# Project 1: Manifold Learning for Fashion-MNIST Classification with Multi-Layer Perceptrons

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

***Abstract*—This paper investigates the use of manifold learning as a preprocessing procedure for object classification in grayscale imagery. Three manifold learning methods exhibited in the literature were compared in terms of their contributions to enforcing class discriminability. Two individual multi-layer perceptron classifier architectures were trained for each dimensionality reduction technique and compared to a baseline model. Experiments were conducted on the Fashion-MNIST dataset and results were presented in the form of confusion matrices. The optimal detector demonstrated performance of 0.66 accuracy, which was an increase of 0.27 over the baseline. More experimentation could be performed to optimize the parameters of each manifold learning method and to potentially discover the true intrinsic dimensionality of the Fashion-MNIST dataset. Additionally, it is believed that performance could be improved with better feature extraction and more careful training.**

***Index Terms*—Neural Network, Dimensionality Reduction, Manifold Learning, Multi-Layer Perceptron, Fashion MNIST**

## 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 focuses on the use of dimensionality reduction/ manifold learning in conjunction with multi-layer perceptron artificial neural networks to automatically classify clothing items from 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 that 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 all of the discussed methods is that they suffer from high-dimensionality. Because of this fact, this work explores the use of dimensionality reduction as a preprocessing procedure for classification with fully-connect multi-layer perceptrons.

The remainder of this paper is organized as follows. Section II describes the methodology used to perform dimensionality reduction and classification with multilayer perceptrons. 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 used throughout this work. Analysis of the data is performed, dimensionality reduction techniques are described, various network architectures under analysis are elaborated on and the experimental procedure is outlined.

### A. Data Analysis

The data was first 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. Histograms of one-versus-all Euclidean distances to each of the classes are shown in Figure 2 to gain a sense of class separability using the raw images, solely. Given that the classifier would be a multi-layer perceptron artificial neural network, it was determined that dimensionality reduction should be utilized to combat the Curse of Dimensionality, while potentially improving class discriminability.

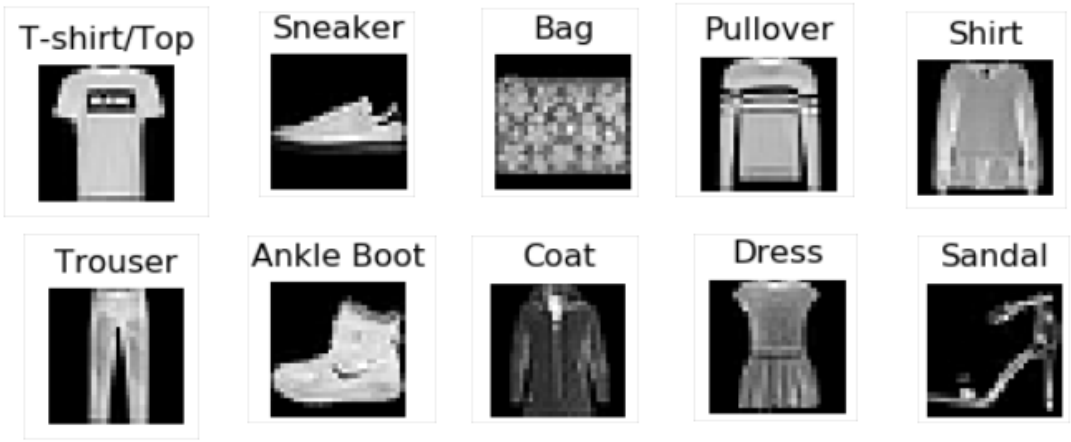



Fig. 1: Samples from the Fashion-MNIST dataset. One sample from each class was randomly chosen for visualization. The gray-scale images are size 28x28, each representing an article of clothing.

