Fashion-MNIST

Carl Edwards
Alec Yen

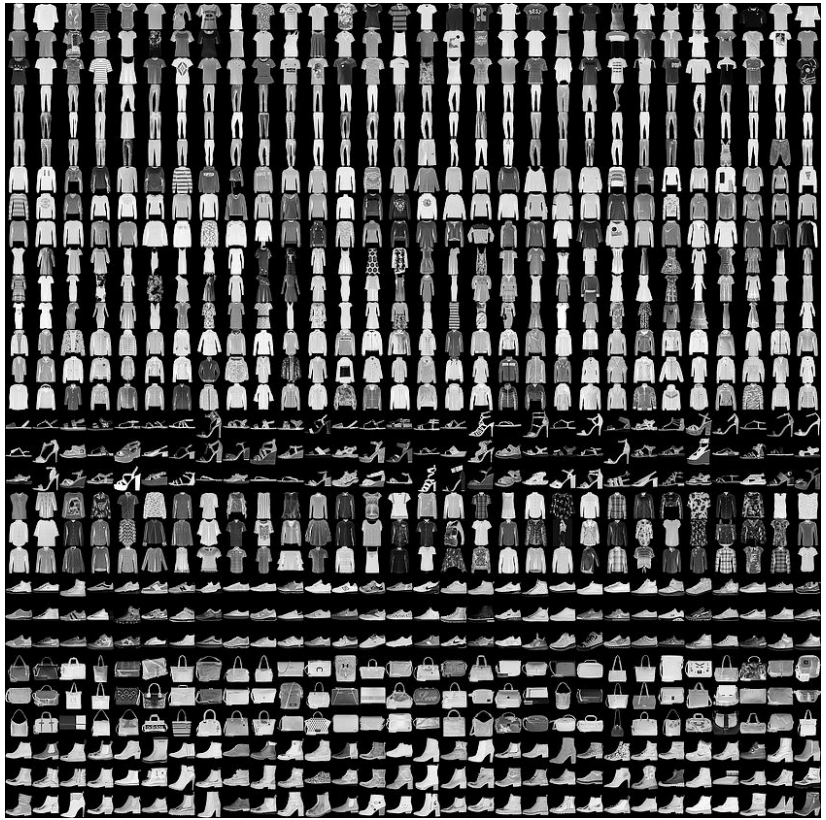
ECE 471: Intro to Pattern Recognition
Final Project

About the Dataset

- The original MNIST (1998) is a popular 10 category dataset of images of 70,000 handwritten digits
- Fashion-MNIST was introduced by Han Xiao, Kashif Rasul, and Roland Vollgraf in 2017 aiming to provide a more challenging classification **problem: classify the fashion articles**
    - Deep neural networks could attain 99.7% on the original MNIST
- Fashion-MNIST consists of 28x28 grayscale images, 60000 training and 10000 testing
- Our Motivation: Evaluate classifiers from class on a challenging high-dimensional benchmark image classification dataset and improve understanding of computer vision.
- Challenges
    - High dimensionality (784 features)
    - The non-Gaussian nature of the data
    - Higher variation within classes than MNIST
    - Similar classes (e.g. coats and t-shirts)
- State of the Art
    - The current leader, Andrew Brock, has achieved a remarkable accuracy of 96.7% using wide residual networks (WRN) in PyTorch and 8,900,000 parameters
    - He uses a novel technique of training networks in which he progressively freezes layers, thereby accelerating training

