Project 2: A Comparative Study Between Traditional and Information Theoretic Neural Network Models for Fashion-MNIST Classification

Connor McCurley
Deep Learning, Fall 2019
University of Florida
Gainesville, FL, USA 32611
Email: cmccurley@ufl.edu

***Abstract*—This paper explores the utilization of information theory in supervised classification of Fashion-MNIST images. Autoencoder neural networks were trained and used in conjunction with support vector machines to perform classification. Similar multilayer perceptron networks were trained using information theoretic learning. Both models were compared to a baseline convolutional neural network to gauge differences in performance. Classifier performance was evaluated using overall accuracy. Additionally, confusion matrices provided insight to each classifiers' confusers. The baseline CNN obtained a test classification accuracy of 90%, which was the highest amongst the methods compared. This was followed closely by the information theoretic neural network. Due to architectural differences which should have limited performance when compared to the CNN, the accuracy achieved by the information theoretic network suggests the usefulness of ITL in the learning process. It is believed that performance could be improved by considering further classification refinement on the difficult classes, namely: shirt, coat and pullover.**

***Index Terms*—Fashion MNIST, Autoencoder, Information Theoretic Learning, Support Vector Machine**

I. Introduction

AUTONOMOUS image classification is a challenging problem which offers potential for significant advancement in the areas of biometrics, biology, medical diagnosis, security, and more [1], [2]. This paper provides a comparison of information theoretic to non-information theoretic learning approaches for autonomous classification of clothing items in the well-known Fashion MNIST dataset [3].

A wide variety of approaches have been taken in attempt to solve detection and classification problems in imagery. [4] used dissimilarity-based classifiers along with metric learning to dually drive samples toward their respective class representatives while also enforcing separation between classes. [5], [6] utilized vector embeddings with linear support vector machines to discriminate between low-dimensional image representations. The work in [7] found sparse weighted combinations of dictionary atoms to accurately reconstruct images where specific bases equated to the various classes. The authors of [8] utilized statistical properties to match samples to generating distributions. The work in [9] employed traditional template matching to locate objects or compositions in imagery. The review in [10] demonstrated the expansive uses of artificial neural networks in image classification. This, of course, is just a small sample of image classification techniques. The reviews in [1], [2] elaborate extensively on the myriad of methods. A commonality among the discussed methods is that they fail to take advantage of higher-order statistics to capture the error between prediction values [15]. To this end, this work explores the utility of information theoretic cost in terms of classification performance on the Fashion-MNIST dataset.

The remainder of this paper is organized as follows. Section II describes the methodology used in developing three disparate classification systems utilizing artificial neural networks. Classification results are presented in Section III. Practical insights to results are given in Section IV. Finally, Section V reveals concluding remarks and discusses future lines of research.

II. Methodology

This section describes the methodology implemented in this work. Analysis of the data is performed, dimensionality reduction and classification procedures are described, various network architectures under analysis are elaborated on and experimental procedures are outlined.

A. Data Analysis

The following data analysis was originally described in [11], but is pertinent to this work and is thus re-analyzed here. The data was plotted as shown in Figure 1 to gain an understanding of the format. Each sample in the Fashion-MNIST dataset is a 28x28, gray-scale image of a clothing item belonging to one of ten classes [3]. This translates to 784 length feature vectors with values ranging between 0-255. There were exactly 60000 training images included in the training dataset and 10000 which were held-out for test. The 60000 samples were later sub-divided in the experimentation for cross-validation. As with Project 1, dimensionality reduction(DR) was incorporated to reduce data complexity. This DR was employed through the use of stacked autocoder neural networks (SAE). Elaboration on the SAE networks is provided in the following section.