### *B. Dimensionality Reduction*

Manifold learning, feature extraction, dimensionality reduction (DR) and representation learning are all synonymous for methods that learn representations of data that make it easier to extract useful information when building classifiers or other predictors [11]. Traditionally, DR transforms high-dimensional data into meaningful representations of reduced dimensionality. There is an expansive taxonomy of DR techniques, ranging from linear to non-linear, globally preserving to locally preserving, variance retaining to discriminability enforcing, among others [12]. Dimensionality reduction has been used in a wide variety of applications, including: speech recognition and signal processing, object recognition, computer vision, multi-task learning and domain adaptation [11], multi-modal sensor alignment [13], [14], pose estimation [15], land-use classification [16], medical diagnosis, meteorology, environmental monitoring, economic forecasting and more [17]. Nine methods were originally considered for this work, constituting a mix of linear and non-linear techniques. These methods include: Principal Component Analysis (PCA) [18], [19], Fisher's Linear Discriminant (FLDA) [19], [20], t-Distributed Stochastic Neighbor Embedding (t-SNE) [21], Uniform Manifold Approximation and Projection (UMAP) [22], Auto-encoders [23], [24], Self-Organizing Feature Maps (SOM) [23], [25], [26], Isometric Feature Mapping (Isomap) [12], [27], [28], Locally Linear Embedding (LLE) [29], [30], and Laplacian Eigenmaps [12], [31]. While each of these methods have shown efficacy in various arenas, only three methods were selected for further review in this work. Principal Component Analysis is a well-known method for dimensionality reduction which is unsupervised. UMAP considers label information to ensure within-class compactness and between-class separation. Autoencoder networks were also used to provide a neural approach. They are referred to as "unsupervised" in this work because they do not consider class information when defining latent feature representations. These techniques are further described in the following:

*Principal Component Analysis (PCA):* PCA is arguably the most widely known, and commonly used, dimensionality reduction technique. The goal of PCA is to transform data along its principle axes so that maximum variance is retained in the new coordinate space [18], [19]. A key assumption when using PCA as a preprocessing step for classification is that the most discriminative features are also the most highly varying. Given that PCA is a linear dimensionality reduction technique, it was unreasonable to assume that it would be able to approximate highly nonlinear manifolds well. However, linear methods are applicable to out-of-sample datapoints, meaning the transformation can easily be applied to new data during test. While it might have made more sense to employ Fisher's Linear Discriminant for the linear dimensionality reduction technique (because its goal is to enforce discriminability), FLDA limits the dimensionality of the latent representation to one less than the number of classes. For this reason PCA was used to project the data into varying low-dimensional spaces.

*Uniform Manifold Approximation and Projection (UMAP):* UMAP is currently a state-of-the-art method for dimensionality reduction and data visualization which is a generalization of t-Distributed Stochastic Neighbor Embedding (t-SNE). To explain the workings of UMAP it is first beneficial to review SNE. Stochastic Neighbor Embedding (SNE) uses radial basis functions to convert high-dimensional Euclidean distances between datapoints into conditional probabilities representing similarities. SNE then uses a cost based on KL divergence to match pairwise distributions between data representations in the high and low-dimensional spaces. Since KL divergence is asymmetric, different types of errors are not weighted equally in the cost. For example, there is a larger penalty for using very dissimilar points to represent nearby datapoints. t-SNE is a modification of SNE in which the RBF kernels are replaced by the heavy-tailed student-t distribution [21] to alleviate crowding and optimization problems present in SNE. While outwardly

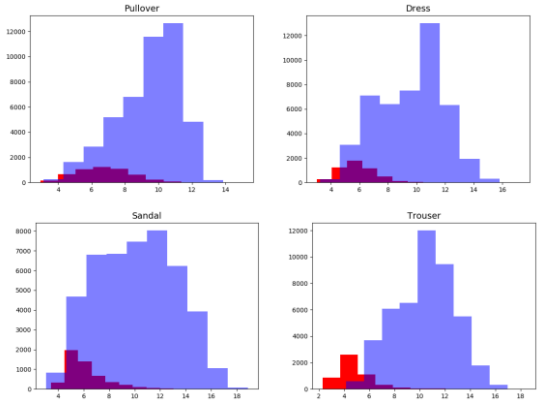



Fig. 2: Histograms of one versus all Euclidean distances for top left): Pullover, top right): Dress, bottom left): Sandal, and bottom right): Trouser classes. The red bars represent Euclidean distances for samples of the given class to their mean. Blue represents the distance of every other training sample to the same mean.

equivalent to t-SNE, UMAP addresses some of the pitfalls of t-SNE. UMAP substitutes a binary cross-entropy cost function for KL-divergence which makes it capable of capturing global structure, and by ignoring probability normalization, the time for high-dimensional graph computation is drastically reduced [22].

*Autoencoder:* An autoencoder is a specific taxonomy of artificial neural networks whose output has the same dimensionality as the input and whose desired is actually the input sample [23], [24]. Typically, autoencoders enforce dimensionality reduction operations up to their middle layers. This portion of the network is known as the 'encoder', since it projects data into a lower dimensional space. The second half of the network projects the data back into its original dimensionality in attempt to reconstruct the original sample. This section of the network is known as the 'decoder'. (See Figure 3.) In practice, samples can be passed through the encoder to perform dimensionality reduction.

