ECEN 758 Course Group Project

Xin Tong
ECEN 758
Texas A&M
College Station, USA
xintong@tamu.edu

Dharmenkumar Patel
ECEN 758
Texas A&M
College Station, USA
dppatel1@tamu.edu

Griffin Wagenknecht
ECEN 758
Texas A&M
College Station, USA
gwagenknecht@tamu.edu

***Abstract*— Developing a method to classify different articles of clothing based on imaging can have a profound impact on the retail industry. Through the mastery of predicting what category different pieces of clothing fall into, organizations can take business model approaches to maximizing revenue for different types of clothing during different times of the year. Furthermore, optimization of how much inventory to have on hand and when to keep up with the inconsistent rise and fall of supply and demand are pivotal decisions that need to be made correctly. In this report, a large-scale dataset consisting of greater than 60,000 28 x 28 grayscale images is utilized. The images are linked to one of ten classes, which are T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle Boot. Overall, this Fashion-MNIST dataset from [1] will be used in the development and comparison of multiple machine learning models in Python, including but not limited to density-based supervised learning, ensemble learning, and neural networks. Through multiple experiments, it was determined that the convolutional neural network was the most successful due to higher accuracy.**

I. INTRODUCTION

The dataset [1] gives a well-defined high-dimensional geometric dataset in high-dimension space. The sophisticated geometric structure in fact presents essential challenging factors to the literature. The structure of the dataset is complicated therefore it is significant to have some deep feeling about the structure through more concrete experimental consideration. We study this dataset as an assignment project from TAMU ECEN 758. To give some elaboration on the literature on this, we refer to [16] for all the relevant papers. More relevant considerations to our experiment can be referred to [17], [18], [19], [20], [21], [22], [27], [28], [29] such as the gradient boosting strategy and KNN classification.

Dataset Summary

Table I Descriptive Statistics

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7
count	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000
mean	4.500000	0.000900	0.006150	0.035333	0.101933	0.247967	0.411467	0.805767
std	2.872305	0.094689	0.271011	1.222324	2.452871	4.306912	5.836188	8.215169
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	4.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	7.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	9.000000	16.000000	36.000000	226.000000	164.000000	227.000000	230.000000	224.000000

pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783	pixel784
60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.000000	60000.00000
23.300683	16.588267	17.869433	22.814817	17.911483	8.520633	2.753300	0.855517	0.07025
48.854427	41.979611	43.966032	51.830477	45.149388	29.614859	17.397652	9.356960	2.12587
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
9.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	170.00000

II. METHOD

We apply K-means [2], [3], [4], histogram gradient boosting classification [5], [6], [7], [8], [9], [11], [12], [13], [14], [15], [23], [24], [25], [26], KNN classification [10] to the dataset in our experiment, and a neural network model [27], [28], [29]. The methods are actually quite general in their own direction and differentiation. Our understanding of these algorithms is based on their original function and original implementation. We took the following steps:

Step 1: **Data Importation.** We use the corresponding keras library which contains the corresponding datasets module where we can find the corresponding dataset that we imported into the Jupyter notebooks.

Step 2: **Data Visualization.** The dataset is already split into training, validation, and testing subsets with proportion 48000:12000:10000. We use the pyplot from the matplotlib library to realize the corresponding visualization of the dataset. We enumerate certain items from the whole dataset by writing a class of handlers:

```
plt.figure(figsize=(10, 10))
for i in range(30):
    plt.subplot(10, 10, i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap='gray')
    plt.xlabel(class_names[train_labels[i]])
plt.show()

# Reshape the images
x_train = train_images.reshape((60000, 28, 28, 1))
x_test = test_images.reshape((10000, 28, 28, 1))
```

Step 3. **Data Cleansing and Normalization**. We needed the data to be within the range of 0 to 1. To achieve this we divided the data set by 255 to ensure all points are less than 1 and greater than 0.

Step 4. **Methodology and Algorithms.** We attempted four different methods to achieve our goals which are listed below in section III.

III. EXPERIMENTS

Experiment 1: The first method we tried was to use the corresponding K-means unsupervised learning algorithm implemented in Python. The idea was to apply the clusterization algorithm to the original flattened image datasets. The idea is to find clusterization behaviors hidden in the high-dimensional data. However, the method seems to be sub-optimal with our dataset. This did shed some light on the corresponding topological and geometrical structures of the data when looking at the corresponding higher dimensional space where the data is distributed.

