Report on exercise 3, group 1

Image processing: traditional vs. deep learning methods

Introduction

The purpose of this document is to provide a report of group 1 for exercise 3 of the course “Machine Learning” in the WS 2021/2022. The exercise chosen was 3.2, which is about comparing traditional methods to a deep learning approach in classifying images. For the traditional methods, two kinds of feature extraction methods are chosen, and the extracted features where used to train multiple ML algorithms and parameter settings. For the deep learning approach, two different architectures are used in a CNN model.

Datasets

The datasets studied are the “Fashion MNIST” data provided by Zalando, and the “Faces Labelled in the Wild” data provided by sklearn.

“Fashion MNIST” is a relatively new dataset, which is supposed to offer more complex “MNIST” data, as the original handwritten-digits MNIST dataset has been used extensively already. In contrast to the handwritten-digits data, the Zalando MNIST dataset is supposed to offer more complexity and difficulty in classifying images. They are provided as grey images in the size of 28x28 pixels, with a total of 60000 training and 10000 test images. The data contains 9 classes in total, among which are clothing types such as pants, skirts, shoes, or shirts.

The “Faces Labelled in the Wild” data comprises 3024 images of people, which can be used to recognize/classify people individually. The images are colorful, in contrast to the Zalando MNIST data, and only people with at least 20 pictures are kept in the dataset. Each person in the data is treated as a single class. For the traditional methods, the original images are used, which are sized as 154x154 pixels and three color channels (red, green, and blue). As the dimensions quickly filled up the GPU memory for the deep learning methods, the images were scaled down to 48x48 pixels.

Methodology

The experiment design is split into two parts: traditional and deep learning methods.

The traditional approach uses two feature extraction methods - color histograms and “scale-invariant feature transform” (SIFT) + “Bag of Visual Words” (BoVW) – to extract information from images. The obtained features are then used by multiple ML algorithms - namely Multi-Layer Perceptron, Random Forrest, Decision Tree, Support Vector Machine, Bayesian Network, and Quadratic Discriminant Analysis – to create classifiers. The following parameter settings were used per classifier:

- Multi Layer Perceptron:
    - Activation: relu or tanh
    - Learning: constant/adaptive
    - Layers of the neural network: 5-2, 50-50-50, 50-100-50, and 100

- Random Forrest:
  - Maximum Depth: 5, 10, and 20
  - Nr. Of Estimators: 10, 50, and 100
- Support Vector Machine:
  - Kernel: linear (gamma = “scale”) or rbf (gamma = 2)
  - C: 0.025, 0.5, 1
- Decision tree:
  - Maximum Depth: 3, 15, 35, and 80
- Quadratic discriminant analysis: single run with default settings
- Naive Bayes: single run with default settings.

The classifiers are compared among the feature extraction methods of the traditional approach, and the experimental results of the deep learning approach.

The deep learning approach uses convolutional neural networks (CNNs), which are trained with two different architectures, namely MiniVGGNet and MiniGoogLeNet. The CNN architectures are available at https://github.com/agoila/lisa-faster-R-CNN/tree/master/pyimagesearch/nn/conv, and they were chosen due to the ligher time and memory resources required. The first architecture is the MiniVGGNet and is a lightweight version of the VGGNet architecture. This architecture follows given requirements found to work especially well for images. The second architecture is the MiniGoogLeNet and is based on the GoogLeNet, which is a more complex CNN. The MiniGoogLeNet has a reduced set of parameters compared to the GoogLeNet architecture.