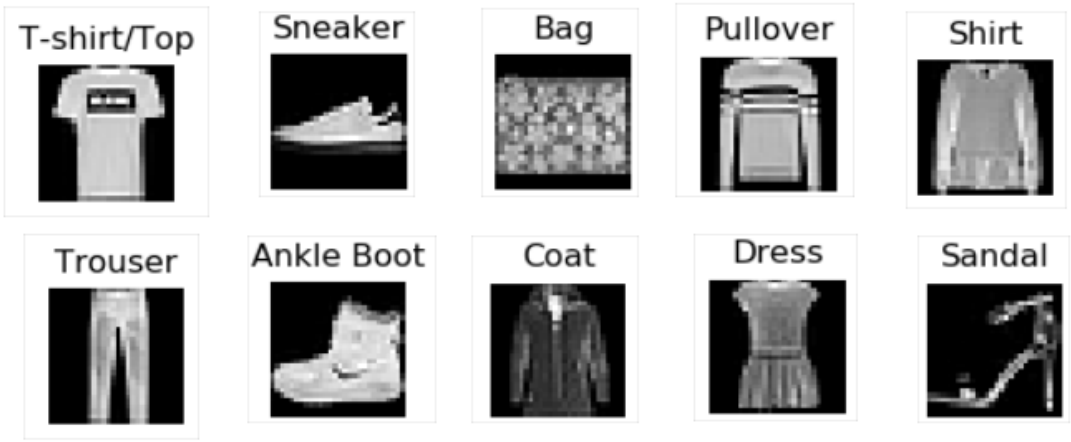


Fig. 1: Samples from the Fashion-MNIST dataset. One sample from each class was randomly chosen for visualization. The gray-scale images are size 28x28, each representing an article of clothing.

B. Autoencoder

Description: An autoencoder is a specific taxonomy of artificial neural network which learns how to compress and de-compress representations of data [12], [13]. The first half of an autoencoder typically performs dimensionality reduction through non-linear transformations until the middle layer, known as the *bottleneck* or *latent* layer. The goal of the *encoder* is to learn efficient representations of the factors which govern variation in the data, or in terms of compression, *codes* which can be used to reconstruct the input data with high accuracy. The second half of an SAE (which is called the *decoder*) projects the data back into its original dimensionality in attempt to reconstruct the original sample.(See Figure 2.) Reconstruction loss between the input and output is used to update the network's parameters. In practice, samples can be passed through the encoder to perform dimensionality reduction.

SAE Architecture: The SAE architecture tested in this work consisted of 5 hidden layers, along with the input and output. The layers were selected as $784 \rightarrow 500 \rightarrow 200 \rightarrow k \rightarrow 200 \rightarrow 500 \rightarrow 784$, where k is the arbitrarily chosen dimensionality of the bottleneck. In this work, k was tested at $[10, 25, 50, 75, 100]$ in order to provided a reasonable comparison of performance changes resulting from dimensionality reduction. ReLU activation functions were used to apply nonlinearity. A sigmoid activation, however, was used at the output layer to enforce image value constraints between $[0-1]$. This was done because the images were normalized between $[0-1]$ before passing through the network. An initial learning rate of $\eta = 0.01$ was selected, and was updated using the Adamax optimizer through training.

SAE Experiments: The SAE network was trained for 20 epochs using mini-batch sizes of 200 samples. The bottleneck layer's dimensionality was varied between $[10, 25, 50, 75, 100]$. Each network configuration was trained 5 times, and the model which provided the lowest reconstruction Mean-Squared Error (MSE) on the hold-out validation set was selected for further use in classification. Results are shown in section III-A.

C. Support Vector Machines

Description: A Support Vector Machine (SVM) is a specific class of sparse kernel machine whose objective is to learn a decision boundary which can adequately discriminate between classes in a high-dimensional space [5], [6]. Because of its sparsity constraints, a SVMs' predictions rely only on a subset of the training data known as *support vectors*. By design, support vectors tend to be examples in the training data which lie closest to the decision boundary, and are thus the most prone to mis-classification. A primary difference between SVM and methods relying on Information Theory is that vanilla SVM classification predictions are not probabilistic [14]. In other words, hard labels are assigned which do not capture the uncertainty of the prediction results. In this work, SVMs were trained on the data passed through the selected autoencoders. A comparison of classification performance on the various sizes of features is provided in section III-C.

