FashionMNIST classification using SVM & CNN

1st Ruthvik Kanumuri
Data Science Department
Texas A&M University
College Station, United States
kruthvik007@tamu.edu

2nd Sri Akash Balu
Data Science Department
Texas A&M University
College Station, United States
sxb3657@tamu.edu

3rd Jnana Preeti Parlapalli
Data Science Department
Texas A&M University
College Station, United States
pj.preeti@tamu.edu

***Abstract*—This study explores the classification on the Fashion-MNIST dataset, employing machine learning algorithms (SVM and RF) and a deep learning algorithm, CNN . CNN extracts intricate features, SVM recognizes patterns using kernel functions, and Random Forest Classifier combines decision trees for robust classification. A comparative analysis of accuracy and efficiency sheds light on the strengths and limitations of these approaches for fashion image recognition.**

***Index Terms*—**

I. Introduction

The classification of fashion-related images is vital in computer vision, impacting e-commerce, trend analysis, and retail automation (Liu et al., 2016; Taha et al., 2020; Chen et al., 2021). This project focuses on classifying the FashionMNIST dataset, a widely used benchmark (Xiao et al., 2017), using Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs). CNNs, powerful for image classification, extract patterns through convolutional layers, capturing local features and extracting global features for a nuanced understanding (Szegedy et al., 2015; He et al., 2016). SVMs, a traditional yet effective algorithm, excel in high-dimensional spaces, leveraging kernel functions for complex decision boundaries (Vapnik, 1995; Cortes and Vapnik, 1995). This project leverages SVMs to discern patterns in the FashionMNIST dataset, focusing on the high-dimensional feature space. By implementing these methodologies separately, the project aims for a comprehensive comparative analysis, providing insights into their classification accuracy and computational efficiency. This understanding benefits computer vision and has practical implications for industries relying on automated image recognition, enhancing user experience and business operations.

II. Method

A. Model Justification

The hierarchical feature learning of CNNs allows for the extraction of complex patterns, textures, and shapes within the images. This results in a more robust and nuanced understanding of fashion items, potentially improving the precision of product recommendations or categorizations.

SVMs are capable of capturing non-linear relationships in the data, allowing businesses to identify subtle distinctions between fashion categories. This can be valuable in scenarios where clear decision boundaries are crucial for accurate classification.

B. Dataset Description

Fashion MNIST is a benchmark grayscale image dataset, serving as a substitute for the traditional MNIST. Comprising 60,000 training and 10,000 testing images, it showcases 10 fashion categories, including clothing items like shirts, trousers, and shoes. Each image is a 28x28 pixel grayscale representation, offering a diverse and challenging dataset for image classification tasks. Fashion MNIST serves as an essential resource for evaluating and benchmarking machine learning models, particularly in the realm of computer vision - fostering advancements in fashion-related image analysis and classification techniques.

Fig. 1. Class Labels of the dataset

C. Classifier Implementation

The Support Vector Classifier (SVC) was selected for its effectiveness with high-dimensional data and its ability to model complex, non-linear patterns. Utilizing scikit-learn's dependable implementation, the SVC was fine-tuned to enhance its performance. Key hyperparameters, such as the regularization parameter C, kernel type, gamma, and polynomial degree, were crucial in this optimization process. The final model, using scikit-learn's default SVC settings, was trained on the entire Fashion MNIST dataset. This process included converting raw pixel data from Fashion MNIST images into features compatible with the selected kernel. In the Fig. 2. below, we show the SVCs that define the decision boundaries between different clothing categories. By identifying the closest data points to these boundaries, SVM creates a margin that optimally separates various fashion items, ensuring accurate classification of each image.