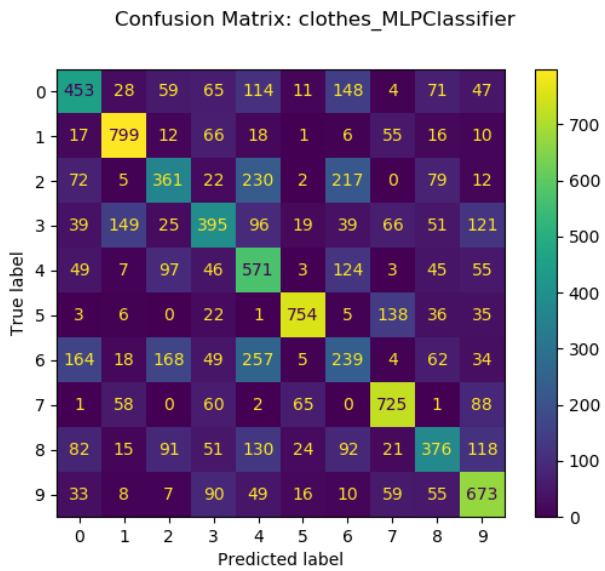
Results – traditional methods

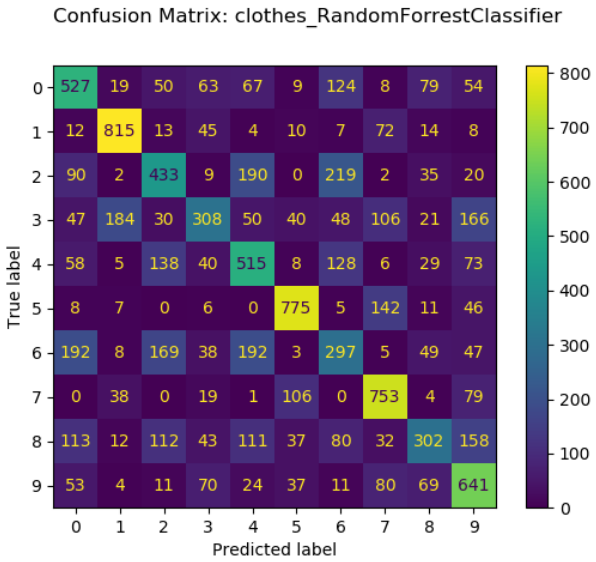
For the traditional approach, we analyzed the performance of multiple ML algorithms on two datasets. The “Fashion MNIST” solely comprises gray images, whereas the “Labelled Faces in The Wild” dataset is colorful and therefore contains three color channels per image.

Overall, we observed that the traditional methods performed considerably better for the “Fashion MNIST” dataset, where almost all models and parameter settings result in very good classification metrics. Nevertheless, neither the color histograms nor the SIFT + BoVW feature extraction approach were able to properly recognize people in the “Labelled Faces in The Wild” dataset. Instead, we observed that most models and settings resulted in classifying almost all test images as the majority class (George W. Bush).

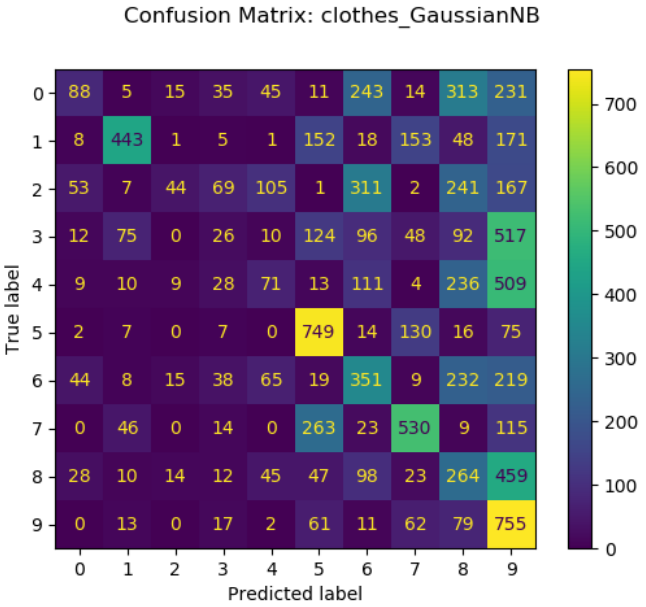
MNIST data – detailed comparison

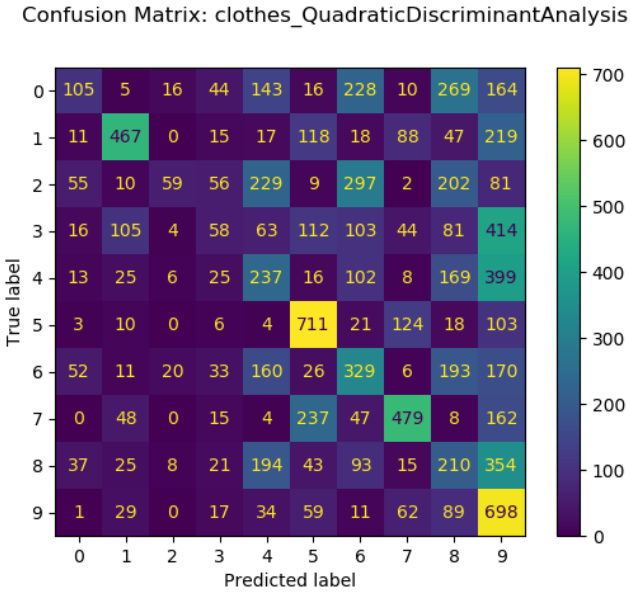
For the “Fashion MNIST” data, the Multi-Layer-Perceptron (MLP) with three layers (50-100-50) , the RELU activation, and adaptive learning rate showed the best results, followed by the Random Forrest model (Maximum Depth: 20, Nr. Of Estimators: 100):