Parameters: Non-linear support vector machines were used as the classifiers in this work. The necessary hyperparameters were the type of mapping kernel, kernel parameters, and a regularization (slack) parameter. A radial-basis function was arbitrarily chosen as the mapping function. Silverman's rule was used to provided a reasonable range for the kernel bandwidth. The regularization parameter was set to the Python Scikitlearn's default value of $C = 1$. These parameter choices provided reasonable results for comparison. More parameter variations for the SVM were not tested due to time limitations.

Experiments: SVMs were employed using Scikitlearn's SVM SVC package. The classifiers were trained using one-versus-one training. At test, a sample was applied to the classifier ensemble and the most-likely label was assigned to the sample. Data was passed through each of the top 5 trained autoencoders, and a single SVM was trained to provide label predictions for each input feature dimensionality, $[10, 25, 50, 75, 100]$. Classification performance is presented in section III-C.

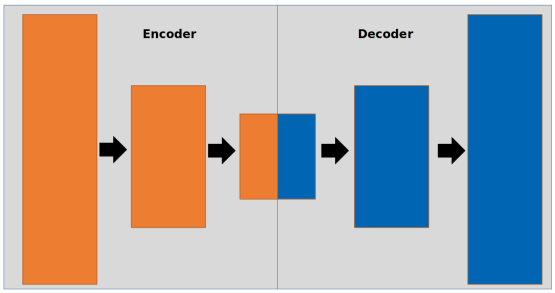


Fig. 2: Block diagram of an autoencoder neural network. The layers consecutively reduce dimensionality until the middle (bottleneck) layer. The second half of the network transforms the data back to the size of the input. The desired value of the network is the original image.

D. Baseline CNN

Description: The autoencoder + SVM classifiers were compared against a baseline convolutional neural network (CNN) architecture. A CNN is a special type of neural network which utilizes weight sharing. A suit of kernels is convolved across the input feature map to detect interesting attributes in an image [12], [13]. CNNs have shown considerable success in a variety of image classification problems. For this reason, a CNN architecture is used as a standard for comparison. Classification performance of the baseline architecture is provided in section III-C.

CNN Architecture: The baseline CNN architecture was chosen based off previous performance on Fashion-MNIST classification [3]. The network consists of two convolutional/ max-pooling layers followed by two fully connected layers. The network provides an output size of 10. This is passed into a soft-max activation function to provide class probability scores. ReLUs are used to provide nonlinearity in each of the hidden layers.

CNN Experiments: The baseline CNN was initialized with a learning rate of $\eta = 0.01$. This was refined during training with the Adamax optimizer. The model was trained 5 times for 20 epochs each. Mini-batches of size 200 were implemented and class prediction scores were evaluated using cross-entropy loss. The model with the best classification accuracy on the hold-out validation set was selected for comparison against the alternative methods in this work.

E. Information Theoretic Learning

The methods previously described are what the author considers as "traditional" forms of learning. In this sense, these methods typically rely on low-order statistics to capture prediction error. Alternatively, Information Theoretic Learning (ITL) approaches have proven effective in a variety of learning applications [15]. These methods take advantage of class distributions to compare higher-order statistics and thus provide richer descriptions of classification error.

Minimum Cross-Entropy (xEnt): One such information-theoretic metric is Minimum Cross-Entropy (MinxEnt). As outlined by Principe in [15], MinxEnt is formulated as a constrained minimization of KL-divergence between the label and predicted-value distributions, p_d and p_y.

$$\min_{p_y} D_{KL}(p_d||p_y) \quad \text{subject to} \quad constraints \tag{1}$$

To implement this comparison, a small amount of Gaussian noise was added to the one-hot encoded label vectors:

$$d_{new} = d + \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{2}$$

where $d \in \mathbb{R}^{10\times 1}$, and σ^2 was arbitrarily chosen as 0.002.