Experiment 2: Our second attempt was with KNN, which is the corresponding supervised classification based on nearest neighbor consideration. Since we realized the clusterization might not give us the best ideal classification result, we considered more classification-relevant considerations. We have the following:

TABLE II

KNN Results

	Precision	***Recall***	***F1-score***	***Support***
0	0.85	0.77	0.81	1109
1	0.97	0.99	0.98	981
2	0.82	0.73	0.77	1123
3	0.86	0.90	0.88	952
4	0.77	0.79	0.78	981
5	0.82	0.99	0.90	828
6	0.57	0.66	0.61	874
7	0.96	0.88	0.92	1094
8	0.95	0.97	0.96	978
9	0.97	0.90	0.93	1080
accuracy			0.86	10000
macro avg	0.86	0.86	0.85	10000
weighted avg	0.86	0.86	0.86	10000

Experiment 3: We then tried HGBC, which is the corresponding supervised classification based on gradient boosting. This includes certain stacking and boosting weak learning steps. The HGBC package is highly encapsulated and implemented by combining the boosting strategies, which we use directly. We have the following:

TABLE III

HGBC Results

	precision	*recall*	*f1-score*	*support*
0	0.86	0.83	0.85	1033
1	0.97	1.00	0.98	977
2	0.82	0.79	0.81	1042
3	0.90	0.90	0.90	998
4	0.83	0.80	0.82	1030
5	0.97	0.99	0.98	982
6	0.66	0.72	0.69	916
7	0.97	0.94	0.96	1031
8	0.97	0.98	0.98	998
9	0.96	0.97	0.96	993
accuracy			0.89	10000
macro avg	0.89	0.89	0.89	10000
weighted avg	0.89	0.89	0.89	10000

Experiment 4: Lastly we built a convolutional neural network. We chose this since it is often used for image classification. When building the model we built it around several layers that first extract the features from the image in the convolution layer. Then we reduced the size of the image in the pooling layer. In the flattening layer, we reduced our array to a one-dimensional array. In our dropout layer, it is fully connected to the next layer with 128 neurons in the first dense layer and 10 neurons in the second layer. Finally, we have a dropout layer to help prevent us from overfitting the model. We ran it with 15 epochs, the number of times the dataset is sent through the network. After running it those 15 times we got a training loss of .1188 and a training accuracy of .9602. When we trained it 20% of the training set was validation. When we tested it we got a test loss of .2719 and a test accuracy of .9143.

FIGURE I

CNN Train vs Test

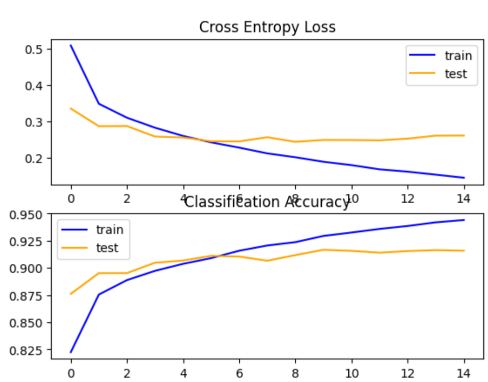

TABLE IV

CNN Results

Epoch	***Loss***	***Accuracy***	***Validation Loss***	***Validation Accuracy***
1	0.5091	0.8224	0.3360	0.8761
2	0.3490	0.8754	0.2873	0.8952
3	0.3107	0.8887	0.2878	0.8952
4	0.2832	0.8973	0.2589	0.9047
5	0.2606	0.9037	0.2562	0.9068
6	0.2429	0.9090	0.2460	0.9111
7	0.2284	0.9158	0.2456	0.9104
8	0.2126	0.9206	0.2570	0.9066
9	0.2020	0.9237	0.2444	0.9118
10	0.1896	0.9294	0.2496	0.9167
11	0.1804	0.9325	0.2493	0.9157
12	0.1688	0.9359	0.2484	0.9140
13	0.1624	0.9386	0.2532	0.9155
14	0.1542	0.9419	0.2614	0.9163
15	0.1457	0.9441	0.2616	0.9158

IV. CONCLUSION

Through experimentation, we come to the conclusion that the geometric and topological behavior of the dataset is not satisfactory in the situation of K-means clusterization, and its performance presents certain essential difficulties. This may be a very serious problem in higher dimensions since the clusterization internal compactness cannot be optimized in some nontrivial sense. The KNN works well, however, the HGBC approach is more powerful but not the most as we can see through the experiment results above. This approach combines the corresponding weak learners together to produce a certain ideal result. The neural network model is the most powerful and accurate through the layers of nodes and the predictions that are made.

Overall, the neural network is our best learning model with the highest average accuracy and smallest loss in all the classes. Furthermore, a clothing company looking to boost output and minimize cost would be able to extend our learning models in predicting what price range needs to be associated with what class and when these specific products associated with these classes need to be focused on during different seasons. One important thing to remember is the dataset has inconsistencies and could require transformations and imputations that vary depending on the end goal of the company. You can see our output below with the blue lettering being correctly identified and red being incorrectly identified, 23/25 were identified correctly.