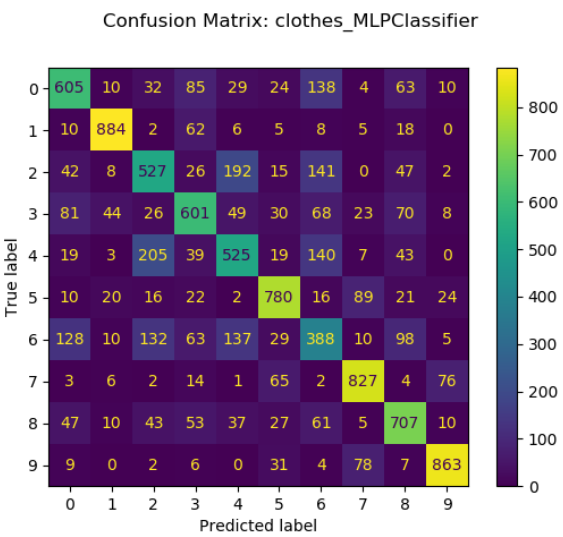
Other models such as the SVM or the DecisionTree models have comparatively good results. The Bayesian model and the QDA show skewed distributions in classifications, rendering them to be worse classifiers for the MNIST data:

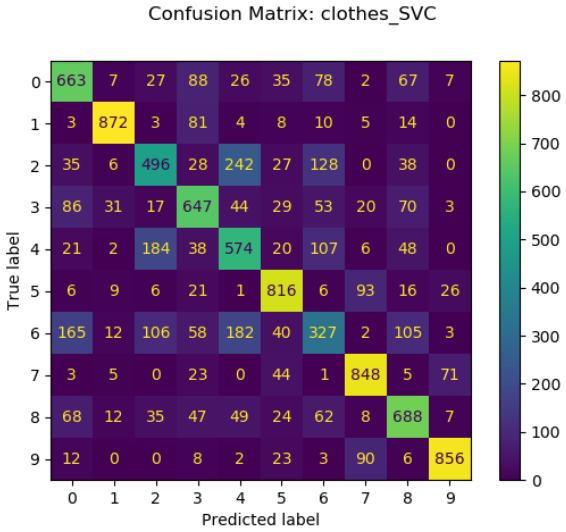




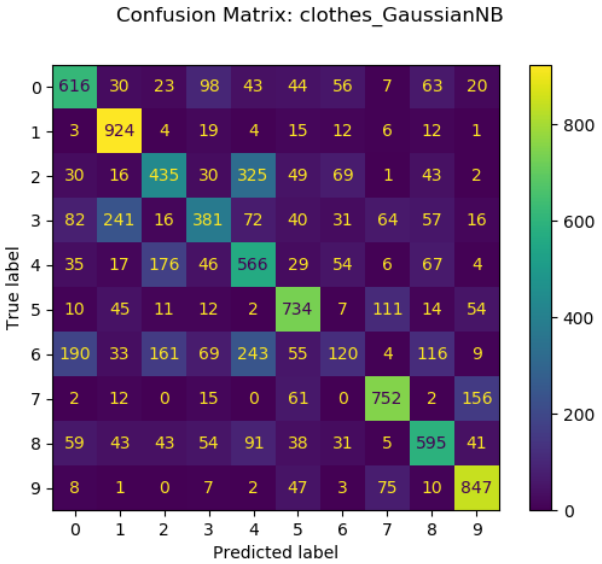
The SIFT approach generally resulted in a more accurate classification of the MNIST data, where the SVM proved to show the best classification results, outperforming the MLP in most classes and the Random Forrest in almost all

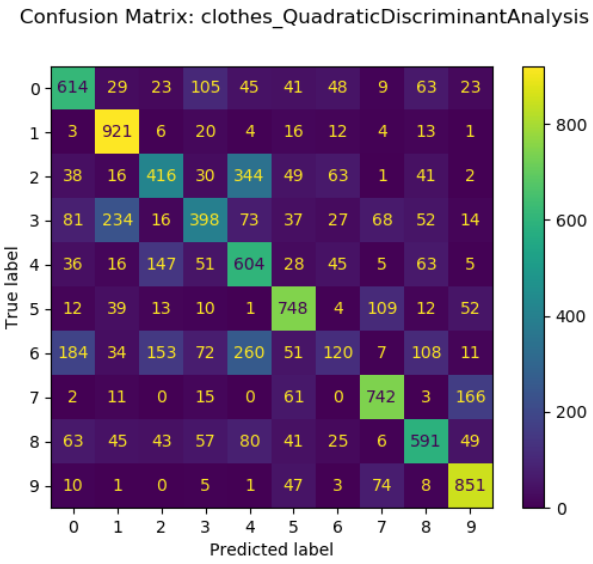
classes. Below, the MLP (with the same settings as described for the Color Histogram approach) and the SVM ('rbf' kernel, C=1) is shown:





