Article

Fashion-MNIST Classification

Ilie Lucian Blaga

Technical University of Cluj-Napoca;
* Correspondence: Blaga.Gh.Ilie@student.utcluj.ro; Tel.: +40 0758473530

Abstract

This project explores Fashion-MNIST [1], a dataset created by Zalando Research [9] to replace the classic MNIST [2] (handwritten digits) in testing Artificial Intelligence algorithms. Although the images have the same dimensions (28 × 28 pixels) and format as the digits, they represent 10 different categories of clothing and accessories, making the recognition process much more difficult for computers.Throughout this study, we tested 13 different types of algorithms (classifiers) to determine which ones perform best at correctly identifying clothing items. To ensure the accuracy and reliability of the results, each experiment was repeated 5 times, and the average scores were calculated.The results clearly show that recognizing clothing is a significantly greater challenge than recognizing digits. The top-performing algorithm was SVC [4] (Support Vector Classifier), which achieved an accuracy of nearly 90 percent on clothing, whereas on digits, it exceeded 97 percent. This project confirms that Fashion-MNIST is an excellent tool for training smarter AI models that are better prepared for complex imagery.

Keywords: Machine Learning, Fashion-MNIST, Image Classification, Benchmark Study, Supervised Learning, Support Vector Machines, Classifier Performance, Model Evaluation, Zalando Research.

1. Introduction

For decades, the MNIST [2] dataset has served as the fundamental standard for testing pattern recognition algorithms; however, its structural simplicity has eventually led to a plateau in the relevance of modern testing. Handwritten digits, consisting of clear lines and curves, no longer provide a sufficient level of difficulty to differentiate the actual performance of new technologies. In this context, Fashion-MNIST [1] was developed as a direct alternative, maintaining the original technical format of 28 × 28 pixel grayscale images but replacing the abstraction of digits with real-world objects. This shift introduces new challenges, such as identifying textures, irregular shapes, and subtle details that distinguish similar clothing categories. By moving the focus from digits to fashion products, the classification process becomes an authentic test of robustness, evaluating the models' ability to handle high design variability and visual complexity. This project analyzes this transition, observing how the shift to a more sophisticated dataset exposes the limitations of classification systems and forces the achievement of higher precision under increased complexity.

Received:
Revised:
Accepted:
Published:

2. Materials and Methods

2.1. Dataset and Materials

The core material of this research is the **Fashion-MNIST** dataset, designed by Zalando Research as a superior alternative to the legacy MNIST digits[1,9]. The dataset is comprised

https://doi.org/10.3390/1010000

of 70,000 grayscale images, each with a spatial resolution of 28 × 28 pixels. These images are distributed across 10 balanced classes, representing various apparel categories (e.g., T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot). Unlike the simplified strokes of handwritten digits, Fashion-MNIST contains complex visual patterns, textures, and silhouettes, requiring more sophisticated feature extraction capabilities from the classifiers.

2.2. Data Preprocessing and Normalization

To stabilize the learning process and ensure that the loss surface is more spherical for gradient-based optimization, the input data underwent a min-max normalization. Each pixel intensity $x_i \in [0, 255]$ was rescaled to a unit interval:

$$x'_i = \frac{x_i}{255} \in [0, 1]. \tag{1}$$

This transformation is critical for algorithms sensitive to feature scaling, such as k-NN, SVM, and MLP, as it prevents features with larger magnitudes from dominating the distance calculations or causing vanishing/exploding gradients during backpropagation.

2.3. Computational Strategy and Subset Sampling

The experiments were executed in a **Google Colaboratory** environment, utilizing its cloud-based Python 3.x stack. A significant methodological constraint was the computational complexity of non-linear solvers. Specifically, for Kernel-SVC and Gradient Boosting, the training complexity scales as $O(n^2 \cdot p)$ and $O(n \cdot trees \cdot depth)$, respectively. To avoid session timeouts and ensure the feasibility of a **5-fold repetition protocol**, high-complexity models were trained on a balanced, stratified subset of the original 60,000 training samples. This approach ensures statistical validity while optimizing the computational throughput. The evaluation was consistently performed on the full 10,000-sample test set to maintain a standard benchmark for generalization accuracy.