### C. Network Architecture

Three individual multi-layer perceptron architectures were implemented in this work. Each of the architectures maintained similar structure with the exception of the input layer, which was varied to test the effects of dimensionality reduction. After the input, the networks consisted of layers (input-256-128-100-10). The input sizes were varied from the original dimensionality (784), to reduced dimensionality of 400 and 100. ReLU activation functions were used in all layers of the network, excluding the output. Cross-entropy was the utilized cost function and the Adamax optimizer implementation in Pytorch was used to update the weights.

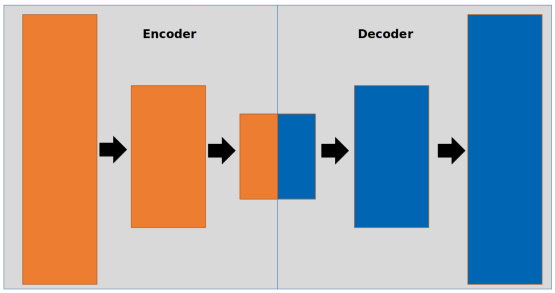



Fig. 3: Block diagram of an autoencoder neural network. The layers consecutively reduce dimensionality before increasing back to the input size. The desired value of the network is the original image.

### D. Experiments

Experiments were conducted to test the effects of various dimensionality reduction approaches on image classification with multi-layer perceptron artificial neural networks. The baseline architecture described in Section II was first trained with the original dimensionality of the data. Each of the networks trained in this work were updated on 51000 training images and validated on 9000 (evenly distributed) samples. Each model was trained in batch 5 times with a learning rate of $\eta = 0.01$. The best-performing classifiers on the held-out validation set were then applied to the blind test set to obtain final accuracies. Early stopping was applied when the validation loss began to increase. Since cross-entropy was used as the loss function, a lower score implied more accurate results. Networks were trained for the following situations:

1) *Baseline:* Model trained on the original data (described in section II).
2) *PCA 100:* Principal vectors estimated from the training set were used to project the data down to 100 dimensions, thus retaining 91.1% of the original variance. The model was the same as the baseline excluding the input layer.
3) *PCA 400:* Principal vectors estimated from the training set were used to project the data down to 400 dimensions, thus retaining 98.5% of the original variance. The model was the same as the baseline excluding the input layer.
4) *UMAP 100:* UMAP was used to project the data down to 100 dimensions. 15 neighbors were used in the local neighborhood and the minimum distance of points in the latent space was constrained to 0.1, which was measured by Euclidean distance. The model was the same as the baseline excluding the input layer.
5) *UMAP 400:* UMAP was used to project the data down to 400 dimensions. 15 neighbors were used in the local neighborhood and the minimum distance of points in the latent space was constrained to 0.1, which was measured by Euclidean distance. The model was the same as the baseline excluding the input layer.
6) *AE 100:* An additional multi-layer perceptron was first

trained in the form of an autoencoder. This MLP demonstrated layers as (784-500-400-150-100) in the encoder with the reverse in the decoder. A hyperbolic tangent activation was applied at the output layer. The network was trained with mean-square error loss and updated with Adamax. The original dataset was passed through the encoder to reduce dimensionality. The model was the same as the baseline excluding the input layer.

7) *AE 400:* An additional multi-layer perceptron was first trained in the form of an autoencoder. This MLP demonstrated layers as (784-500-400) in the encoder with the reverse in the decoder. A hyperbolic tangent activation was applied at the output layer. The network was trained with mean-square error loss and updated with Adamax. The original dataset was passed through the encoder to reduce dimensionality. The model was the same as the baseline excluding the input layer.

Classification results were quantified in the form of Confusion Matrices. Classifier performance is provided in section III.

## III. Results

In this section, classification results for each experimental method tested are presented in the form of 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 diagonal matrix signifies zero mis-classifications among all categories.

### A. Confusion Matrices

Figures 4,5,6, 7, 8, 9, and 10 demonstrate confusion matrices for all 7 trained MLPs. Table I shows the cross-entropy loss on a blind test set for each dimensionality reduction/ MLP pair. It can be observed that preprocessing with Principal Component Analysis to project the data to 100 dimensions resulted in the lowest cross-entropy loss. Alternatively, the worst performance was exhibited by the PCA projection into 400 dimensions, with cross-entropy loss of 2.23.