Interestingly, extracting the features using the SIFT method resulted in very competetitive results for the Naïve Bayes and the QDA method, which both had very unbalanced classification results in the color histogram approach. With SIFT, both models outperform the best-performing models in several classes, such as the T-shirt (0) and Trouser (1) class for the Naïve Bayes, or the Trouser (1) class for the QDA model.





Regarding the misclassifications, it is important to note that in both approaches (color histograms and SIFT), the Shirt class (6) results in the largest number of classifications, being misclassified as either T-shirts (0), Pullover (2), or Coat (4). In contrast, the Bag class (8) resulted in many misclassifications for the color histogram approach, but showed considerable improvement in prediction rates

for the SIFT method. In terms of prediction accuracy, we observed more than 50% improvement in precision, recall, and f1-score of the MLP for classes 6 and 8:

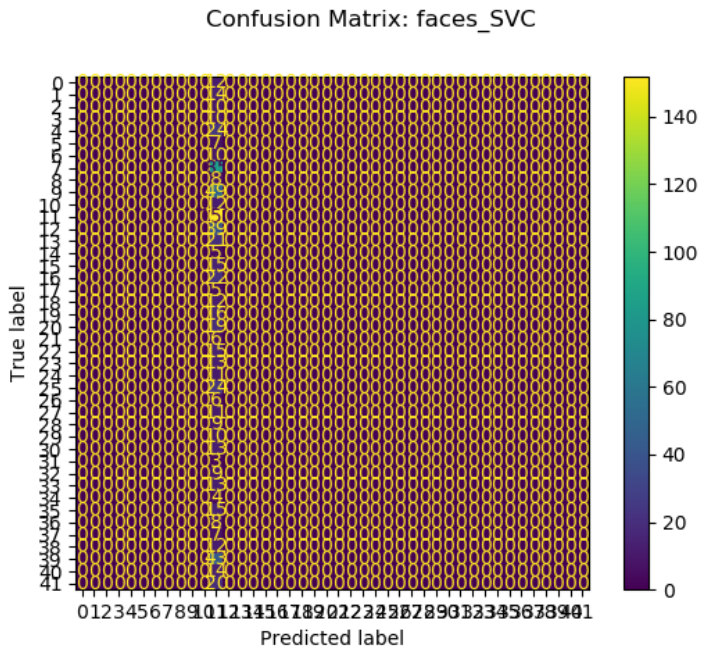
Model	Method	Class	Precision	Recall	F1-Score
MLP	Color Histogram	6	0.27	0.24	0.25
MLP	SIFT	6	0.41	0.39	0.41
MLP	Color Histogram	8	0.47	0.38	0.42
MLP	SIFT	8	0.63	0.69	0.66

We can observe similar behaviour for other classes and models, where the prediction performance generally improves with the SIFT method.

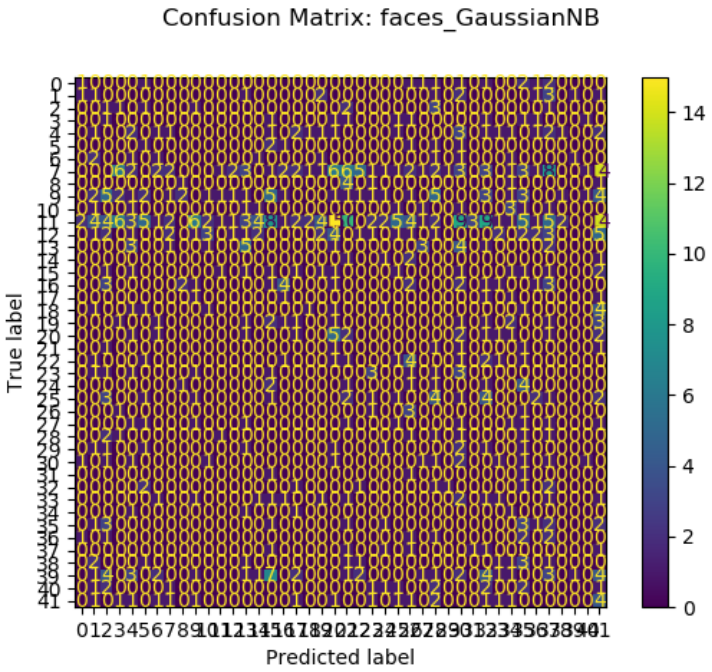
Regarding the parameter settings, our experiments confirm that an increase in tree depth improves the predictions for the tree-based methods studied, where best performance was obtained using a depth of 85 for DecisionTree and 100 for RandomForrest. For the SupportVectorClassifier, the linear kernel seems to overall offer better classification results compared to the “rbf” model, and a higher C parameter similarly improves performance.