The Convolutional Neural Network (CNN) was chosen for its effectiveness in handling high-dimensional image data and its capability to capture intricate, non-linear patterns. Leveraging the robust implementation provided by popular frameworks like TensorFlow or PyTorch, the CNN underwent a fine-tuning process to enhance its performance on the Fashion MNIST dataset. Critical hyperparameters, including but not limited to, the architecture's depth, filter sizes, activation functions, and pooling strategies, played a pivotal role in optimizing the CNN. Additionally, the learning rate, batch size, and dropout rates were carefully adjusted to achieve the best trade-off between model generalization and overfitting. The final CNN model was trained on the complete Fashion MNIST dataset, encompassing the entire range of fashion-related images. This training process involved preprocessing the raw pixel data from Fashion MNIST images, transforming them into a format compatible with the CNN architecture. The model was designed to automatically learn hierarchical features from the input images, enabling it to discern complex patterns and representations within the fashion dataset.

```
Support Vectors: [[-4.97814649 -3.77356513 -2.09364714 ...  0.04613227  0.00782572
  -0.0979526 ]
 [-3.30124587 -3.11313135 -0.83853322 ... -0.00759114  0.45703169
  -0.1944823 ]
 [-1.79087591 -3.05330968 -1.00567215 ... -0.33444499 -0.03911446
  -0.41102814]
 ...
 [-3.94151892  3.79204135  3.09013179 ... -0.72704637  0.02501902
   0.25657042]
 [-5.12679347 -0.30437448 -0.86582919 ...  0.43959795  0.21697192
  -0.26330613]
 [-5.56727085 -0.63183974  0.03990639 ...  0.35694624 -0.03508344
   0.20703217]]

 Number of Support Vectors for each class: [2250  389 2981 1567 2836  909 4114 1118  678  740]

 Coefficients of the support vector machine: [[ 0.00621013  0.55645848  0.51359597 ...  0.21084004 -1.10668438
  -0.02511182]
 [-0.04943992 -0.34917412  0.59399959 ... -0.18575376 -0.08965734
   0.17394131]
 [ 0.24276859  0.20562043  0.06744825 ... -0.42824548 -1.00516417
  -0.62021993]
 ...
 [-1.22481624  0.31947343  0.7452689  ... -1.0871329   0.01892447
   0.34724813]
 [ 0.17683747 -0.99789389 -0.15440103 ... -1.48942113  0.78387936
   1.4411207 ]
 [ 0.7940565  -0.29676165 -0.90843534 ... -0.64871898 -0.28580402
   1.04647035]]

 Intercept: [ 1.50996448 -0.64033644 -0.42350788 -0.3482135   3.96058274 -0.74212506
  4.29430008 -0.54340151  2.29768713 -3.416713   -2.10205325 -4.3287451
  1.91432792 -2.81280854  1.84367417 -1.42503491  0.71292805  1.06022728
  0.94173658  3.38688744 -0.34184156  3.06370106  0.44221864  1.71658124
 -0.4321051   2.58940964 -1.05068256  2.50083715 -0.15647432  1.60866063
  3.69686165 -0.91721201  3.27725938  0.07860391  1.52917679 -5.01165923
  1.04067325 -5.8068164   2.96573391  4.14146016  0.27270211  2.68419293
 -5.48404202  2.18156483  3.29416969]
```

Fig. 2. Support Vectors for each class

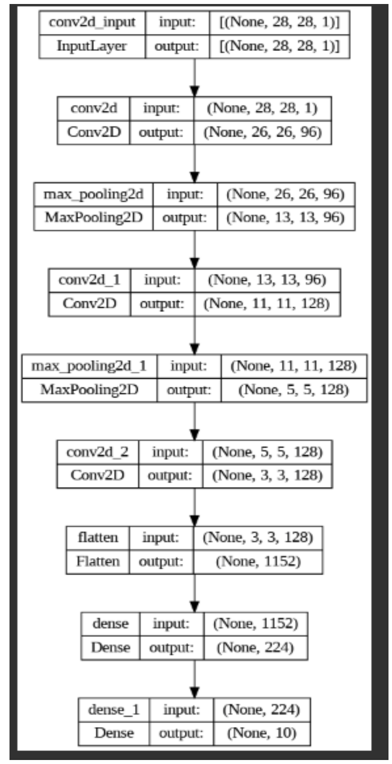


Fig. 3. CNN Architecture

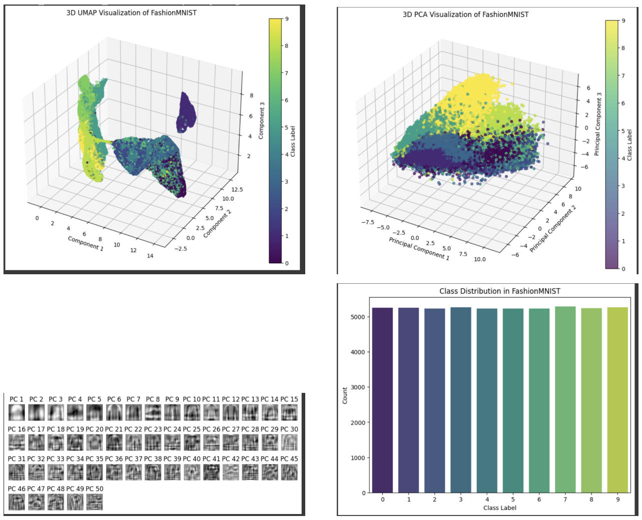
III. EXPERIMENTS

A. Data Preparation

In the Fashion MNIST data preparation, TensorFlow imports and meticulously divides the dataset into 70:10:20 ratios for training, validation, and test sets. To ensure feature uniformity, a normalization process addresses pixel value variations. Principal Component Analysis (PCA) strategically reduces dimensionality, streamlining the dataset and expediting training. This accelerates training while retaining crucial data variations. PCA benefits traditional classifiers like Support Vector Machines (SVMs) and enhances Convolutional Neural Network (CNN) efficiency by capturing essential features and minimizing redundancy. Additionally, images are reshaped from 28x28 to a flattened 1x784 configuration, enabling seamless integration into various classification algorithms. These steps optimize the preprocessed dataset for training and evaluation across diverse machine learning models, ensuring efficient performance.