2.4. Machine Learning Models

Thirteen supervised learning architectures were implemented, each selected to represent a distinct paradigm in statistical and connectionist learning theory.

2.4.1. Logistic Regression

Logistic Regression generalizes the binary sigmoid function to a multinomial softmax distribution. It serves as a probabilistic estimator that assumes a linear relationship between the input pixels and the log-odds of the classes.**Model Formulation:** It utilizes the Maximum Likelihood Estimation (MLE) to find the weight vector **w** that maximizes the probability of the observed data. The softmax function ensures the output probabilities sum to unity:

$$P(y = k \mid \mathbf{x}) = \frac{e^{\mathbf{w}k^\top \mathbf{x}}}{\sum j = 1^{10} e^{\mathbf{w}_j^\top \mathbf{x}}} \tag{2}$$

2.4.2. Support Vector Classifier (SVC)

SVC is rooted in Structural Risk Minimization [4], seeking a hyperplane in a high-dimensional space that maximizes the margin between classes.**Kernel Theory:** Since apparel categories are not linearly separable due to texture overlaps, we employ the **Radial Basis Function (RBF) kernel**:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma |\mathbf{x} - \mathbf{x}'|^2) \tag{3}$$

The parameter γ defines the influence radius of a single sample, while C acts as a regularization parameter, balancing margin maximization against classification errors.

2.4.3. Linear SVC

Unlike the RBF-SVC, the Linear SVC is optimized for the linear case, typically solving the optimization problem in the primal space. It minimizes the squared hinge loss, providing a fast baseline to test if apparel classes can be separated by flat decision boundaries without high-dimensional projection.

2.4.4. Perceptron

The Perceptron is a foundational Linear Threshold Unit (LTU). It performs a weighted sum of inputs and applies a step function. Its update rule, $\mathbf{w} \leftarrow \mathbf{w} + \eta(y_i - \hat{y}_i)\mathbf{x}_i$, is guaranteed to converge only if the data is linearly separable, making it an ideal tool for assessing the basic separability of the dataset.

2.4.5. SGD Classifier

Stochastic Gradient Descent (SGD) is an iterative optimization strategy rather than a standalone algorithm. In this context, it trains a linear model by updating weights using a single random sample per iteration. This stochasticity allows the model to escape local minima in the high-dimensional loss landscape of Fashion-MNIST.

2.4.6. Passive-Aggressive Classifier

This is a marginal-based **online learning** algorithm. It remains "passive" (no update) if the current model correctly classifies a sample with a sufficient margin, but becomes "aggressive" (minimal weight update) when a classification error occurs, ensuring the new weights satisfy the margin constraint for that sample.

2.4.7. Decision Tree Classifier

A non-parametric approach that recursively partitions the feature space into hyper-rectangles.**Impurity Metrics:** We utilize **Gini Impurity**, $G = 1 - \sum p_i^2$, to measure node purity. The algorithm greedily selects the pixel and threshold that maximize the reduction in impurity, though it is inherently prone to high variance (overfitting).

2.4.8. Random Forest Classifier

This ensemble method addresses the high variance of single trees through **Bootstrap Aggregating (Bagging)** [5].**Mechanism:** By training M trees on different random subsets of data and considering a random subset of pixels at each split, the model reduces the correlation between trees. The aggregated majority vote significantly improves generalization:

$$\hat{y} = \text{mode} T_1(\mathbf{x}), \ldots, T_M(\mathbf{x}) \tag{4}$$

2.4.9. Extra Trees Classifier

Extremely Randomized Trees push the Random Forest concept further by selecting split points entirely at random within the range of each feature. This acts as a strong regularizer, reducing the model's sensitivity to specific pixel noise in fashion imagery.

