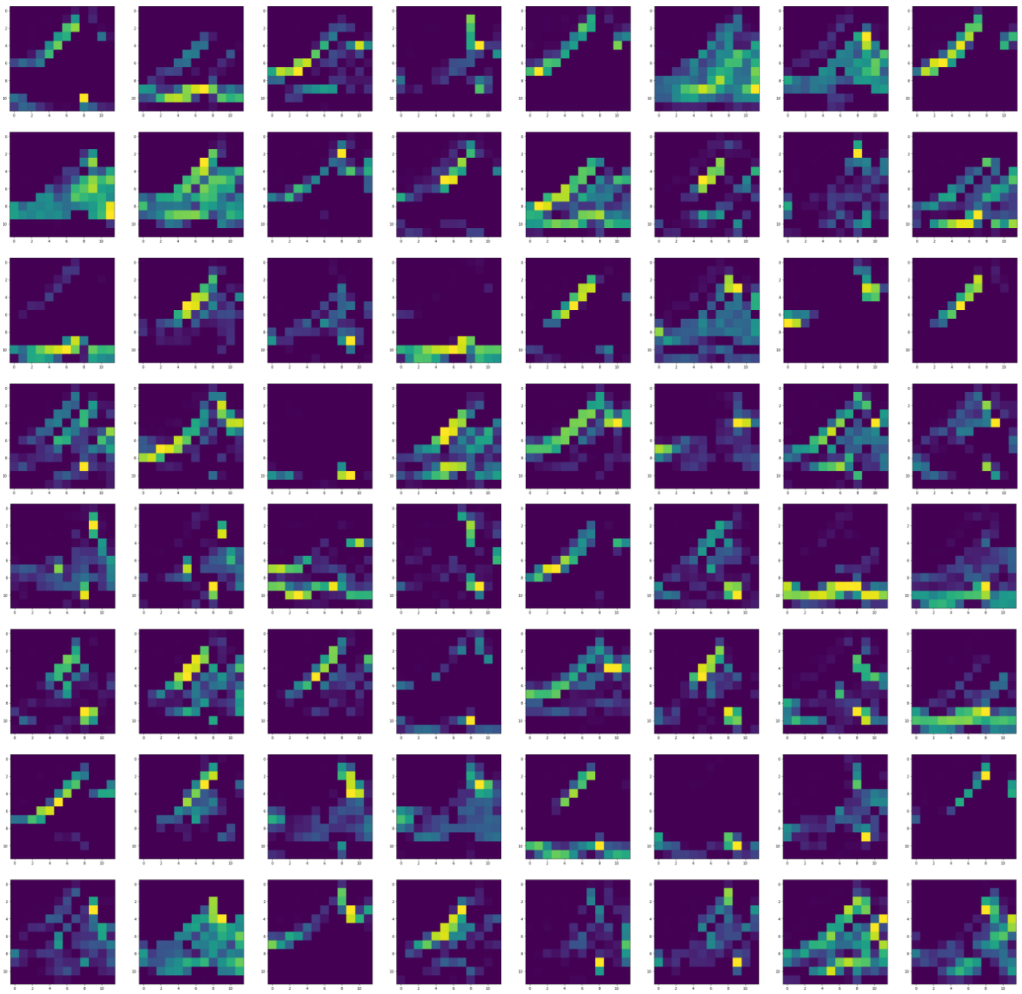
Original image



PHASE 1



PHASE 2



Further tweaking of the neural network is could improve results, but there are many ways that images can be separated and identified through the visual properties show in this document.

The results of Multi-layer Perceptron (MLP), SVC and Logistic regression were also compared with each other across different classification methods. MLP was the most efficient with Normalization as the image preprocessing.

MLP (size of hidden layers 784-100-100):

- With no preprocessing – 87.65% accuracy
- Normalized(0-1) – 90.37% accuracy
- Standardized(0-1) – 90.09% accuracy

SVC:

- With no preprocessing – 70.67% accuracy
- Normalized(0-1) – 85.57% accuracy
- Standardized(0-1) – 81.87%

Logistic Regression:

- With no preprocessing – supposed to be normalized
- Normalized(0-1) – 84.45% accuracy
- Standardized(0-1) – 85.19% accuracy

CNN:

- Normalized(0-255) – 92.25% accuracy

- Standardized – **92.32% accuracy**

Ensemble the above models (Random Forest with 20 trees):

- 90.56% accuracy

Ensemble the above models (Random Forest with 20 trees):

- 91.25% accuracy

Use VGG19 as a feature extractor before a neural net:

- 85.71% accuracy

From my tests I learned that Normalization and standardization are roughly equivalent, the initial data is normalized to a range of 0-255. CNN preforms best with an input range of 0-255 and the rest with 0-1. Surprisingly the standardization of the logistic regression model is more effective despite sklearn stating that normalization it important for it. MLP is the most effective of the non-CNN classifier.