About the Dataset

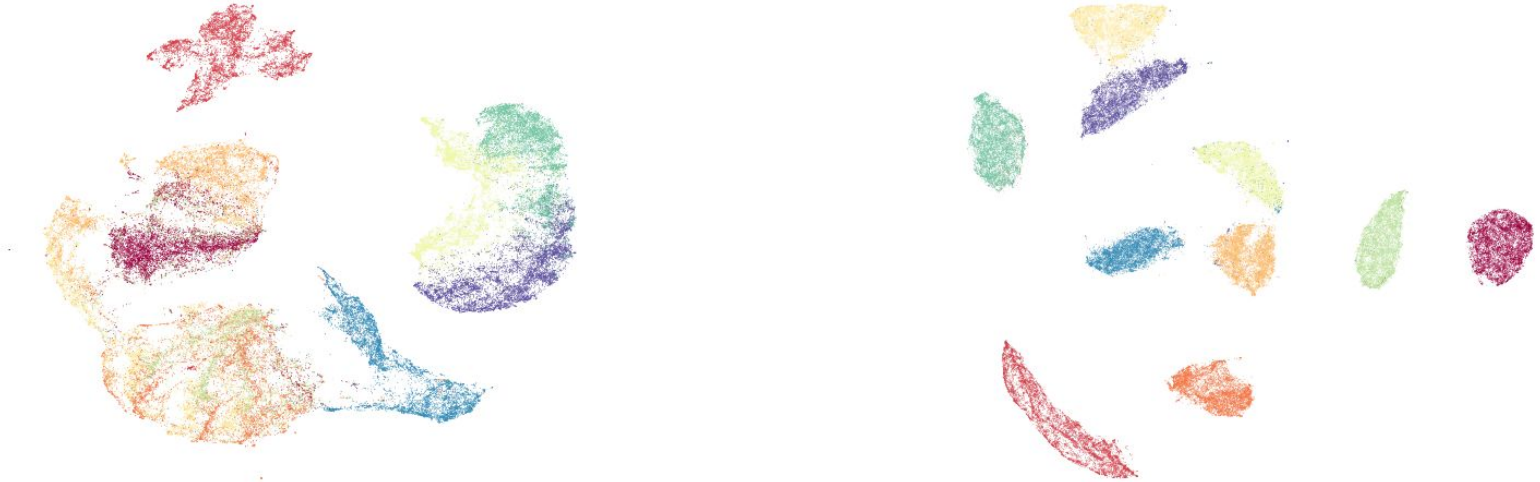
- The categories were as follows
  - 0 - T-shirt/Top
  - 1 - Trouser
  - 2 - Pullover
  - 3 - Dress
  - 4 - Coat
  - 5 - Sandal
  - 6 - Shirt
  - 7 - Sneaker
  - 8 - Bag
  - 9 - Ankle Boot
- There were the same number of items in each category in the training as well as in the testing set



https://github.com/zalandoresearch/fashion-mnist/blob/master/doc/img/fashion-mnist-sprite.png

Visualizing Fashion-MNIST and MNIST

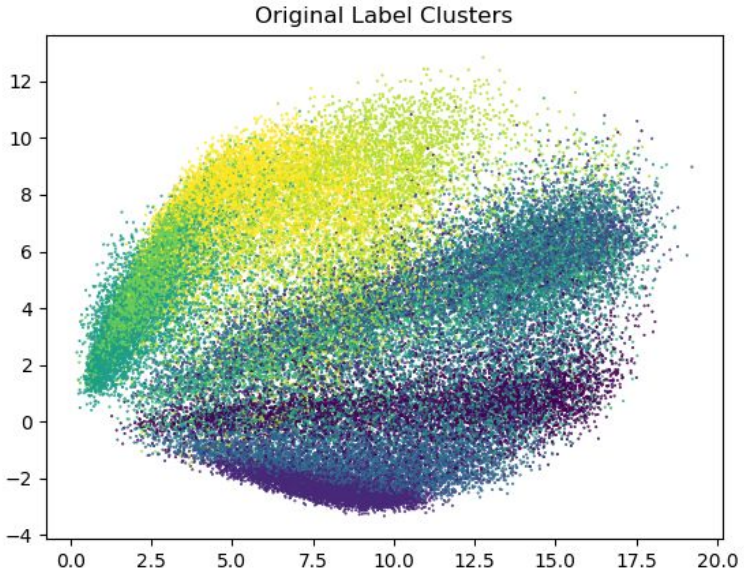
- Uniform Manifold Approximation and Projection
- On the left is Fashion-MNIST. On the right is the original MNIST.
- Fashion-MNIST has more mixing between the classes.

