Image Classification for the FashionMNIST dataset

1st Vishnu Vamsi Bezawada
dept. Management Information Systems
Texas A&M University
College Station, USA
vishnuvamsibezawada@tamu.edu

2nd Pragya Mittal
dept. Computer Science and Engineering
Texas A&M University
College Station, USA
pragya10@tamu.edu

3rd Tanay Mahesh Mehendale
dept. Management Information Systems
Texas A&M University
College Station, USA
tanay.mehendale@tamu.edu

4th Akshat Pandey
dept. Electrical and Computer Engineering
Texas A&M University
College Station, USA
akshatp24@tamu.edu

5th Logan David Garcia
dept. Electrical and Computer Engineering
Texas A&M University
College Station, USA
logan7@tamu.edu

***Abstract*—This paper performs a comparative analysis of various image classification algorithms on the Fashion MNIST dataset, one of the benchmarking tasks in the fashion domain for image recognition. Logistic regression, decision trees, SVMs, and CNNs are evaluated using key metrics such as accuracy, precision, recall, and F1-score. Experiments show distinctive points of strength and deficiency among these classifiers, with CNNs giving the best overall performance due to their ability to model the spatial hierarchies in the image data. On the other hand, traditional methods like logistic regression and decision trees are competitive with a lower computational load, hence performance and efficiency trade-offs. Results obtained will be highly informative in the selection of suitable models for image classification tasks and thus have supporting roles for researchers and practitioners.**

I. Introduction

Image classification is a fundamental task in computer vision, which ranges from medical imaging to autonomous vehicles. Accurately categorizing images into one of the predefined classes is an essential skill in many modern systems. Various machine learning algorithms have been developed over the years that solve this problem. Image classification is a task that, though improved, has many obstacles that it needs to work its way around. First and foremost is the image quality and features that have great effects on classifiers' performances.

Traditional machine learning (ML) algorithms, such as logistic regression and decision trees, have been widely used for classification because of their simplicity and interpretability. Support Vector Machines (SVM) are traditionally useful for handling small datasets and high-dimensional feature spaces. On the other hand, Convolution Neural Networks (CNN) reflect state-of-the-art techniques in deep learning (DL) and are explicitly designed to extract spatial hierarchies from image data. Such a diverse set of models allows for evaluating traditional and modern approaches.

Performance metrics such as accuracy, precision, recall, and F1-score will evaluate the models by bringing about a multi-dimensional perspective, especially critical in those applications where the cost of a false positive and that of a false negative are different.

II. Literature Review

Many recent studies on fashion image classification have explored ML and DL techniques to improve model performance, address overfitting, and enhance classification performance. This section reviews key works in the fashion sector, highlighting the advancements and challenges in applying these techniques to fashion image datasets such as Fashion MNIST.

Ren et al. (2021) compared traditional ML models and CNNs for garment image classification, finding CNNs outperformed models like random forests and k-nearest neighbors (KNN), especially with complex data, highlighting their superior accuracy and robustness [1]. Sarowar et al. (2019) combined Histograms of Oriented Gradients (HOG) for feature extraction, Principal Component Analysis (PCA) for dimensionality reduction, and SVM for classification, achieving 94.02% accuracy and demonstrating the effectiveness of combining traditional and advanced techniques for improved classification [2].

Building on CNN architectures, Xhaferra et al. (2022) evaluated two CNN architectures, CNN-C1 and CNN-C2, for classifying Fashion MNIST images, finding CNN-C2, with batch normalization and dropout, outperformed CNN-C1 by reducing overfitting and improving generalization [3]. When comparing CNNs with traditional ML models, Rastogi et al. (2022) showed that deep neural networks (DNNs), particularly CNNs, consistently outperformed traditional models like SVM and KNN for tasks like digit classification, highlighting their superiority for complex image recognition [4]. Samia et al. (2022) used PCA to reduce dataset dimensionality and enhance the performance of ML, DL, and transfer learning models. Their approach improved traditional ML accuracy from 79.86% to 88.71%, showcasing PCA's importance in fashion image classification[6]. Finally, Ranzato and Zanella

(2019) introduced a robustness verification method for SVMs trained on datasets like Fashion MNIST [7].