FIGURE II

Results

V. REFERENCES

1. Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." *arXiv preprint arXiv:1708.07747* (2017).

2. Steinhaus, Hugo. "Sur la division des corps matériels en parties." *Bull. Acad. Polon. Sci* 1, no. 804 (1956): 801.

3. MacQueen, James. "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth*

20. Dong, Manqing, Lina Yao, Xianzhi Wang, Boualem Benatallah, and Shuai Zhang. "GrCAN: gradient boost convolutional autoencoder with neural decision forest." arXiv preprint arXiv:1806.08079 (2018).

21. A. D. Bhat, H. R. Acharya and S. H.R., "A Novel Solution to the Curse of Dimensionality in Using KNNs for Image Classification," 2019 2nd International Conference on Intelligent Autonomous

Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14, pp. 281-297. 1967.

4. Lloyd, S. P. "Least square quantization in PCM. Bell Telephone Laboratories Paper. Published in journal much later: Lloyd, SP: Least squares quantization in PCM." *IEEE Trans. Inform. Theor.(1957/1982)* 18, no. 11 (1957).

5. Breiman, Leo. "Arcing the Edge Technical Report 486." *Statistics Department, University of California, Berkeley*(1997).

6. Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.

7. Friedman, Jerome H. "Stochastic gradient boosting." *Computational statistics & data analysis* 38, no. 4 (2002): 367-378.

8. Mason, Llew, Jonathan Baxter, Peter Bartlett, and Marcus Frean. "Boosting algorithms as gradient descent." *Advances in neural information processing systems* 12 (1999).

9. Ll, Mason, and J. Baxter. "Boosting algorithms as gradient descent in function space." *NIPS: New Orleans, LA, USA*(1999).

10. Fix, Evelyn, and Joseph Lawson Hodges. "Discriminatory analysis. Nonparametric discrimination: Consistency properties." *International Statistical Review/Revue Internationale de Statistique* 57, no. 3 (1989): 238-247.

11. Kearns, Michael. "Thoughts on hypothesis boosting, 1988." *ML class project* 319 (2020): 320.

12. Kearns, Michael, and Leslie Valiant. "Cryptographic limitations on learning boolean formulae and finite automata." *Journal of the ACM (JACM)* 41, no. 1 (1994): 67-95.

13. Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).

14. Meng, Qi, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, and Tie-Yan Liu. "A communication-efficient parallel algorithm for decision tree." Advances in Neural Information Processing Systems 29 (2016).

15. https://github.com/Microsoft/LightGBM.

16.https://docs.google.com/spreadsheets/d/1cGX7Juedn_KVUgjDk298v5uUjc_wPk930tKyEoZhTQM/edit#gid=785895781.

17. Datsi, Toufik, Khalid Aznag, and Ahmed El Oirrak. "Fashion Image Classification Using Convolutional Neural Network-VGG16 and eXtreme Gradient Boosting Classifier." In *International Conference on Advanced Intelligent Systems for Sustainable Development, pp. 419-431. Cham: Springer Nature Switzerland, 2022.*

18. Samia, Bougareche, Zehani Soraya, and Mimi Malika. "Fashion Images Classification using Machine Learning, Deep Learning and Transfer Learning Models." In 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), pp. 1-5. IEEE, 2022.

19. Yepdjio Nkouanga, Hermann, and Szilárd Vajda. "Optimization Strategies for the k-Nearest Neighbor Classifier." SN Computer Science 4, no. 1 (2022): 47.

22. Mukherjee, Sudipto, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. "Clustergan: Latent space clustering in generative adversarial networks." In Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, pp. 4610-4617. 2019.

23. Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. "Classification and regression trees (Wadsworth, Belmont, CA)." ISBN-13 (1984): 978-0412048418. (on decision tree)
24. Loh, Wei‐Yin. "Fifty years of classification and regression trees." International Statistical Review 82, no. 3 (2014): 329-348. (on decision tree)
25. Fisher, Ronald A. "The use of multiple measurements in taxonomic problems." Annals of eugenics 7, no. 2 (1936): 179-188. (on decision tree)
26. Morgan, James N., and John A. Sonquist. "Problems in the analysis of survey data, and a proposal." Journal of the American statistical association 58, no. 302 (1963): 415-434. (on decision tree)
27. Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
28. Simard, Patrice & Steinkraus, Dave & Platt, John. (2003). Best Practices for Convolutional Neural Networks.
29. W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," in Neural Computation, vol. 29, no. 9, pp. 2352-2449, Sept. 2017, doi: 10.1162/neco_a_00990.