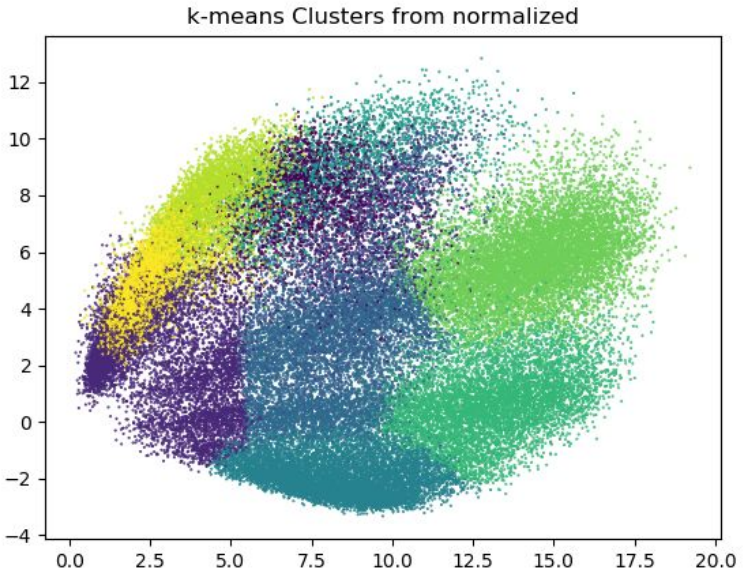


https://github.com/zalandoresearch/fashion-mnist

Clustering Visualization with PCA (d=2)

- The number of dimensions was reduced to 2 using PCA
- k-means and winner-takes-all were tested





Algorithm Summary

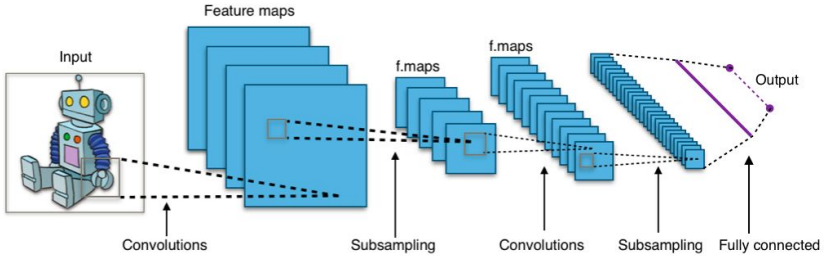
- Dimensionality Reduction
  - PCA
  - FLD
- Supervised Classification
  - Case 1 - Minimum Euclidean distance
  - Case 2 - Minimum Mahalanobis distance
  - Case 3 - Nonlinear decision boundary
  - k-Nearest Neighbors
    - Different k's (k=5,10,20,50,100,250)
    - Different Minkowski distances (p=1,2,3,infinity)
  - Decision Trees
  - Backpropagation Neural Network (BPNN)
    - Different numbers of hidden nodes
  - Convolutional Neural Network
- Unsupervised Classification
  - k-Means
  - Winner-takes-all
- Classifier Fusion
- M-Fold Cross Validation

Experimental Design

- The data was normalized and dimensionality reduction was employed
- Testing was done on the (1) normalized, (2) normalized + PCA, (3) normalized + FLD
  - Data was normalized by dividing by 255 (the max pixel value) and had **784 features**
  - PCA allowed for 0.1 error, reduced dimensions to **85 features**
  - FLD reduced dimensions to **9 features**
- Case I, II, and III Discriminant Functions and kNN
  - Accuracy was evaluated on the testing set
  - 10-Fold cross validation
  - kNN was tested with different Minkowski distances (p = 1, 2, 3, and inf)
  - kNN was tested with different k's (k = 5, 10, 20, 50, 100, 250)
- Clustering - k-means and winner-takes-all
  - The algorithms separated the testing set into 10 clusters
  - After clustering, the most common label in each cluster was assigned to the entire cluster
  - Accuracy was then evaluated

Experimental Design - Neural Networks

- BPNN and CNN (convolutional neural networks) were both implemented in Keras using a Tensorflow GPU backend.
- Early stopping was used with a validation split of 10% of the training data.
- BPNN Testing was conducted on normalized data, PCA, and FLD.
  - 3 layers were used. 5, 8, 10, and 15 hidden nodes were evaluated.
  - 10-fold cross validation also used for normalized data
- CNN was evaluated on normalized data.
  - Uses two convolutional layers, a max pooling layer, and two dense layers.
  - First conv. layer uses 32 filters then second uses 64 before max pooling.
  - Dropout is 25% after the max pooling and 50% after first dense layer.