Costa et. al.[8] mentions that Decision trees are widely recognized for their interpretability and transparency, making them crucial for decision-making in domains where understanding model logic is essential. According to Mahbooba et. al. [9] Unlike black-box models such as neural networks, decision trees rely on intuitive rules, with each node representing a feature and decision threshold, which helps foster trust in predictions. While Zihni et. al. [10] states that the more complex models may achieve higher accuracy, decision trees balance interpretability with competitive performance through effective feature selection and importance analysis. Metrics like the Gini index optimize feature splits, enhancing both precision and comprehensibility.

Our work evaluates the performance of logistic regression, KNNs, decision trees, support vector machines, and CNNs on the Fashion MNIST dataset. Using these metrics, we can answer crucial questions about the trade-offs of different models in terms of accuracy, computational complexity, and applicability to real-world problems. The results of the current research will serve to enrich the arsenal of recommendations for both researchers and practitioners seeking to choose relevant classification models with due regard to specific task needs and constraints.

III. Methodology

The dataset used in this project is the FashionMNIST dataset which represents different clothing items. With this approach, before passing the data to machine learning models, various preprocessing steps were applied. Initially, the data set was divided into training, validation, and test sets and features were standardized. At the same time, we performed PCA to reduce the dimensions while capturing 95% of data variance. It helps reduce computational complexity while retaining important data, centered towards traditional machine learning models. After preprocessing, several classical ML algorithms were applied to classify the FashionMNIST dataset. Each model was trained on the standardized and PCA-reduced data.

In addition to classical ML algorithms, a CNN was implemented using PyTorch to take advantage of deep learning techniques for image classification. The CNN architecture consisted of multiple convolutional layers followed by max-pooling layers and fully connected layers. The CNN models were trained for **50** epochs, using backpropagation with an **Adam** optimizer and a **cross-entropy** loss function. A learning rate of 0.001 and 0.002 were experimented with out of which α = **0.002** helped the network generalize better. Hyperparameters such as the number of neighbors for KNN and the learning rate for CNN were tuned to optimize performance. Metrics like Accuracy, precision, recall, F1-score, and confusion matrices on validation and test sets were used to evaluate the performance of the models.

IV. Experiments

A. Dataset

The Fashion-MNIST dataset is a collection of 70,000 grayscale images, each sized 28x28 pixels, representing various fashion items from Zalando. It consists of 60,000 training images and 10,000 test images, categorized into 10 classes: T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. Each image is labeled with an integer corresponding to one of these categories.

Fig. 1. FashionMNIST Data Sample

B. Data Preprocessing

The preprocessing steps for the FashionMNIST dataset involved preparing the data for use in both classical ML models and DL models. First, the dataset was extracted using the PyTorch library. The images are then converted into a 784-dimensional vector (28 x 28 pixels), that is flattened to easily process it into the models.

To ensure that the features were on a similar scale and to improve the performance of ML models, the pixel values were standardized using StandardScaler. Standardization scales the data so that its mean is 0 and its standard deviation is 1, which can be very important for algorithms that optimize by using a gradient based approach for faster convergence. This standardization process was applied to both the training and test datasets. Additionally, the training data was split into training and validation sets using a 90-10 ratio via the train_test_split function, ensuring a proper evaluation of the models' performance and initial test dataset is considered as a test set.

For dimensionality reduction, PCA was used to reduce the dataset's dimensionality while retaining 95% of its variance. This step helped decrease the computational complexity and mitigate overfitting by removing less relevant features. The

PCA transformation was performed after standardization, and the resulting components were used as input for various ML algorithms.

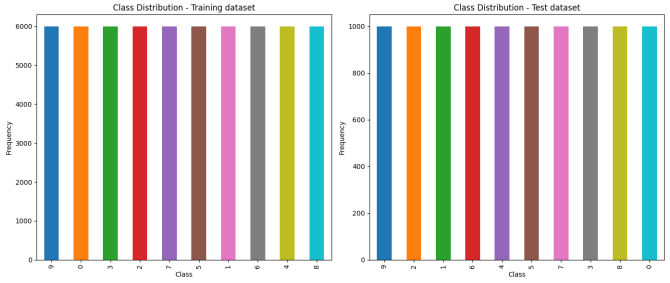

Fig. 2. Train and Test Images Sample Count

C. Machine Learning Models

1) K-Nearest Neighbors (KNN)

KNN is simple to implement, good at classifying similar data, and has been implemented on FashionMNIST before [1]. Ren, Fei, et al. found that a k value of 4 provided the best results for the KNN model, but we found that setting the hyperparameter to the default value of 5 provided comparable scores for the validation data. When the trained models were given the test data the model with the k set to 5 performed better than it did when k was set 4 on the standard data. Thus the k value of 5 was used for KNN when comparing it against the other models. The p value was left to its default value of 2 which uses the Euler distance between samples [1].