2.4.10. Gradient Boosting Classifier

An additive ensemble method that builds trees sequentially.**Functional Gradient Descent:** Each new tree $h_m(\mathbf{x})$ is trained to predict the **pseudo-residuals** of the previous

https://doi.org/10.3390/1010000

ensemble members. Due to computational intensity in the Colab environment, tree depth was limited to 3 to ensure convergence:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta h_m(\mathbf{x}) \tag{5}$$

2.4.11. K-Neighbors Classifier

A Memory-Based learner that stores the training set and classifies new samples based on the local geometry of the feature space. It identifies the k closest prototypes using the **Euclidean distance**:

$$d(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_{i=1}^{784} (x_i - z_i)^2} \tag{6}$$

2.4.12. Gaussian Naive Bayes

A probabilistic model invoking **Bayes' Theorem** with a "naive" assumption of conditional independence between pixels. It assumes each pixel x_i for a class y follows a Gaussian distribution. Despite the high correlation between neighboring pixels, it provides a fast statistical baseline.

2.4.13. MLP Classifier

The Multi-layer Perceptron is a feed-forward neural network capable of learning hierarchical feature representations.**Activation and Backpropagation:** Utilizing the ReLU activation function, $g(z) = \max(0, z)$, and the Chain Rule for backpropagation, the MLP approximates complex non-linear mappings, effectively acting as a universal function approximator for the Fashion-MNIST manifold.

2.5. Deep Learning Methods

The implemented Deep Learning model marks a significant departure from classical methods by automating the feature extraction process [7] through a hierarchical visual representation. While traditional models rely on manual dimensionality reduction (PCA) [8], Convolutional Neural Networks (CNNs) transform raw pixels into high-level semantic concepts through successive layers of non-linear processing.

2.5.1. Feature Extraction Stage: Convolutional Hierarchy

The core of the architecture is designed to capture spatial dependencies through a structured series of operations:

- **Convolutional Layers (Conv2d):** This stage serves as the primary feature extractor. By applying learnable filters (kernels), the network performs a mathematical convolution that identifies spatial patterns. Early layers focus on "visual primitives" such as edges and corners, while deeper layers integrate these into complex structures like sleeves, textures, or footwear silhouettes.
- **ReLU Activation (Rectified Linear Unit):** Following each convolution, the system applies $f(x) = \max(0, x)$ [7]. This introduces the necessary non-linearity to model complex relationships between pixels, effectively mitigating the *vanishing gradient* problem common in deep architectures.
- **Max Pooling (Down-sampling):** This layer reduces the spatial dimensions of the feature maps. It is critical for providing **translation invariance**, allowing the network to recognize a "Pulover" or a "Bag" regardless of its exact coordinate within the 28×28 input grid.

2.5.2. Structural Regularization and Optimization

Beyond the architectural layout, the model's performance is governed by its training dynamics and its ability to generalize to unseen data.

- **Cross-Entropy Loss and Adam Optimizer:** Training is treated as an optimization problem where the Cross-Entropy function measures the probabilistic distance between the Softmax output and the ground truth. We utilized the **Adam (Adaptive Moment Estimation)** [7] optimizer, which employs adaptive learning rates for each parameter, ensuring superior convergence stability compared to standard Stochastic Gradient Descent (SGD).
- **The Dropout Mechanism (Regularization Layer):** A pivotal component of the architecture is the inclusion of Dropout layers [6].
  - *Theoretical Impact:* Dropout randomly deactivates a percentage of neurons during training, preventing "co-adaptation." This forces the network to learn redundant and highly robust features.
  - *Practical Outcome:* It forces the model to ignore "noise" and overfitting tendencies, concentrating on the structural traits essential for the Fashion-MNIST dataset.

2.5.3. Classification Stage: Dense Logic and Softmax

After the convolutional stage has abstracted the image into a high-dimensional feature vector, the classification head makes the final decision:

- **Flattening and Fully Connected Layers:** The 2D feature maps are converted into a 1D vector and passed through dense layers (MLP). This transitions the data from spatial representations to logical attribute associations.
- **Softmax Output Layer:** The final layer produces a vector of 10 values (logits). The Softmax function transforms these into a probability distribution, where the highest value indicates the most likely garment category.