TABLE I: Cross Entropy Loss on 10000 Blind Test Samples

| **Dimensionality Reduction** | **Cross-Entropy Loss** |
|---|---|
| Baseline | 0.61 |
| **PCA 100** | **0.34** |
| PCA 400 | 2.23 |
| UMAP 100 | 0.86 |
| UMAP 400 | 0.87 |
| AE 100 | 1.30 |
| AE 400 | 0.99 |

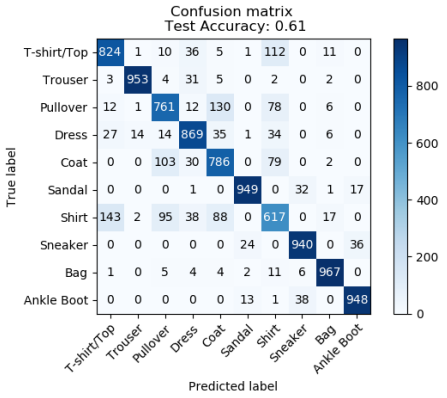



Fig. 4: Confusion matrix for the base MLP on 10000 blind test images.

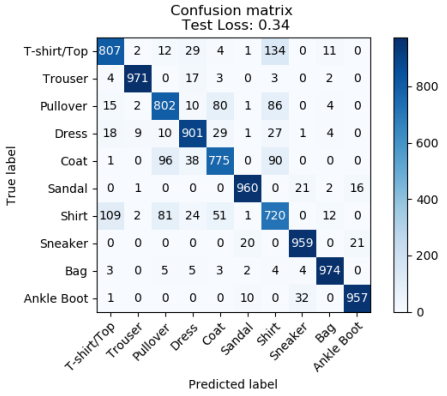



Fig. 5: Confusion matrix for the PCA 100 model on 10000 blind test images.

## IV. Discussion

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

### A. Results

The best classifier exhibited in this paper obtained a cross-entropy loss of 0.34. This is below state-of-the-art object detectors and classifiers in imagery. As can be observed from the confusion matrices, there were consistent errors among each of the models. Specifically, the "Shirt" class had the most mis-classifications between each classifier. This makes sense when looking at the confusion matrices, however. It can be seen that the Shirt class was most commonly mis-classified as Coat, Pullover, or T-shirt/Top, each of which could arguably

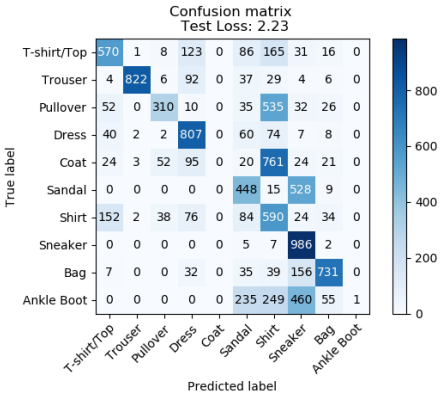



Fig. 6: Confusion matrix for the PCA 400 model on 10000 blind test images.

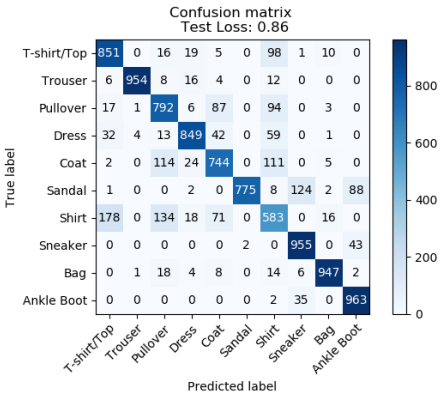



Fig. 7: Confusion matrix for the UMAP 100 model on 10000 blind test images.

full under the hierarchy of Shirt. This signifies methods addressing label uncertainty or class hierarchy are potentially needed to improve results. Moreover, there was a fair amount of inconsistency between expected and actually performance. Given that PCA only retains variance through its projections, it is intuitive that UMAP (which enforces class discriminability) should have made it easier for the network to distinguish the individual classes. This idea is further enforced by Figures 11 and 12 which show PCA and UMAP embeddings of the training data in 2D. It is clear that UMAP better preserves intra-class compactness while enforcing between-class dissimilarity. The one versus all Euclidean distances after UMAP projection into 100 dimensions (Figure 13) clearly show that the classes were pushed apart from the base data. However, PCA 100 still greatly out-performed the other methods. The author can attribute this to a few points. First, PCA in 100 dimensions might just be a good space for this data (although it is impossible to know, and the 2D visualizations may not be accurate depictions of what is happening in higher dimensions). Secondly, error likely came from the stochastic nature of training. Although each network was trained 5 times with random initialization to