ITL Network Architecture: In order to provide a comparison to the alternative methods, multi-layer perceptron networks were implemented and trained with MinxEnt. The network architectures were designed as $784 \to 500 \to 200 \to k \to 10$ which is consistent with the autoencoder networks defined earlier in this work. The value of k was selected to match the bottleneck sizes of the tested autoencoders and an output layer of size 10 was employed to allow for label comparison. ReLU activations were utilized in each of the hidden layers and a softmax was applied at the output to produce class probability vectors.

ITL Experiments: As with the previous experiments, networks were trained 5 times each with an initial learning rate of $\eta = 0.01$ and updated with Adamax. Mini-batches of 200 were utilized for 20 epochs. For each of the sizes of k, $[10, 25, 50, 75, 100]$, models were trained using a range of bandwidth parameters on the parzen window estimator for xEnt. Specifically, bandwidths of $\sigma = [3, 30, 300, 300]$ were implemented to allow for broad generalization of effects on classification performance. During test, maximum a-priori with priors on the noisy one-hot encoded label vectors was used to assign hard class labels. Comparison of performance based on bandwidth and value of k is provided in Section III-D.

III. Results

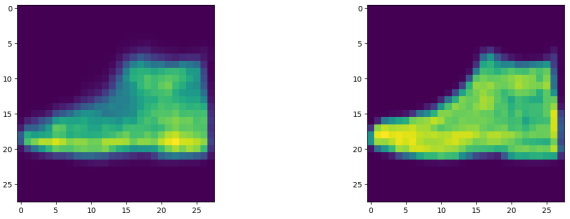
In this section, results from experimentation are presented. Comparison of autoencoder bottleneck size is shown, classification performance on auto-encoded data is presented and a comparison of ITL approaches is revealed.

A. Autoencoder Reconstruction

Each of the autoencoder networks tested achieved near-zero MSE loss. This value was negligible, however, as MSE was being computed on values between $[0, 1]$. This inherently made MSE small. Instead, reconstruction results of a shoe passed through bottlenecks of 10 and 100, respectively, are shown in Figure 3. As can be observed, the reconstructed image from the wider bottleneck retained more fine detail than the smaller. This was at the cost of dimensionality, however.

B. Confusion Matrices

A confusion matrix demonstrates the discrepancies between predicted and true class values for groups of samples. Essentially, it is a way to measure how accurate a classifier is, while providing insight into how the network confuses samples. A



(a) (b)

Fig. 3: Reconstructed images of a shoe after passing through an SAE with bottleneck dimensionality (a) 10 and (b) 100. The images' original dimensionality was 784.

diagonal matrix signifies zero mis-classifications among all categories. Confusion matrices are used in this work to provide insight to classification performance.

C. Comparison of Bottleneck Size

This section presents classification accuracies for the top performing baseline CNN and SVMs trained on the autoencoded data. Table I shows that the baseline CNN achieved the best classification accuracy of 90%. The second-best performance was exhibited by the SVM trained on 100D features, which achieved 86% classification accuracy. Performance dropped sequentially with dimensionality reduction. The lowest performing system was the SVM trained on 10D features. This model obtained 76% classification accuracy. Confusion matrices for the baseline CNN, SVM on 100D data and SVM trained on 10D data are shown in Figure 4. From the figures, it can be observed that the mis-classified classes are fairly consistent between each of the models. The most-commonly confused classes are shirt, pullover, and coat.