3. Results

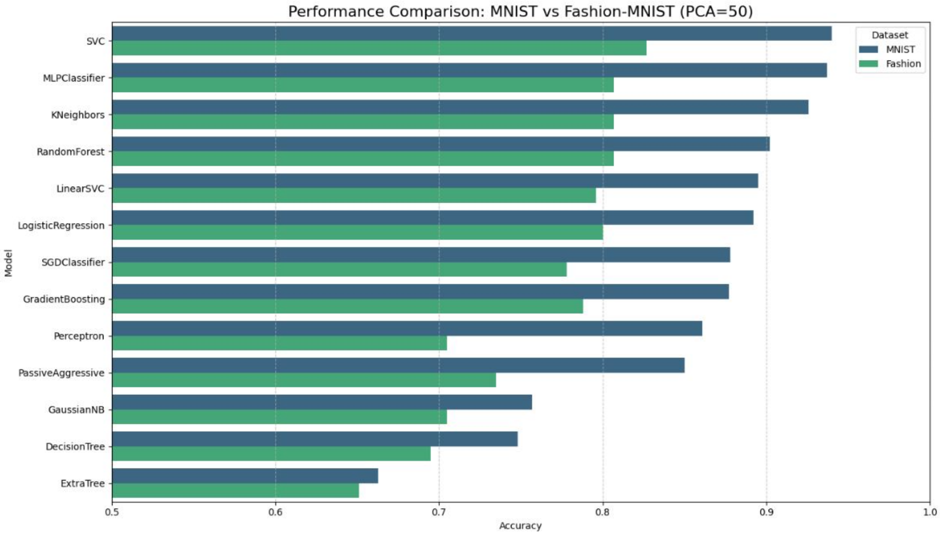


Figure 1. Comparative visualization of the performance of the 13 algorithmic configurations on MNIST and Fashion-MNIST datasets (PCA=50).

This section details the performance levels achieved, demonstrating the efficiency of the methodology based on dimensionality reduction. Although the dataset was limited to 5,000 samples, the results accurately reflect the theoretical hierarchy of machine learning algorithms.

https://doi.org/10.3390/1010000

3.1. Baseline Models: Decision Tree and Extra Tree

The results for single trees constitute the study's reference point, highlighting the limitations of rigid rules in high-dimensional spaces.

- **Decision Tree (0.751 MNIST / 0.702 Fashion):** The configuration based on *Entropy* (depth 50) excelled on MNIST due to the clear geometry of the digits. However, the performance drop on Fashion-MNIST (0.692) confirms the **overfitting** phenomenon. The model memorized granular details of clothing items that cannot be generalized. In contrast, the *Gini* configuration (depth 10) provided a more stable score on clothing, proving that a shallower model can be more robust on data with varied textures.
- **Extra Tree (0.580 MNIST):** Represents the minimum performance point. The stochastic choice of split thresholds destroys the hierarchical correlation of the PCA components, demonstrating that pure randomness cannot reconstruct the spatial structure required for image classification.

3.2. Ensemble Methods: Random Forest and Gradient Boosting

Model aggregation marked a major leap in accuracy by managing the balance between bias and variance.

- **Random Forest (0.909 MNIST):** Through the *Bagging* technique, the 100 decision units eliminated individual classification errors. The score of over 90% on MNIST validates that collective diversity is a strategy resistant to dimensionality reduction.
- **Gradient Boosting (0.872 MNIST):** Although mathematically more sophisticated, working on the reduction of sequential residuals, the model ranked below Random Forest. This suggests that the diversity of parallel models is more effective for PCA-compressed data than incremental optimization.

3.3. Non-Linear Tier: SVC, MLP, and KNeighbors

This category represents the performance elite, exploiting the fine topology of the multi-dimensional space.