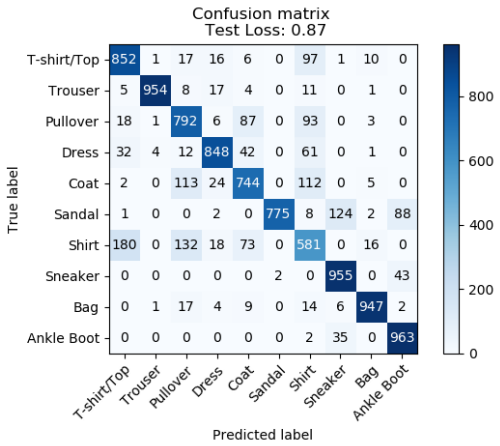



Fig. 8: Confusion matrix for the UMAP 400 model on 10000 blind test images.

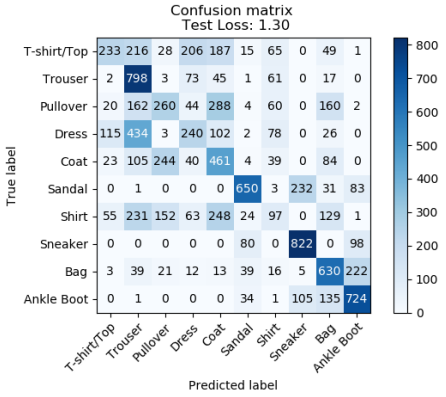



Fig. 9: Confusion matrix for the autoencoder 100 model on 10000 blind test images.

explore the weight-space, some models might have stumbled into better optima than others. The autoencoder methods were at an even greater disadvantage since the encoders had to be stochastically trained before the classifier. While this method of training has proven to demonstrate good results, more work could be done to find improved models and offer better comparisons of approaches. Finally, the parameters of the dimensionality reduction techniques (as well as the feature representations they demonstrate) could be sub-optimal. This suggests that more-suited dimensionality reduction and feature representation techniques could potentially improve classification results on this dataset.

### *B. Potential Improvements*

In addition to the methods mentioned above, two mechanisms which could aid with image classification include

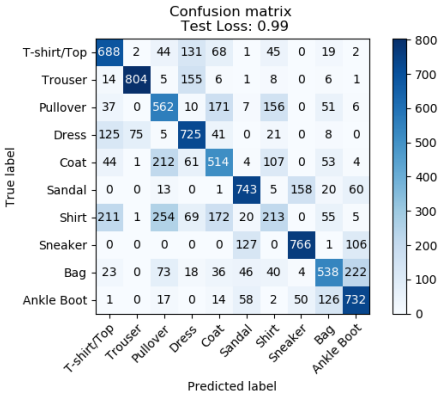



Fig. 10: Confusion matrix for the autoencoder 400 model on 10000 blind test images.

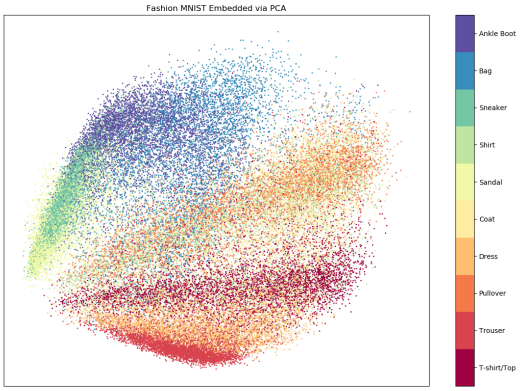



Fig. 11: Fashion MNIST emmbedding of the training set via PCA

discriminative manifold learning/ feature representation and improved training procedures with data augmentation. While finding a lower dimensional representation for a datapoint, it is intuitive that if the final goal is classification, that the DR method should enforce between-class discriminability while ensuring within-class compactness. UMAP was the only method of the three compared which exhibited these qualities, and the results were (visually) promising for discrimination tasks. Moreover, even though the MLPs are universal function approximators, there are better-suited methods for feature extraction in images. Specifically, convolutional neural networks and related methods consider spatial information which provides another level of context for the classier. Finally, for networks with this staggering numbers of features, 51000 training samples is simply not enough to generalize. Data augmentation (along with a more detailed training regimen) would likely improve classification performance on the unseen test set.