CNN Visualization from Wikipedia
https://upload.wikimedia.org/wikipedia/commons/6/63/Typical_cnn.png

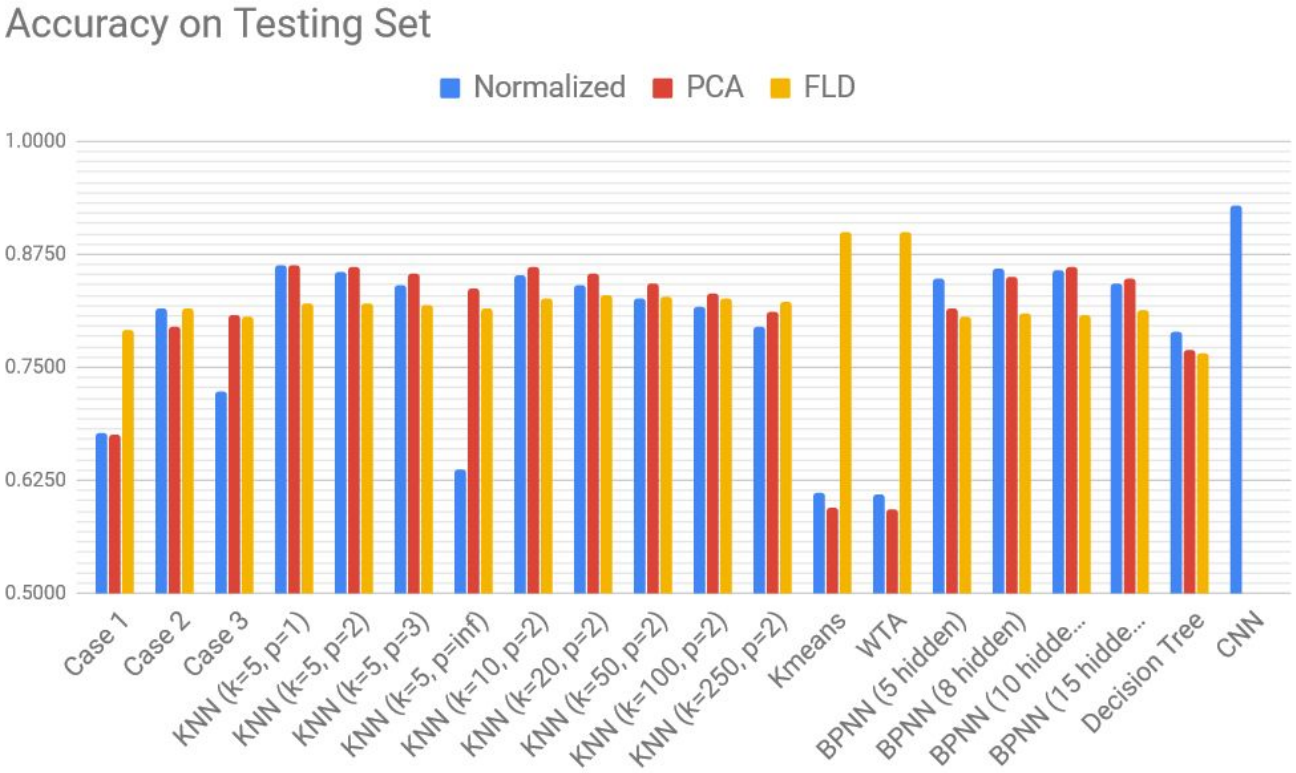
Testing Accuracies and 10-Fold Cross Validation