2) Logistic regression

Logistic regression is chosen for classification due to its simplicity as a ML method, its comprehensibility, and its exceptional training efficiency. Another benefit of logistic regression is that it does not require substantial processing resources for categorization. The LR model uses multinomial classification in order to handle 10 different classes at once, which is better than binary classification for the given case. The lbfgs (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) solver is good for this dataset because of its good handling of the input space of 784 pixels, alongside controlling the loss function. The regularization parameter (C=1.0) was the perfect trade-off between model complexity and generalization. This prevented overfitting while capturing important feature correlations in the image data. The maximum number of iterations (max_iter = 1000) ensures enough time for the model to converge but not at the expense of computational resources. This hyperparameter configuration allows the model to relate the pixel intensity values to the respective clothing categories efficiently.

3) Decision Tree

A Decision Tree is a versatile supervised learning algorithm that is used for both classification and regression tasks. It splits the data recursively based on feature thresholds to create a tree structure, where each internal node represents a decision based on a feature, and leaf nodes provide predictions. The splitting criterion determines how the decision tree divides data at each node, with common metrics including Gini impurity and Entropy (Information Gain). Gini impurity measures the likelihood of misclassifying a randomly chosen element, while Entropy quantifies information disorder or uncertainty to evaluate splits. Several parameters influence decision tree training, such as ‘max_depth‘, which limits the tree’s depth to prevent overfitting by controlling complexity and avoiding noise capture. The ‘criterion‘ parameter specifies the splitting metric, commonly ”gini” or ”entropy.” Additional parameters like ‘min_samples_split‘, defining the minimum samples required to split a node, and ‘min_samples_leaf‘, specifying the minimum samples in a leaf node, further refine the tree’s structure and improve generalization.

4) Linear Support Vector Machine (Linear SVM)

The Linear SVM is used when the classes can be separated by a straight line or hyperplane in higher dimensions. This model is particularly effective when the data is linearly separable or nearly so, as it tries to find the optimal hyperplane that maximizes the margin between different classes. The regularization parameter C is set to 1 to find the trade off between maximizing the margin and minimizing the classification error. By applying PCA or other feature reduction techniques, the data can often be projected into a lower-dimensional space where a linear decision boundary might be sufficient. The linear model is computationally more efficient and helps in understanding the relationships between features and classes.

5) Radical Basis Function (RBF) Support Vector Machine

The RBF kernel maps the input data into a higher-dimensional space even if the data is not linearly separable in the original feature space. The kernel coefficient gamma is used by default to prevent overfitting and to improve generalization. The RBF kernel is useful as it can create complex decision boundaries by mapping the data to a higher-dimensional space. It helps in capturing these non-linear relationships and creates more flexible decision boundaries, potentially improving performance when the images contain complex features.

D. Deep Learning Model

1) Convolution Neural Networks (CNN)

With approaches that rely on machine learning, the main aim is to extract features before we feed the images to the classifier. Deep Learning seemed like an intuitive approach to solving a problem like multi-class image classification for various reasons. A non-linear problem like the one we have with this project will be better solved using a combination of perception layers. A neural network would be able to learn the features on its own due to the vast amount of data we are using with fashion-MNIST. A convolutional neural network is suggested to be very good for interpreting hierarchical data at multiple levels which makes it a good pick for this kind of task. The standard MNIST dataset has state-of-the-art models as CNNs, so it seemed logical to attempt to extend a CNN

for the fashion MNIST dataset as well.

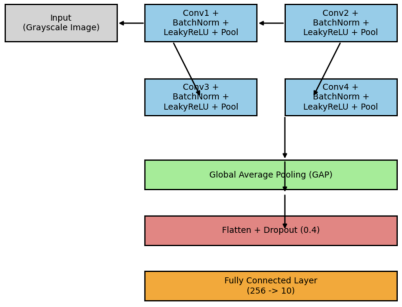

Fig. 3. Proposed CNN Architecture

The CNN architecture is designed to understand input data at multiple levels. Convolution slides a smaller matrix over a larger one to compute pairwise dot products and create a feature map, representing initial feature interpretations. A Leaky ReLU activation function is used after each convolution to address the vanishing gradient problem, improving validation accuracy slightly compared to ReLU. Batch normalization stabilizes activations, mitigating internal covariance shifts, speeding up training, and enabling higher learning rates. The Global Average Pooling (GAP) layer reduces model parameters and overfitting by replacing fully connected layers, which would otherwise have too many weights [11]. After flattening the feature maps, a dropout layer prevents overfitting by randomly disabling neurons, forcing the model to generalize better. Finally, the fully connected layer uses the feature vector to make predictions.