The precision, recall and F1 score of each function were calculated as well:

MLP (size of hidden layers 784-100-100):

	precision	recall	f1-score	support
T-shirt/top	0.81	0.89	0.84	1000
Trouser	0.98	0.99	0.98	1000
Pullover	0.85	0.78	0.82	1000
Dress	0.91	0.91	0.91	1000
Coat	0.82	0.89	0.85	1000
Sandal	0.98	0.94	0.96	1000
Shirt	0.78	0.69	0.73	1000
Sneaker	0.94	0.95	0.94	1000
Bag	0.98	0.98	0.98	1000
Ankle boot	0.94	0.96	0.95	1000
avg / total	0.90	0.90	0.90	10000

SVC:

	precision	recall	f1-score	support
T-shirt/top	0.78	0.81	0.80	1000
Trouser	0.96	0.97	0.96	1000
Pullover	0.77	0.77	0.77	1000
Dress	0.84	0.88	0.86	1000
Coat	0.77	0.80	0.79	1000
Sandal	0.93	0.90	0.91	1000
Shirt	0.66	0.55	0.60	1000
Sneaker	0.90	0.92	0.91	1000
Bag	0.92	0.93	0.92	1000
Ankle boot	0.93	0.94	0.93	1000
avg / total	0.84	0.85	0.85	10000

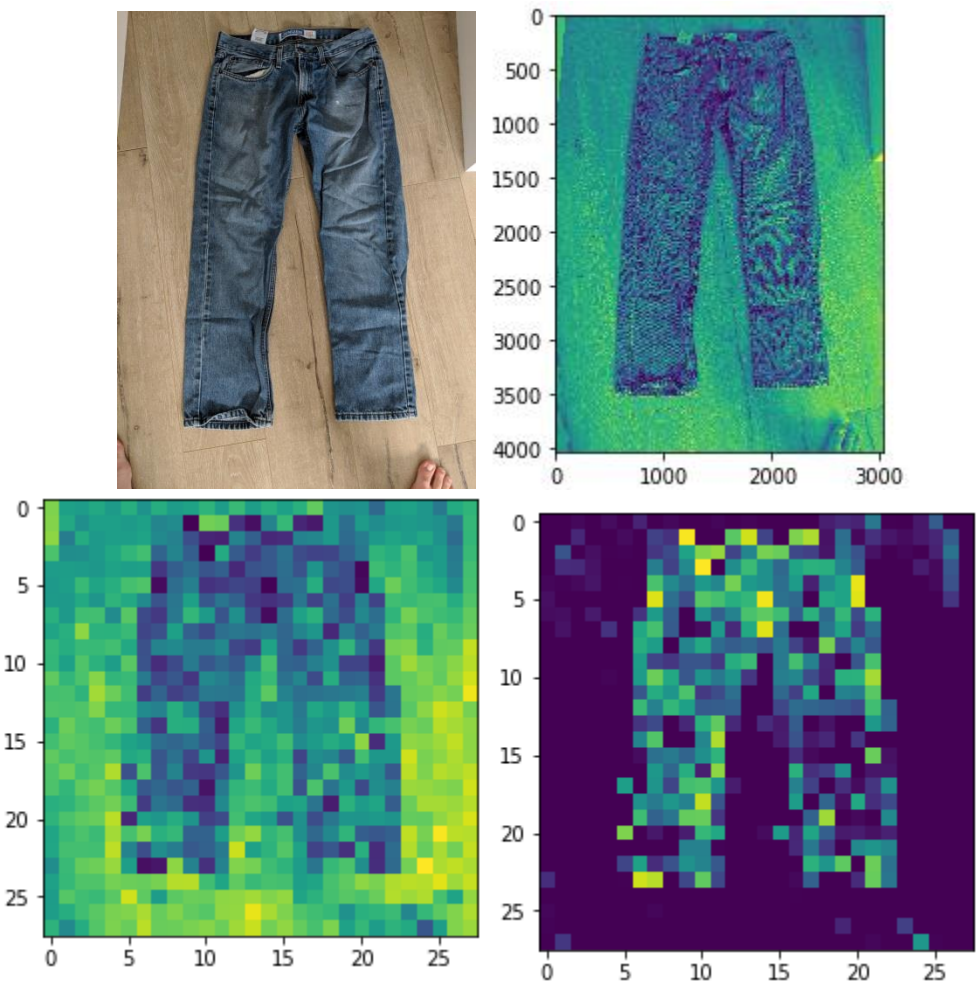
Logistic Regression:

	precision	recall	f1-score	support
T-shirt/top	0.78	0.81	0.80	1000
Trouser	0.95	0.97	0.96	1000
Pullover	0.78	0.77	0.78	1000
Dress	0.84	0.88	0.86	1000
Coat	0.77	0.80	0.78	1000
Sandal	0.95	0.92	0.93	1000
Shirt	0.66	0.56	0.60	1000
Sneaker	0.90	0.92	0.91	1000
Bag	0.93	0.94	0.94	1000
Ankle boot	0.93	0.95	0.94	1000
avg / total	0.85	0.85	0.85	10000