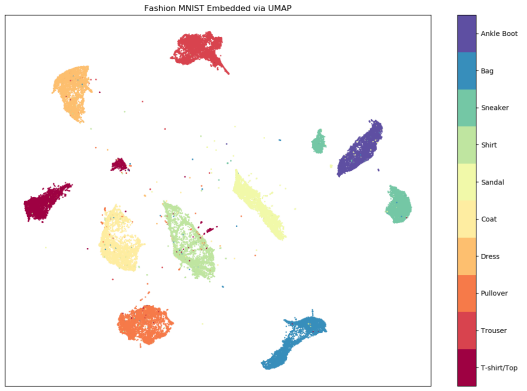



Fig. 12: Fashion MNIST embedding of the training set via UMAP

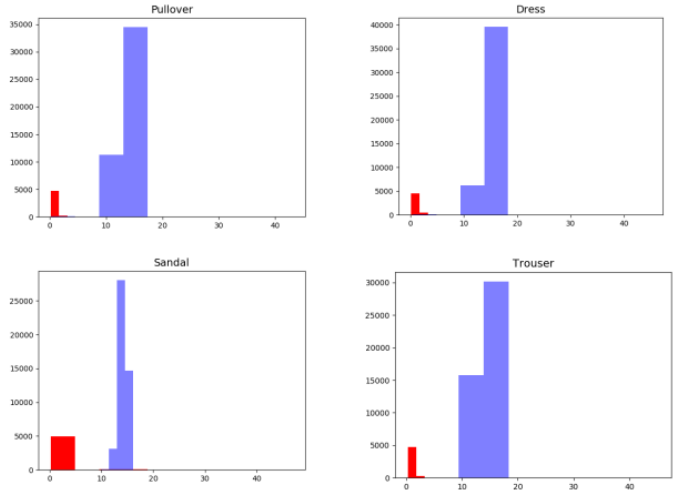


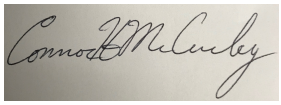

Fig. 13: Histograms of one versus all Euclidean distances after projecting to 100 dimensions with UMAP for top left): Pullover, top right): Dress, bottom left): Sandal, and bottom right): Trouser classes. The red bars represent Euclidean distances for samples of the given class to their mean. Blue represents the distance of every other training sample to the same mean.

## V. Conclusions

Three dimensionality reduction techniques were tested as preprocessing procedures for images classification with multi-layer perceptrons.

## 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] Y. Bengio, A. C. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," *CoRR*, vol. abs/1206.5538, 2012. [Online]. Available: http://arxiv.org/abs/1206.5538

[12] L. van der Maaten, E. O. Postma, and J. van den Herik, "Dimensionality reduction: A comparative review," 2009.

[13] J. Zhang, "Multi-source remote sensing data fusion: Status and trends," *International Journal of Image and Data Fusion*, vol. 1, pp. 5–24, 03 2010.

[14] M. A. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "Joint manifolds for data fusion," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2580–2594, Oct 2010.

[15] R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla, "The joint manifold model for semi-supervised multi-valued regression," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[16] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (lema): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 147, pp. 193 – 205, 2019.

[17] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003.

[18] M. E. Tipping and C. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 21, no. 3, pp. 611–622, January 1999. [Online]. Available: https://www.microsoft.com/en-us/research/publication/probabilistic-principal-component-analysis/

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

[20] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 905–912.

[21] L. van der Maaten and G. E. Hinton, "Visualizing data using t-sne," 2008.

[22] L. McInnes and J. Healy, "Umap: Uniform manifold approximation and projection for dimension reduction," *ArXiv*, vol. abs/1802.03426, 2018.

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

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

[25] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[26] B. Fritzke, "A growing neural gas network learns topologies," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, ser. NIPS'94. Cambridge, MA, USA: MIT Press, 1994, pp. 625–632. [Online]. Available: http://dl.acm.org/citation.cfm?id=2998687.2998765

[27] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. [Online]. Available: https://science.sciencemag.org/content/290/5500/2319

[28] N. Thorstensen, "Manifold learning and applications to shape and image processing," 2009.

[29] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. [Online]. Available: https://science.sciencemag.org/content/290/5500/2323

[30] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," *Journal of Machine Learning Research*, vol. 7, 01 2001.

[31] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.