B. Exploratory Data Analysis

In our research, multiple data visualization techniques were employed. Initially, we analyzed the distribution of different classes within the dataset. Subsequently, we incorporated Principal Component Analysis (PCA) to project the data into a lower-dimensional space, including both 3D visualizations of PCA and the examination of PCA components. Additionally, we conducted 3D UMAP (Uniform Manifold Approximation and Projection) visualization. The ensuing figures demonstrate these visualization methodologies.



C. Models

- SVM: Support Vector Machines (SVM) are used for classifying Fashion MNIST dataset, a collection of clothing images, by distinguishing between different fashion categories. In this process, each image, representing an item like a shirt or a shoe, is transformed into a set of features. These features typically derive from the pixel values of the image, capturing the unique aspects of each fashion item. The SVM operates by finding the optimal hyperplane which separates the different categories in the feature space. This separation is crucial in classifying the images into their respective categories. The algorithm considers not just the closest points (support vectors) to the decision boundary but also maximizes the margin between these points and the boundary, enhancing classification robustness. Tuning hyperparameters like the regularization parameter (C), kernel type (linear, polynomial, etc.), and others, is a significant step to ensure the SVM effectively handles the non-linear and high-dimensional nature of image data. By adjusting these parameters, the SVM can more accurately classify the diverse and complex patterns present in the Fashion MNIST dataset.

- CNN: Initially, for the CNN model, we reshaped the images to include a channel dimension. The CNN architecture is then defined using the TensorFlow Keras library. The model comprises three 2D convolution layers with increasing filter sizes to learn spatial hierarchies of features within input data, each succeeded by a 2D max-pooling layer for downsampling the spatial dimensions of feature maps, the model extracts hierarchical features from input images of clothes. After the convolutional layers, a flatten operation is applied to prepare the data for fully connected layers. Two dense (fully connected) layers follow, with the final layer having 10 neurons corresponding to the number of classes in the dataset. ReLU activation functions are used for the convolutional and dense layers to learn hierarchical features from input images, except for the output layer where softmax activation is applied for multi-class classification. The model is compiled using the Adam optimizer and sparse categorical cross entropy loss. Finally, the model is trained for 10 epochs on the training data, with validation performed on the test data. The accuracy and loss metrics are monitored during training. Additionally, the model's performance is evaluated using metrics such as classification reports and confusion matrices.

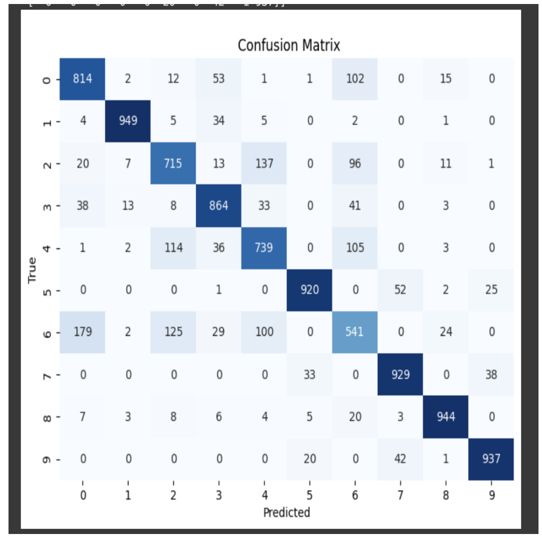


Fig. 4. Confusion Matrix for SVM

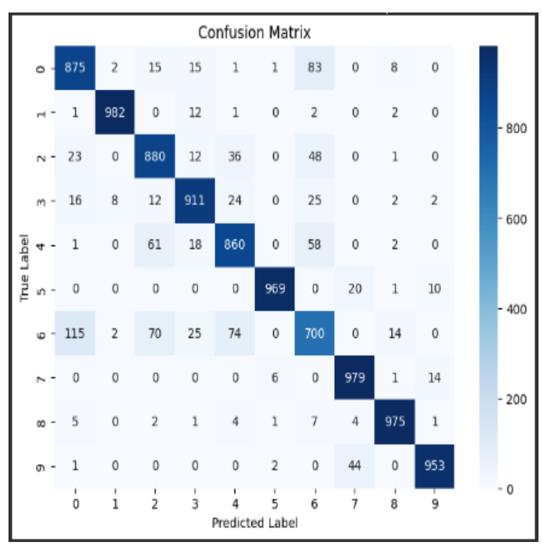


Fig. 5. Confusion Matrix for CNN

IV. Conclusion

CNNs are highly effective for image classification tasks, making them ideal for the Fashion MNIST dataset, which comprises grayscale fashion images. Businesses can leverage CNNs to accurately classify and identify various clothing items, enhancing customer experience in e-commerce platforms or inventory management systems. SVMs excel in handling high-dimensional data and are particularly effective in scenarios with a clear margin of separation between classes. In the context of Fashion MNIST, SVMs can provide accurate classification for different types of clothing, contributing to inventory management, trend analysis, and targeted marketing efforts.