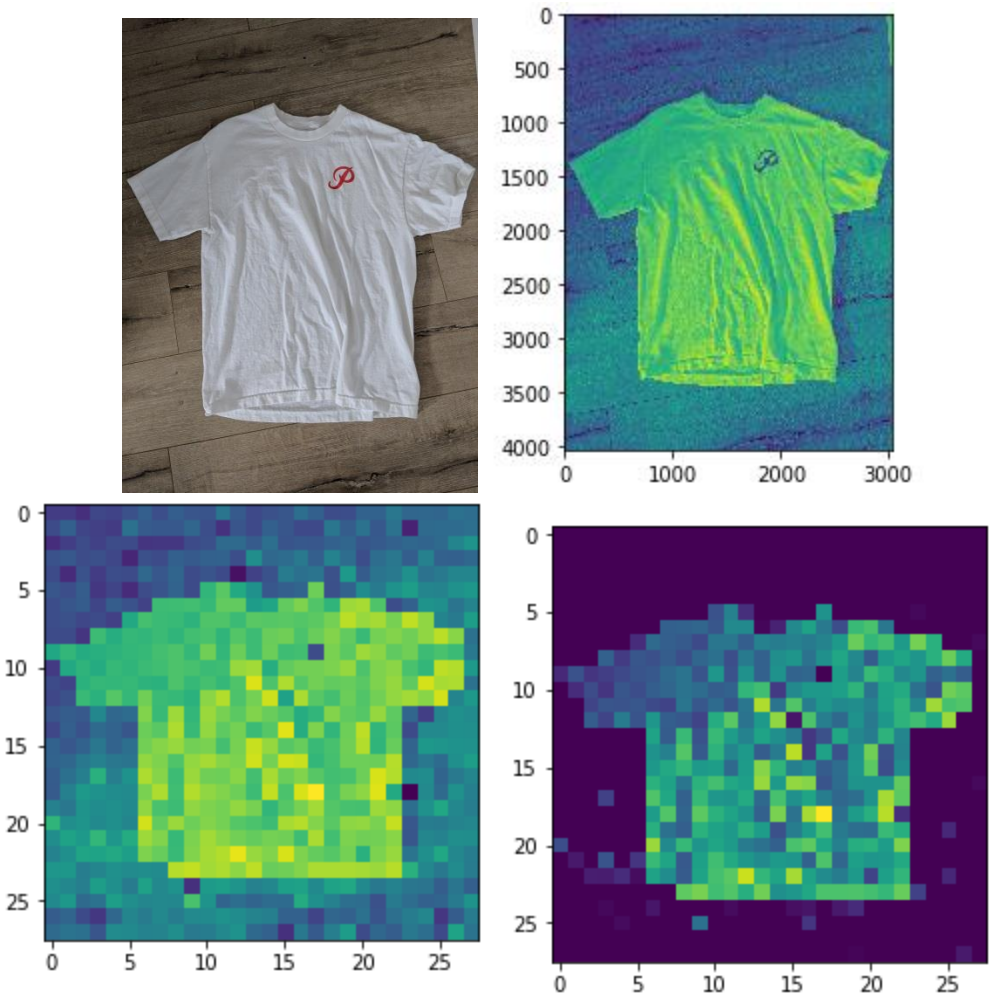
CNN:

	precision	recall	f1-score	support
T-shirt/top	0.90	0.86	0.88	1000
Trouser	0.99	0.99	0.99	1000
Pullover	0.94	0.85	0.89	1000
Dress	0.93	0.94	0.93	1000
Coat	0.89	0.90	0.89	1000
Sandal	0.99	0.98	0.98	1000
Shirt	0.76	0.85	0.80	1000
Sneaker	0.97	0.96	0.96	1000
Bag	0.99	0.98	0.98	1000
Ankle boot	0.96	0.98	0.97	1000
avg / total	0.93	0.93	0.93	10000

I further put the classifiers to the test by taking my own, somewhat noisier, images and running them through the classifiers with a grayscale conversion and making them look as similar to the training images as I could by inverting and normalizing them. This was just a rough test because I did it with a tiny insignificant sample size, but it gives some insights on the models general robustness. First I did a pair of genes I had to convert the image to grayscale, resize it to 28x28, invert the image, filter out everything less than the mean and normalize the image to 0-1. In general it looks like one of the images, though it is a little spottier. MLP classified it as a coat, SVM and LogReg both correctly said trousers and CNN said shirt. The T-shirt I preprocessed similarly, but I did not invert the image, since the T-shirt was white and therefor brighter than the background. MLP classified it as a bag, whereas SVM, LogReg and CNN classified it as a shirt. So none of the classifiers said T-shirt, although the distinction between shirt and T-shirt is small I expected one to be correct at least. Those results do tend to indicate that although the SVM and LogReg are not as good at classifying the training and test data they may be better for data not cleaned in the exact same way. The MLP especially may be overfitting to the sample dataset. The size of this test is not representative and could be done with more data but look out for MLP overfitting. It also shows how important it is to train on data filtered and taken in via a consistent manner. The trouser and T-shirt images are shown in the next two pages, original, through the image fed to the classifiers.



(Top left) Original image, (top right) gray scale image, (bottom left) gray resized image, (bottom right) final inverted and filtered image fed to classifiers



(Top left) Original image, (top right) gray scale image, (bottom left) gray resized image, (bottom right) final filtered image fed to classifiers