D. Comparison of xEnt Kernel Bandwidths

This section demonstrates classification accuracies for the networks trained with MinxEnt. Table II shows test classification accuracies for the networks trained with different values for the parzen window estimator bandwidth and dimensionality of the smallest layer (before 10D final output). Addtionally, confusion matrices for the best-performing models for each latent dimensionality are provided by Figures 5. From the table, it can be seen that a bandwidth of $\sigma = 300$ consistently provided the best results among the parameters compared. The highest performing model was the MLP with a final layer with 100 units and a MinxEnt bandwidth of 300. Moreover, the confusion matrices show that the networks had similar confusers to the previous experiments. The shirt, coat, and pullover classes were the most difficult for the networks to discriminate between.

IV. Discussion

In this sections, observations are made on results and insight is given to potential influences.

TABLE I: Classification Accuracies for the Baseline CNN and AE + SVMs with Varying Bottleneck Dimensionalities

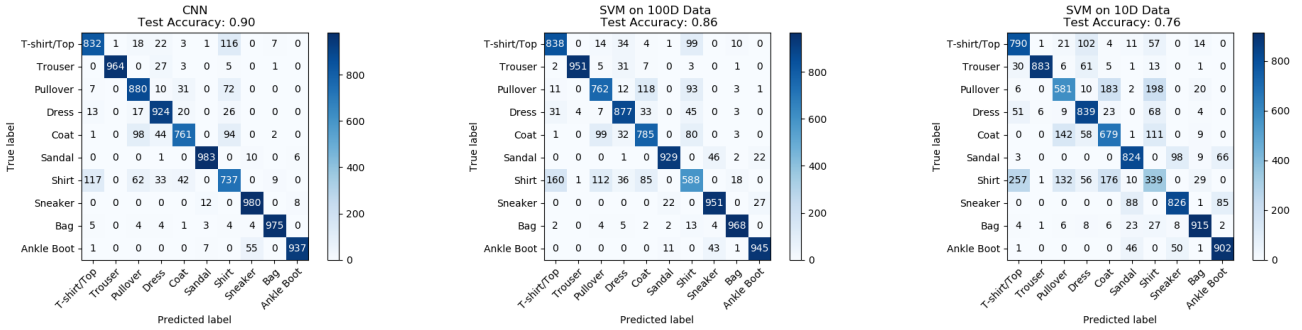
Classification Model	**Accuracy**
Baseline CNN	**0.90**
SVM 100D	0.86
SVM 75D	0.85
SVM 50D	0.84
SVM 25D	0.81
SVM 10D	0.76

TABLE II: Classification Accuracies for Varying xEnt Kernel Bandwidths and Latent Dimensionalities

Latent Dim. k	**Bandwidth** σ	**Accuracy**
100D	3	0.74
	30	0.79
	300	**0.89**
	3000	0.86
75D	3	0.60
	30	0.65
	300	**0.74**
	3000	0.50
50D	**3**	**0.88**
	30	0.87
	300	0.78
	3000	0.71
25D	3	0.72
	30	0.62
	300	**0.73**
	3000	0.52
10D	3	0.68
	30	0.78
	300	**0.80**
	3000	0.66

A. Results and Experimental Design

The first experimentation performed was a comparison of reconstruction error from various autoencoder bottleneck sizes. Because the image values were normalized between $[0, 1]$ when converting to a tensor, and the outputs were passed through a sigmoid activation to scale them appropriately, all MSE values were very low. This made it difficult to evaluate performance quantitatively. However, when plotting images passed through the various networks (as shown in Figure 3), a clear trend could be seen. Specifically, as the bottleneck layer's dimensionality decreased the reconstructed image became less accurate (compared to the original). This result is intuitive as there is a greater loss of information through smaller bottleneck sizes. The goal, in practice, would be to use the bottleneck layer with the smallest dimensionality which still preserved discriminative qualities between the classes. When tacking on a support vector machine, a clear correlation between bottleneck dimensionality and classification accuracy can be seen. They highest performing AE + SVM was the



(a) (b) (c)

Fig. 4: Confusion matrix for the (a) baseline CNN, (b) SVM trained on data passed through an encoder network with bottleneck dimensionality 100 and (c) SVM trained on data passed through an encoder network with bottleneck dimensionality 10. The CNN model classified 90% of the test data correctly while the SVM 100D model classified 86% correctly and the SVM 10D classified 76% correctly.

model which took the bottle neck features of size 100. The performance dropped sequentially with a decrease in bottleneck size. Again, this result is intuitive and, in practice, one would use the smallest feature dimensionality which still allowed the model to achieve a desired level of performance.