Accuracy							
				10-Fold			
Classifier	Norm	PCA	FLD	Norm	PCA	FLD	Notes
Case 1	0.6768	0.6759	0.7906	0.6857	0.6851	0.804	
Case 2	0.815	0.7951	0.8151	0.8242	0.8034	0.8324	
Case 3	0.7242	0.8072	0.8064	0.7279	0.8084	0.8173	
kNN 5, 1	0.8623	0.8636	0.8203	0.8632	0.867	0.8335	k=5, p=1 (Manhattan)
kNN 5, 2	0.8554	0.8603	0.8203	0.8569	0.8633	0.8334	k=5, p=2 (Euclidean)
kNN 5, 3	0.8402	0.8544	0.8182	0.8436	0.859	0.832	k=5, p=3
kNN 5, ∞	0.6366	0.8365	0.8144	0.6406	0.8434	0.8271	k=5, p=∞
kNN 10, 2	0.8515	0.8619	0.8264	0.8554	0.8643	0.8404	k=10, p=2
kNN 20, 2	0.8415	0.8541	0.829	0.8459	0.8579	0.8425	k=20, p=2
kNN 50, 2	0.8262	0.843	0.8288	0.8315	0.8465	0.8427	k=50, p=2
kNN 100, 2	0.8164	0.8314	0.8263	0.8184	0.8346	0.8401	k=100, p=2
kNN 250, 2	0.7949	0.8121	0.8221	0.7986	0.8157	0.8377	k=250, p=2
k-means	0.6115	0.5943	0.8996				10 clusters
WTA	0.6086	0.5937	0.8997				10 clusters
BPNN 5	0.848	0.8155	0.8065	0.8492			5 hidden nodes
BPNN 8	0.8593	0.8498	0.8092	0.852			8 hidden nodes
BPNN 10	0.8573	0.8602	0.8079	0.8502			10 hidden nodes
BPNN 15	0.8435	0.8474	0.8129	0.8543			15 hidden nodes
Decision Tree	0.7887	0.7694	0.7663	0.7952			from sklearn
CNN	0.9287						based on Keras

Norm - 784 features
PCA - 85 features
FLD - 9 features

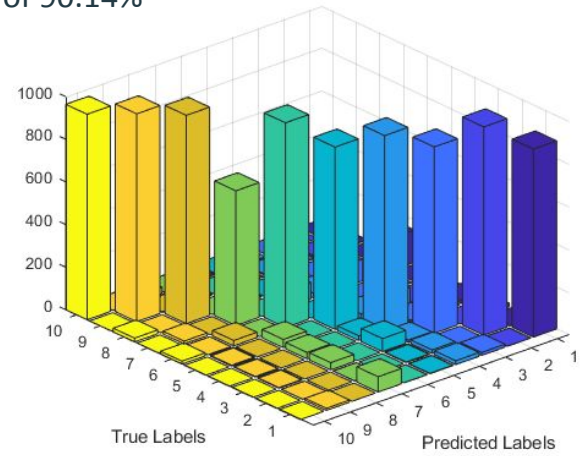
PCA reduced the number of dimensions to **85**, allowing for a maximum error of 0.1

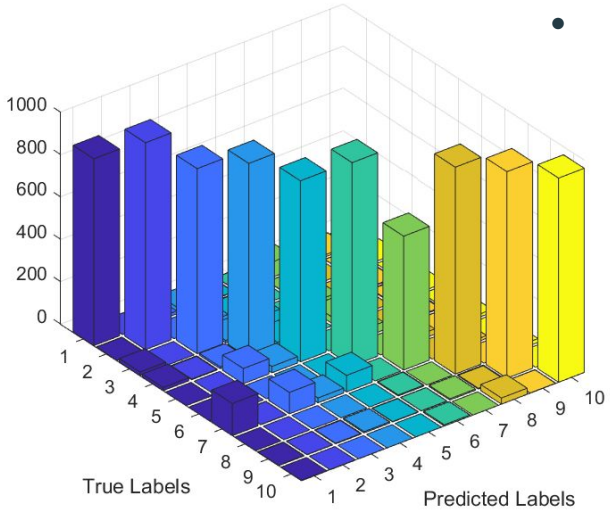
Testing Accuracies



Classifier Fusion

- Classifier fusion using 3 classifiers
  - Using Naive Bayes Approach
  - Convolutional Neural Network (CNN)
  - Backpropagation Neural Network (BPNN) with 8 hidden nodes (using normalized)
  - k-Nearest Neighbors (KNN) with k=5 and Euclidean distance with PCA
- Accuracy of 90.14%