Labelled Faces in the Wild – detailed comparison

For the traditional methods, the “Labelled Faces in the Wild” dataset presented a very difficult challenge in our experiments. Overall, neither the color histogram approach nor the SIFT method offered competetive prediction results. Instead, most models and parameter settings resulted in the majority class being classified in almost all cases. The dataset contains more than 500 image instances of George W. Bush alone, which heavily influenced the classification abilities of our models. As an example, the SVM with a “rbf” kernel and C=1 is shown below:



Due to the large number of classes (every person is considered to be an individual class), the axis labels are overlapping, but it should still be visible that the SVC in this example solely classifies images to be of class 11 (George W. Bush). All the other models, except the MLP using a single layer NN with 100 neurons and the Naïve Bayes show a pattern distinct from the rest. The single-layer MLP might not classify all images with a single class, but still has extremely low prediction accuracy, where only ~15 classes were accurately classified in 15-30% of cases. To our surprise, the Naïve Bayes with features extracted using the color histograms had the best results for this dataset:



Here, approximately 25 were at least partially recognized by the model, with prediction accuracies reaching up to 42% for some classes (16, 30, or 39), but George W. Bush still remains the majority class being classified.

In order to improve the results, we tried converting the colored images into gray scale (taking average values and combining 3 color channels into 1 gray channel), standardizing the pixel values, varying individual model parameters, and lastly testing a wide range of words for our “Bag Of Visual Words” clustering approach. We did notice that for the MLP, a single layer NN proved to be better than 2 or 3 layer settings, and also that cluster size in the Kmeans clustering of at least 10x the number of classes considerably improved the prediction performance, meaning that 100 words for the MNIST dataset (10 classes) and 500 words for the Labelled Faces dataset proved to be optimal. We varied the number of words from 3 to 2000. Additionally, we tested two python implementations (opencv and scikit-image) of the SIFT method and also tried varying the number of octaves, upsampling, and the thresholds for low contrast extrema (c_dog) and the edge extremas (c_edge) – with negligible improvement. Lastly, we tested two versions of clustering – MiniBatchKmeans and Kmeans – considering that the clustering approach might affect the majority class being

labelled in most cases – with only slight improvements using the Kmeans implementation. Unfortunately, we were not able to break the pattern of most classifiers assigning test images to the majority class only.

Conclusions – traditional methods

Both approaches – color histograms and SIFT + BoVW – presented very interesting use cases where each method showed superior results. For example, the color histogram has a very quick computation time and seems to perform better for the Labelled Faces dataset, the biggest difference being visible for the Naïve Bayes. In contrast, the SIFT approach might require longer preprocessing time from the start that depend on the number of words chosen for the clustering, it outperforms the color histogram method in precision and recall for almost all models and setting. Furthermore, we observed that the misclassification rates for the SIFT method are generally lower as it was mentioned before.

Overall, the color histogram approach is very easily implemented, whereas the SIFT method required a lot of time and thorough understanding to be properly incorporated in our analysis pipeline. One of the difficulties was to find appropriate thresholds for the SIFT feature extraction, as both implementation would throw warnings as soon as not enough information could be extracted from an image. Here, we implemented an exception-handling solution which would repeatedly loosen the thresholds of the SIFT extraction until enough information could be extracted.

We are happy to report that the traditional methods can be adjusted and tweaked in two ways, either through command line arguments or JSON configs. Moreover, our implementation allows to individually train models, meaning that it does not require to run all models at once. Lastly, we invested a lot of effort to modularize the code as good as possible, meaning that each processing step in our analysis pipeline is abstracted into individual functions and modules.

Deep methods

For the deep approach, we analyzed the performance of MiniVGGNet and MiniGoogLeNet on the two datasets. Each of the datasets was trained on augmented data and compared to training on non-augmented data.

It was found that it took a long time for the learning curves to converge. We ended up limiting the training epochs so they would end up training around 1000 seconds (=around 17 minutes) on our machine. Under this setup, the performance of was highly dependent on the learning rate. For each setup, we trained the model using varying learning rates, namely 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1} and compared the results.