Next, multilayer perceptrons of the same size to the encoder networks were employed and trained with MinxEnt. When comparing classification performance over a range of bandwidth values, the sensitivity to this parameter in terms of classification accuracy can be observed. The bandwidths of $\sigma = 3$ and $\sigma = 3000$ consistently resulted in the lowest level of performance, while a $\sigma = 300$ typically provided the best. This suggests that, for this dataset, the optimal bandwidth value might be somewhere in the magnitude of the hundreds. However, performance due to bandwidth, which effects both the magnitude and granularity of the error, might not be well represented in this experimentation. Error could have resulted from the training time of the networks, which was fixed at 20 epochs for every model. A smaller bandwidth may have taken more epochs of training to reach a minimum (which may have been lower than what was achieved).

When comparing the CNN, AE + SVM, and ITL MLP, the differences in performance can be analyzed. The baseline CNN model obtained the best performance among the models tested with an overall classification accuracy of 90%. This was followed in suit by the ITL MLP with a final 100D layer and a bandwidth of $\sigma = 300$ which achieved 89% accuracy and the ITL MLP with a final 50D layer and a bandwidth of $\sigma = 3$ which achieved 88% accuracy. The best performing AE + SVM model was the one with a bottleneck size of 100. This classification system obtained 86% classification accuracy on the hold-out test set. These results match the author's intuition. To begin, autoencoders are not designed to preserve class discriminability in the latent space. They specifically attempt to compress the data into representations which capture the highest level of variation in the dataset. On images of faces, for example, latent features might represent (presumably) color, pose, eye shape, lighting variations, etc. While these features might provide appropriate features for class discrimination, there is no guarantee they will meet this objective. Therefore, one would expect the AE + SVM networks to achieve lesser classification accuracy than the alternatives compared in this work. Next, while both the CNN and ITL MLP methods were trained to learn features which promote discriminability, the author expected the CNN to outperform the MLP. This expectation was simply based off influence from the literature which suggests that convolutional features are able to capture interesting image artifacts easier than MLPs due to their inherent consideration of spatial information. Therefore, it was somewhat surprising that the ITL MLP was able to achieve comparable classification performance. This suggests the benefit of information theoretic learning approaches compared to what the author would consider as "traditional" learning strategies. Because ITL takes advantage of higher-order data statistics, a more-accurate measure of error between the prediction and label distributions can be computed. This better representation of error can guide a learning algorithm to potentially locate the optimal parameter values of a model. From this insight, it would be interesting to compare the baseline CNN to a CNN implementing an information theoretic cost function. Presumably, this may only significantly alter the parameters of the fully connected layers, but the author believes it would be interesting to evaluate how the convolutional features are altered through the backpropagation.

It should be noted that, due to time limitations, hyper-parameters for each method were chosen only to provide a rough generalization of performance over a wide range of parameter values. Hyperparameters were not necessarily tuned in any scenario, which could mis-lead results concerning the "overall best" algorithm. If given more time, a more thorough parameter search would be conducted in attempt to discover the optimal models for comparison.

In this work, models were only trained five times each, and

the best performing model on validation data was analyzed further. It would have, however, been more insightful to the stability of the methods to run an extensive cross-validation analysis. This type of analysis could provide bounds on performance variations due to training and could highlight the robustness of particular methods which may not have been captured by this set of experiments.