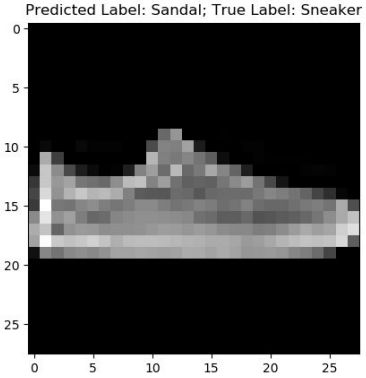


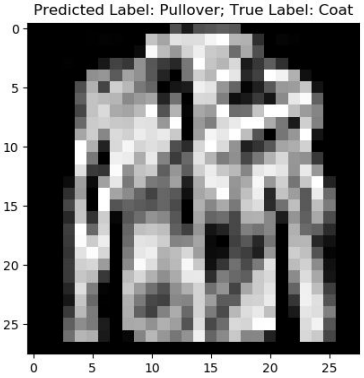


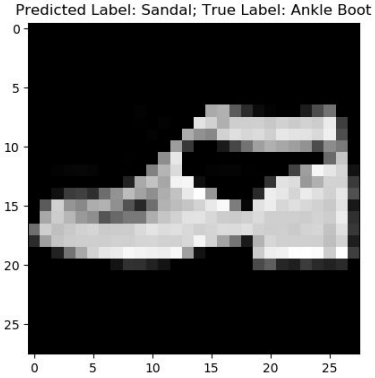
Front and back of confusion matrix

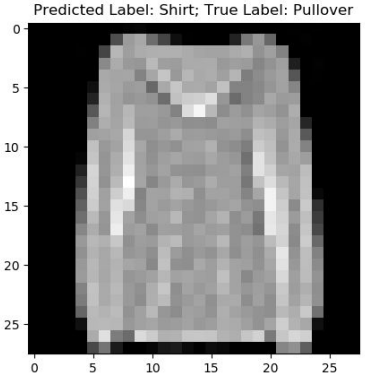
- The categories were as follows
  - 1 - T-shirt/Top
  - 2 - Trouser
  - 3 - Pullover
  - 4 - Dress
  - 5 - Coat
  - 6 - Sandal
  - 7 - Shirt
  - 8 - Sneaker
  - 9 - Bag
  - 10 - Ankle Boot

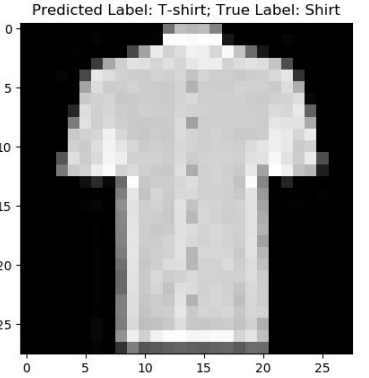
Examples of Misclassified Garments

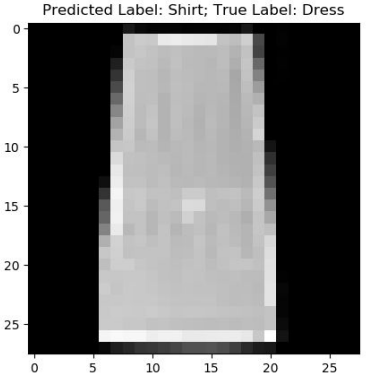












Performance Comparison

State of the Art using Wide Residual Networks	96.7%
CNN with Normalized	92.9%
Winner-Takes-All with FLD	90.0%
kNN with PCA using k=5 and city block distance	86.4%
BPNN with PCA using 10 hidden nodes	86.0%
Case II with FLD	81.5%