The setup can be summarized with the following table:

Fashion MNIST				LFW Faces			
Without augmentation		With augmentation		Without augmentation		With augmentation	
MinVGGN	MinGoogN	MinVGGN	MinGoogN	MinVGGN	MinGoogN	MinVGGN	MinGoogN
LR: 10^{-4}	LR: 10^{-4}	LR: 10^{-4}	LR: 10^{-4}	LR: 10^{-4}	LR: 10^{-4}	LR: 10^{-4}	LR: 10^{-4}
LR: 10^{-3}	LR: 10^{-3}	LR: 10^{-3}	LR: 10^{-3}	LR: 10^{-3}	LR: 10^{-3}	LR: 10^{-3}	LR: 10^{-3}
LR: 10^{-2}	LR: 10^{-2}	LR: 10^{-2}	LR: 10^{-2}	LR: 10^{-2}	LR: 10^{-2}	LR: 10^{-2}	LR: 10^{-2}
LR: 10^{-1}	LR: 10^{-1}	LR: 10^{-1}	LR: 10^{-1}	LR: 10^{-1}	LR: 10^{-1}	LR: 10^{-1}	LR: 10^{-1}

Each white table cell represents a model that we trained that consists of the given learning rate and all its header rows. For example, the very top left white cell represents Fashion MNIST trained on MiniVGGNet without augmentation and a learning rate of 10^{-4}. MiniVGGNet and MiniGoogLeNet were abbreviated to fit their cells in the table.

We managed to achieve better results using the deep methods, compared to the traditional methods for both datasets.

Setup and results for Fashion MNIST data – deep methods

We use data augmentation to create more examples from our training data by varying image filters such as rotation, skew or orientation that works in favor of generalization. As evident from the picture below, the Fashion MNIST dataset has very low variability in that sense.

The notebook the assignment description referred to by Thomas Lidy featured transformations including rotations, width shifts, height shifts, zooming and flipping. For the Fashion MNIST dataset we went with a limited set of augmentations including rotations and horizontal flipping, with the rationale that for example a flipped shirt will still be a valid variation of a shirt that we may expect in the test data, and not all shoes being completely horizontal. More variations in that respect may represent valid additional training data.

The Fashion MNIST dataset consisted of 60000 training samples and 10000 test samples. MiniGoogLeNet is considered to be more sophisticated compared to MiniVGGNet, however our dataset managed to perform better training with the

MiniVGGNet. In roughly 1000 seconds, MiniVGGNet managed to train 50 epochs. MiniGoogLeNet managed to only train 5.

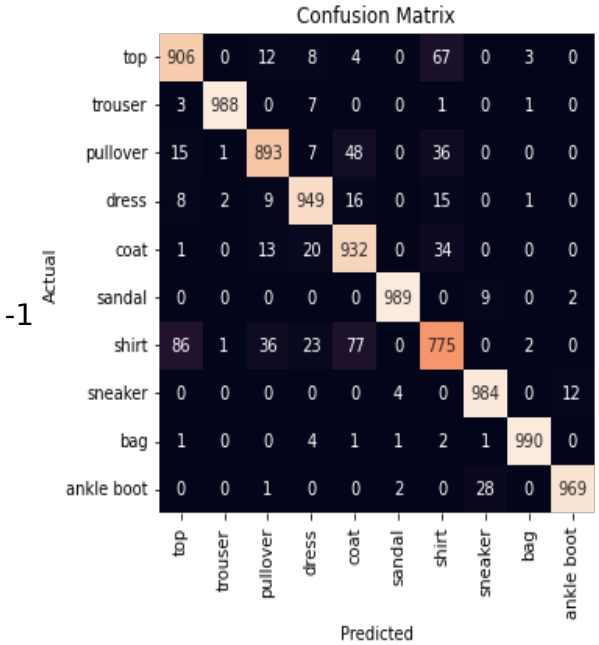
MiniVGGNet without augmentation and a learning rate of 10^{-2} yielded the highest results:

	Precision	Recall	F1
Macro Average	0.94	0.94	0.94
Weighted Average	0.94	0.94	0.94

Accuracy	0.94

The following curves show changes in accuracy and loss by epoch when training with this setup:

The resulting correlation matrix for the test data looks like the following:



-1

Following observations can be made:

- Tops get most frequently confused with shirts.
- Pullovers get most frequently confused with tops, coats, and shirts.
- Shirts get most frequently confused with tops and coats.
- Ankle boots get most frequently confused with sneakers.