It should also be noted that run-time was not compared in this work. This was due to the fact that the author was switching between machines when performing this experimentation, which resulted in highly-varying run-times for the same algorithms. For clarity, this should not be done when demonstrating performance of methods, but the author felt it was sufficient for this work.

B. Potential Improvements

In addition to a more thorough study of optimal parameter selection, two methods which could potentially aid with overall system performance might be to include an additional form of pre/post-processing to improve the confused classes and to train the model with a type of rank loss. It terms of pre/post-processing, additional models could be trained to learn explicit differences between the coat, shirt and pullover classes, and between the top and dress classes. This could done beforehand by learning alternative features representations for these samples before being passed to the main network, or after-the-fact with additional classification on any labels predicted to be in those difficult classes. This could even be implemented in an ensemble approach to (hopefully) improve classification performance on the most-likely confused classes. Another alternative would be to employ a form of rank-loss, such as contrastive or triplet. In this scenario, additional constraints could be applied to the objective function to enforce inter-class separability between disparate classes and promote intra-class compactness. These approaches utilize pairs or triplets of similar and dissimilar data samples to embed samples in close/far relations according to a predefined metric. The author believes this could be applied in conjunction with the cost functions utilized in this work to improve results on the difficult classes.

V. Conclusions

Classification of Fashion-MNIST images using information theoretic and non-information theoretic cost functions was investigated in this work. Dimensionality reduction was performed using autoencoder neural networks and classification was performed using support vector machines. Additionally, multilayer perceptrons were trained using minimum cross-entropy. Both classification frameworks were compared to a baseline CNN model. The baseline achieved the highest classification accuracy amongst the models compared with a 90% correct classification rate. Specific implementations of the AE + SVM and ITL MLP networks obtained comparable results.

The author believes that the primary differences in performance between the models comes from their methods of feature learning. Autoencoders learn latent representations to preserve variation in the data, and do not necessarily learn latent features which are optimized for discrimination. However, both the CNN and ITL MLP networks are trained with classification in mind and thus have an advantage over the autoencoded features. Moreover, as can be observed from the literature, convolutional features seem to represent image qualities better than traditional MLPs. However, the ITL MLP was able to obtain comparable results to the CNN despite this. This might suggest the usefulness of the information theoretic cost function in improving classification results. Performance might be improved more using a convolutional network with an information theoretic cost function.

Future research endeavors toward this topic include pre/post-processing or ensembling to improve classification results on the difficult-to-classify classes and investigating the incorporation of rank-loss through metric embedding to provide improved separability of classes in the feature space.

Honor Statement

* I confirm that this assignment is my own work, it is not copied from any other person's work (published or unpublished), and has not been previously submitted for assessment either at University of Florida or elsewhere.

References

[1] S. Prasad, T. S. Savithri, and I. V. M. Krishna, "Techniques in image classification; a survey," vol. 15, 2015.

[2] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, pp. 823 – 870, 03 2007.

[3] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[4] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2624–2637, Nov 2013.

[5] J. Sanchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1665–1672.

[6] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and svm training," in *CVPR 2011*, June 2011, pp. 1689–1696.

[7] S. Shao, Y.-J. Wang, B.-D. Liu, W. Liu, and R. Xu, "Label embedded dictionary learning for image classification," 2019.

[8] R.Timofte, T. Tuytelaars, and L. van Gool, 2013.

[9] P. R. S. Swaroop and N. Sharma, "An overview of various template matching methodologies in image processing," 2016.

[10] S. B. Driss, M. Soua, R. Kachouri, and M. Akil, "A comparison study between mlp and convolutional neural network models for character recognition," in *Commercial + Scientific Sensing and Imaging*, 2017.

[11] C. H. McCurley, "Project 1: Manifold learning for fashion-mnist classification with multi-layer perceptrons," 2019.

[12] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 2009.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[14] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[15] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, 1st ed. Springer Publishing Company, Incorporated, 2010.