After implementing the two models, a machine learning model (SVM) and a deep learning model (CNN) on the FashionMNIST dataset, we conclude that the CNN outperformed as a classifier as compared to SVM because of its ability capture spatial relationships between pixels which guide in recognizing patterns and objects in images. The accuracy of the CNN model after hyperparameter tuning is 0.9732 (on implementing for 5 Epochs). Even for 1 Epoch, accuracy is 0.9061 for CNN, which is very high. The accuracy of the SVM model is 0.8346. Hence, with these high accuracy values of implementing the CNN model, we can conclude that it is the best classifier on the FashionMNIST dataset.

```
Accuracy: 0.8358
Classification Report:            precision    recall  f1-score   support

           0       0.77      0.81      0.79      1000
           1       0.97      0.95      0.96      1000
           2       0.73      0.71      0.72      1000
           3       0.83      0.87      0.85      1000
           4       0.73      0.75      0.74      1000
           5       0.94      0.92      0.93      1000
           6       0.60      0.54      0.57      1000
           7       0.90      0.93      0.91      1000
           8       0.94      0.94      0.94      1000
           9       0.94      0.94      0.94      1000

    accuracy                           0.84     10000
   macro avg       0.83      0.84      0.83     10000
weighted avg       0.83      0.84      0.83     10000
```

Fig. 6. Accuracy for SVM

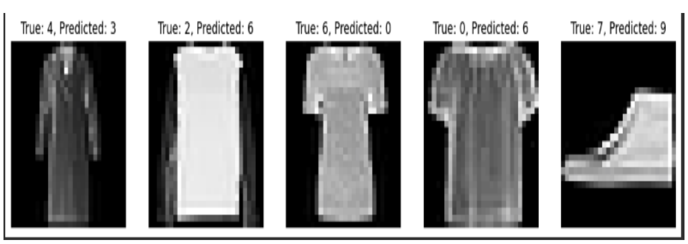


Fig. 7. Incorrect Predictions

```
313/313 [==============================] - 15s 49ms/step - loss: 0.3740 - accuracy: 0.9084
Test accuracy: 0.9083999991416931
313/313 [==============================] - 10s 32ms/step
              precision    recall  f1-score   support

           0       0.84      0.88      0.86      1000
           1       0.99      0.98      0.98      1000
           2       0.85      0.88      0.86      1000
           3       0.92      0.91      0.91      1000
           4       0.86      0.86      0.86      1000
           5       0.99      0.97      0.98      1000
           6       0.76      0.70      0.73      1000
           7       0.94      0.98      0.96      1000
           8       0.97      0.97      0.97      1000
           9       0.97      0.95      0.96      1000

    accuracy                           0.91     10000
   macro avg       0.91      0.91      0.91     10000
weighted avg       0.91      0.91      0.91     10000
```

Fig. 8. Accuracy for CNN

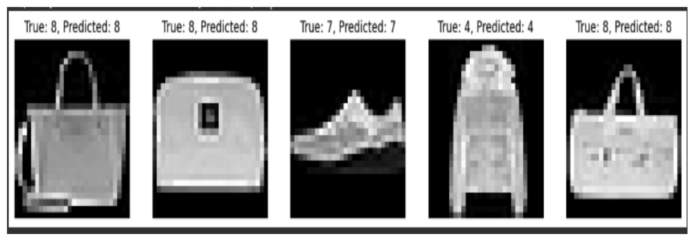


Fig. 9. Correct Predictions

References

[1] Chen, H., Zhang, Y., & Sun, C. (2021). Fashion image classification based on improved deep convolutional neural network. IEEE Access, 9, 147759-147771.

[2] Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. arXiv preprint arXiv:1512.04328.

[4] Liu, S., Xu, J., & Chang, S. (2016). Fashion landmark detection and attribute recognition with deep learning. arXiv preprint arXiv:1610.08656.

[5] Szegedy, C., Liu, W., Yang, Y., & Chen, P. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[6] Taha, I., El-Kadmy, I. M., & El-Sappagh, S. M. (2020). Multi-label fashion image classification using convolutional neural networks. International Journal of Electrical and Computer Engineering, 10(1), 28-32.

[7] Vapnik, V. N. (1995). The nature of statistical learning theory. Springer Science & Business Media.

[8] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

https://strawhatsfashionmnistcnnsvm3.wordpress.com/