These confusions make sense from a visual understanding point of view: Tops, pullovers and shirts consist of a bigger middle part to cover one's body as well as sleeves. Ankle boots and sneakers are both types of shoes.

Setup and results for LFW people data – deep methods

Training succeeded with the original dimensions for MiniVGGNet. MiniGoogLeNet however, had had problems: The original pixel size of 28x28 was too small for the model to work with. We increased it to 36x36 for it to work.

For the LFW people dataset, all of the augmentation filters present in Thomas Lidy's notebook were applied. Reason for that being the faces aren't as homogenously laid out as the Fashion MNIST one, as you can see in the following. The faces may be turned or raised or rotated.

In roughly 1000 seconds, we were able to train the MiniVGGNet models for 500 epochs, and 75 epochs for MiniGoogLeNet. The latter yielded superior results.

Both, augmented and non augmented are potential candidates for the best setups. Both setups have their advantages.

Candidate 1: MiniGoogLeNet with augmentation, learning rate: 10^{-1}

	Precision	Recall	F1
Macro Average	0.87	0.83	0.83
Weighted Average	0.90	0.88	0.88

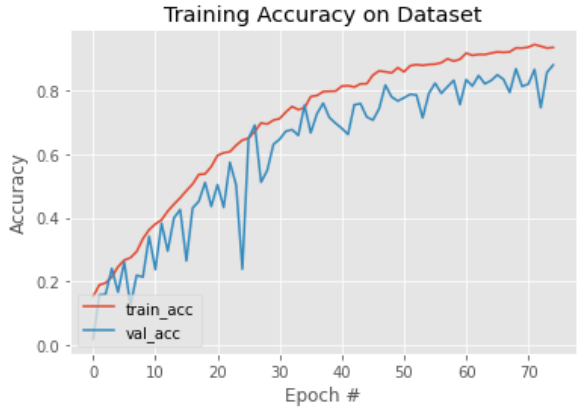
Accuracy	0.88

Candidate 2: MiniGoogLeNet without augmentation, learning rate: 10^{-1}

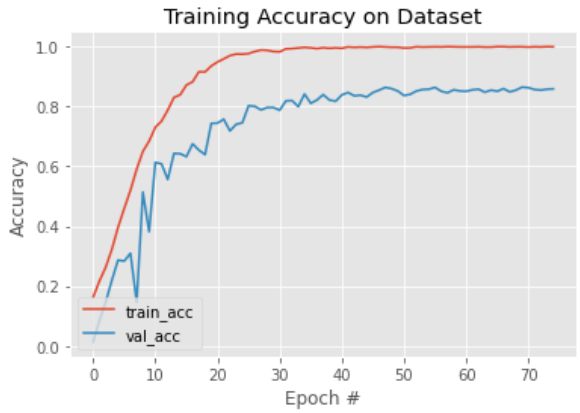
	Precision	Recall	F1
Macro Average	0.84	0.82	0.82
Weighted Average	0.87	0.86	0.86