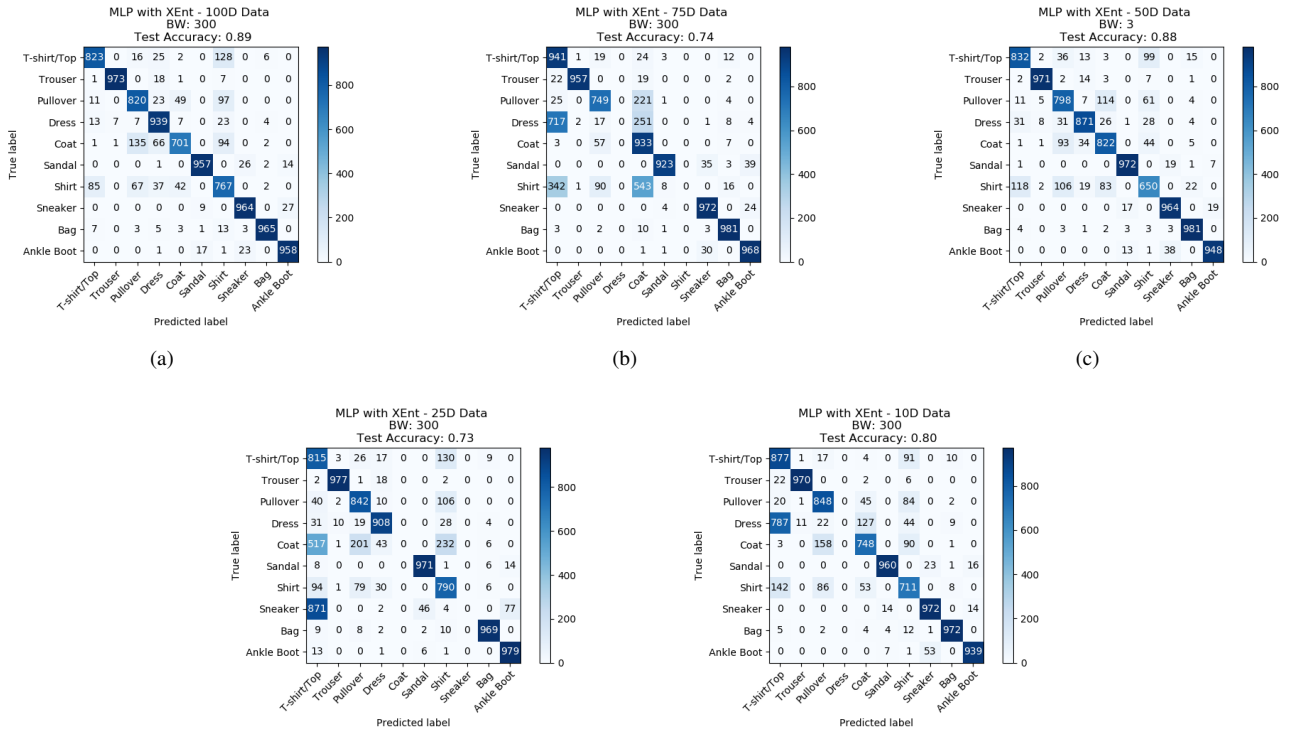


Fig. 5: Confusion matrices for the ITL MLPs with a final dimensionality (before output) of (a) $k = 100$ and a xEnt bandwidth of $\sigma = 300$ which classified 89% of the test data correctly, (b) $k = 75$ and a xEnt bandwidth of $\sigma = 300$ which classified 74% of the test data correctly, (c) $k = 50$ and a xEnt bandwidth of $\sigma = 3$ which classified 88% of the test data correctly, (d) $k = 25$ and a xEnt bandwidth of $\sigma = 300$ which classified 73% of the test data correctly and (e) $k = 10$ and a xEnt bandwidth of $\sigma = 300$ which classified 80% of the test data correctly.