Layer	**Output Size**
Input	$(1, 1, 28, 28)$
Conv1	$(1, 32, 28, 28)$
Conv2	$(1, 64, 14, 14)$
Conv3	$(1, 128, 7, 7)$
Conv4	$(1, 256, 3, 3)$
Global Average Pooling (GAP)	$(1, 256, 1, 1)$
Flatten	$(1, 256)$
Fully Connected (FC)	$(1, 10)$

V. Results and Discussion

PCA data Analysis	**Metrics(%)**			
	Accuracy	**Precision**	**Recall**	**F1-Score**
KNN	85.35	85.66	85.35	85.36
Logistic	83.61	83.42	83.61	83.50
Decision Tree	80.26	80.66	80.26	80.31
Linear SVM	84.00	83.60	83.50	83.50
RBF SVM	88.00	88.10	88.20	88.20
CNN	91.98	91.99	91.98	91.96

TABLE I
Performance Metrics for Different Models - Scaled Data

The dataset was split by selecting 10% of the training data to serve as validation data, with the random state set to 42. This approach ensures to recreate of the same consistent splits of the training, validation, and test datasets during hyperparameter tuning for each model and model comparison. Subsequently, the data was scaled for use in training and testing across all models. PCA is performed on the duplicate of the scaled data to create PCA-transformed versions of datasets. These PCA versions were used to train alternative versions of the models to evaluate whether further PCA preprocessing could improve the performance of each model. Once optimal hyperparameters for each model are finalized and set, a function is used to train the models on the scaled training data. Once training was completed, the trained models were saved. This process was repeated two additional times from start to end, resulting in three saved models per model type. The same process was then carried out for each model using the PCA-transformed versions of the data. This method was followed to account for variability caused by random initializations or training data order, ensuring reproducible results. By saving multiple trained instances of each model, the likelihood of relying on a single instance that may have achieved peak performance due to chance or dismissing a model that performed poorly due to settling at a local optima.

PCA data Analysis	**Metrics(%)**			
	Accuracy	**Precision**	**Recall**	**F1-Score**
KNN	85.71	85.86	85.71	85.70
Logistic	84.25	84.12	84.25	84.16
Decision Tree	76.62	77.30	76.62	76.80
Linear SVM	84.00	83.60	83.50	83.50
RBF SVM	88.00	88.10	88.20	88.20
CNN	-	-	-	-

TABLE II
Performance Metrics for Different Models - PCA Data

At the end of the training, each model's performance was evaluated using accuracy, F1-score, precision, and recall, with the results displayed along confusion matrices for both validation and testing datasets. For the CNN model, a specialized data loader was required instead of a DataFrame. So, a custom function was developed to process the scaled training, validation, and testing datasets, converting them into a format suitable for the data loader to feed the CNN model in batches. The best model based on validation accuracy was saved for later evaluation on the test set. This process was also adapted for the PCA-transformed data, with additional adjustments made to account for the dimensionality reduction from 784 features to 256.

Similarly, models trained on PCA-transformed training data were evaluated on the corresponding PCA testing data. The average performance metrics for all models are summarized in Tables 1 and 2. The CNN model demonstrated superior performance on the scaled data, followed by the RBF SVM and KNN models. Among these, KNN requires the least computational time to train on a standard CPU.

For the PCA-transformed testing data, both the RBF SVM and the KNN models continued to be the top performers in the same order, the CNN, however, was unable to process

the PCA-transformed data and could therefore not give valid results. This means that RBF SVM did the best on the PCA-transformed testing data with KNN and then Logistic regression following that. This is consistent with the literature [5] that mentions PCA improving SVM performance.

An interesting result from the FashionMNIST dataset is how close the accuracy, F1, precision, and recall scores are indicating a high level of agreement among these evaluation metrics for this dataset.