Accuracy	0.86

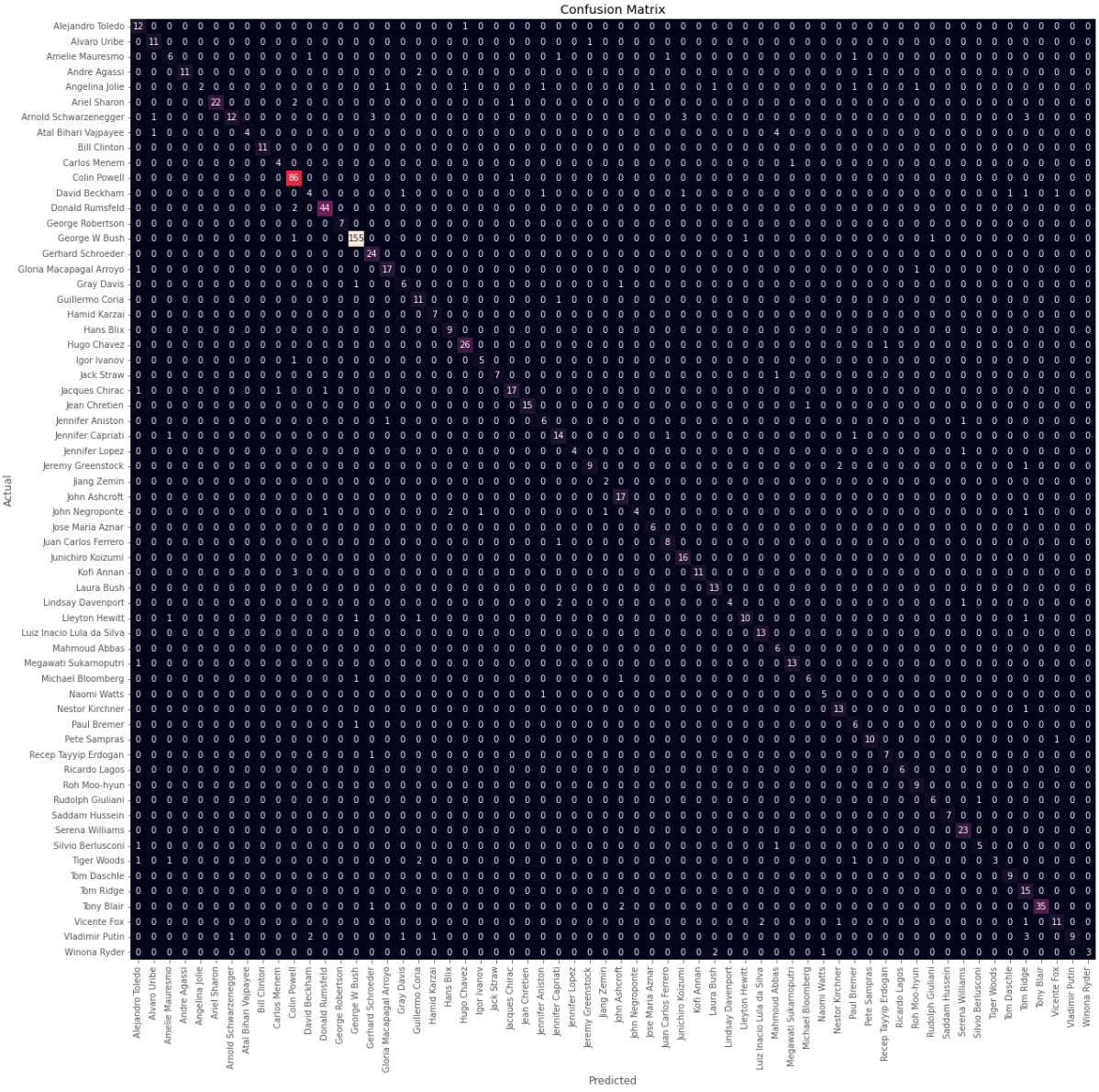
Candidate 1 clearly outperforms Candidate 2 in terms of scores. Candidate 2 has a more stable learning curve. How each of the candidates would perform beyond the set epoch limitations could be basis for further research. Candidate 1:



Candidate 2:



The confusion matrix of Candidate 1 looks like the following:



The graphic does a good job displaying the class imbalance in the dataset. The algorithm does a good job dealing with the class imbalance. We cannot detect patters in which two people are systematically being confused for one another. Misclassifications are rare in total numbers.

In relative numbers, misclassifications are more common. While still rare, John Negroponte was misclassified six out of ten times. People with more training samples are generally less likely to be classified in relative numbers, supporting the expectation that more training data would help with these cases.

Conclusions – deep architectures

Using CNNs, we managed to achieve better scores for both datasets compared to the traditional methods. Most validation curves do not appear to be fully converged with our given limitations, meaning there potential for yet better results.

A disadvantage compared to the traditional methods were the longer training times.

Testing times looked like the following for 1000 samples:

- Fashion MNIST + MinVGGNet: 0.1 seconds
- LFW Faces + MinVGGNet: 0.3 seconds
- Fashion MNIST + MinGoogLeNet: 0.9 seconds
- LFW Faces + MinGoogLeNet: 1.6 seconds

A challenge when running the tests was to get the research to be reproducable with the help of a seed. We saw that Thomas Lidy had set a numpy seed for that purpose. But, this didn't ensure the same training curve in our case. We had tried various seeds that we could set for this purpose online unsuccessfully. Some sources suggested generating multiple samples for reproducability.

While we lacked the time to let the results converge more or to create more samples, it was sufficient to address the assignment task, to compare traditional methods to deep methods: our CNN results consistently have higher scores to our traditional results.

Final Conclusions

The purpose of this assignment is to analyse differences between traditional and deep learning methods, to compare the performance and runtime measures.

Traditional vs deep methods represents a classical case of a interpretability vs predictability tradeoff with the traditional methods being more on the former side of the spectrum and deep methods being very far on the latter side of the spectrum.

Deep learning outperformed traditional methods when it came to scores. Traditional methods outperformed deep learning methods when it came to speed. One can argue that depending on the use case, your limitations and your priorities, both represent methods that may be appropriate. Assuming the resources are available and the task is to have a good predictor, we expect CNNs to do a better job for image related machine learning tasks.