Fig. 4. Training and Validation Loss over Epochs

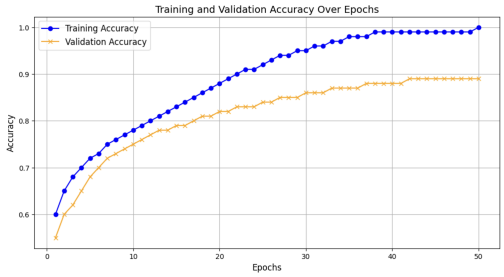

Fig. 5. Training and Validation Accuracy over Epochs

VI. Business Perspective

Image classification technology revolutionizes online and physical retail operations through the automation of product categorization, visual search capabilities, and intelligent inventory management. These systems can process thousands of images of clothing simultaneously to reduce human effort and improve accuracy in product organization. For example, large fashion retailers can automatically sort and categorize merchandise at each of their multiple distribution centers, significantly reducing processing time and labor costs.

For various levels of inspection, manufacturing facilities could implement a hierarchical quality control system using different models. While CNNs can perform more detailed defect detection, other models such as Logistic Regression and Decision Trees can be faster and perform initial rapid screening. For example, in the clothing production line, the basic category verification could be done with Decision Trees, fabric pattern matching with SVM, and CNNs for the detailed quality check, a multi-stage quality control system.

VII. Conclusion

This project analyzed the FashionMNIST dataset by comparing classical machine learning algorithms with a Convolutional Neural Network (CNN) for image classification. The dataset was carefully preprocessed, including standardization and dimensionality reduction techniques (PCA), to improve model performance and efficiency. We tested models like Logistic Regression, KNN, Decision Trees, and SVM, with KNN and RBF SVM showing promising results. However, the CNN outperformed all classical models, achieving the highest accuracy due to its ability to automatically extract features from images. While classical models are easier to understand and interpret, deep learning models like CNNs perform better in tasks that involve complex data, like image classification, by automatically learning complex features and patterns. The results underline the importance of choosing the right model based on the complexity of the task, the size of the dataset, and the available resources. For the future, we plan to experiment with combinations of augmentation for the dataset to unlock higher performance. More details can be found in our blog here: Link

References

[1] Ren, Fei, et al. ”Research on garment image classification algorithm based on machine learning.” 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST). IEEE, 2021.

[2] Sarowar, Md Golam, Md Abdur Razzak, and Md Abdullah Al Fuad. ”HOG feature descriptor based PCA with SVM for efficient & accurate classification of objects in image.” 2019 IEEE 9th International Conference on Advanced Computing (IACC). IEEE, 2019.

[3] Xhaferra, Edmira, Elda Cina, and Luçiana Toti. ”Classification of standard fashion MNIST dataset using deep learning based CNN algorithms.” 2022 international symposium on multidisciplinary studies and innovative technologies (ISMSIT). IEEE, 2022.

[4] Rastogi, Akshit Rajan, et al. ”A Comparative Statistical Analysis Between ML Algorithms & DNN Techniques Using MNIST Dataset.” 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). IEEE, 2022.

[5] Khan, Wisal, et al. ”Data Dimension Reduction makes ML Algorithms efficient.” 2022 International Conference on Emerging Technologies in Electronics, Computing and Communication (ICETECC). IEEE, 2022.

[6] Samia, Bougareche, Zehani Soraya, and Mimi Malika. ”Fashion images classification using machine learning, deep learning and transfer learning models.” 2022 7th international conference on image and signal processing and their applications (ISPA). IEEE, 2022.

[7] Ranzato, Francesco, and Marco Zanella. ”Robustness verification of support vector machines.” Static Analysis: 26th International Symposium, SAS 2019, Porto, Portugal, October 8–11, 2019, Proceedings 26. Springer International Publishing, 2019.

[8] V. G. Costa and C. E. Pedreira, “Recent advances in decision trees: An updated survey,” Artif. Intell. Rev., vol. 56, no. 5, pp. 4765–4800, May 2023.

[9] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, “Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model,” Complexity, vol. 2021, pp. 1–11, Jan. 2021.

[10] E. Zihni, V. I. Madai, M. Livne, I. Galinovic, A. A. Khalil, J. B. Fiebach,and D. Frey, “Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome,” PLoS ONE, vol. 15, no. 4, Apr. 2020, Art. no. e0231166.

[11] Kumar, R.L., Kakarla, J., Isunuri, B.V. and Singh, M., 2021. Multi-class brain tumor classification using residual network and global average pooling. Multimedia Tools and Applications, 80(9), pp.13429-13438.