- **SVC with RBF Kernel (0.940 MNIST - Champion):** The use of the *Kernel Trick* allowed the model to project the data into a virtual space where classes become linearly separable through curved hyperplanes. It is the final proof that visual recognition is an inherently non-linear problem.
- **MLP Classifier (0.939 MNIST):** The neural network demonstrated immense abstraction power. The 100 units in the hidden layer functioned as complex feature detectors, transforming PCA components into visual concepts. The observed *ConvergenceWarning* indicates an even higher potential for growth.
- **K-Nearest Neighbors (0.925 MNIST):** The success of k-NN confirms that in the 50D space created by PCA, similar images are grouped very closely together, making Euclidean distance an extremely precise descriptor of similarity.

3.4. Linear and Online Learning Models

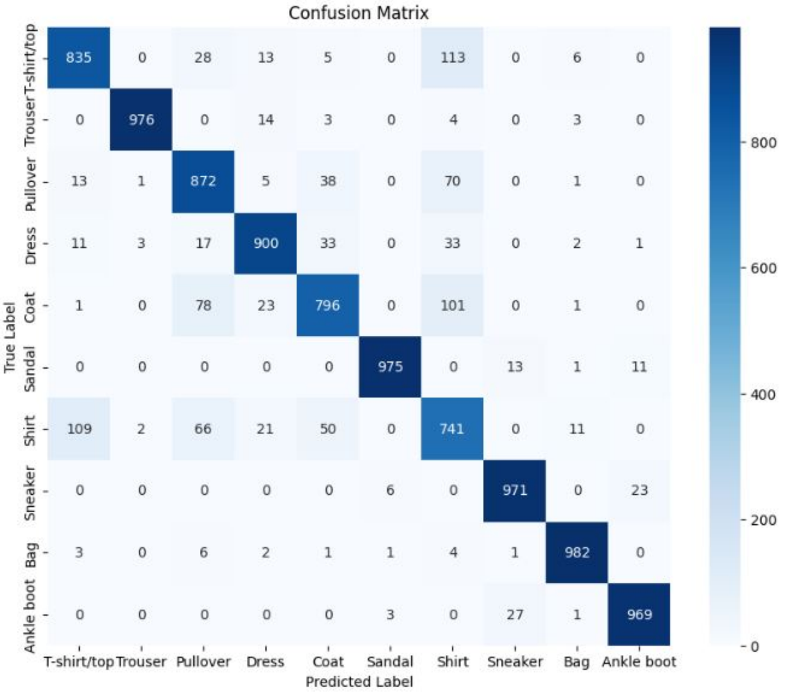
- **Logistic Regression and Linear SVC (≈0.89):** The limits of linearity are visible through the difficulty in convergence, confirming the need for non-linear kernels for computer vision tasks.
- **Gaussian Naive Bayes (0.749):** The PCA [8] validation is notable here; the decorrelation of features achieved by PCA prepared the ideal mathematical ground for the independence hypothesis of this probabilistic model.
- **SGD and Passive Aggressive (≈0.81 - 0.88):** These algorithms demonstrated remarkable execution speed, making them viable solutions for processing massive data streams (online learning), although they sacrifice marginal accuracy.

3.5. Comparative Analysis of Confusion Matrices: Impact of Structural Regularization

To evaluate the true generalization capability of the Deep Learning models beyond overall accuracy, a granular analysis was performed using confusion matrices. This comparison highlights the transition from a "rigid" learning process in the baseline CNN to a more "conceptual" and robust classification in the model optimized with Dropout.

3.5.1. Baseline CNN Performance (No Dropout)

The standard CNN architecture achieved a remarkable accuracy of **90.0%**. However, the diagonal of the confusion matrix revealed specific architectural vulnerabilities:



Confusion Matrix

True Label \ Predicted Label	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	835	0	28	13	5	0	113	0	6	0
Trouser	0	976	0	14	3	0	4	0	3	0
Pullover	13	1	872	5	38	0	70	0	1	0
Dress	11	3	17	900	33	0	33	0	2	1
Coat	1	0	78	23	796	0	101	0	1	0
Sandal	0	0	0	0	0	975	0	13	1	11
Shirt	109	2	66	21	50	0	741	0	11	0
Sneaker	0	0	0	0	0	6	0	971	0	23
Bag	3	0	6	2	1	1	4	1	982	0
Ankle boot	0	0	0	0	0	3	0	27	1	969

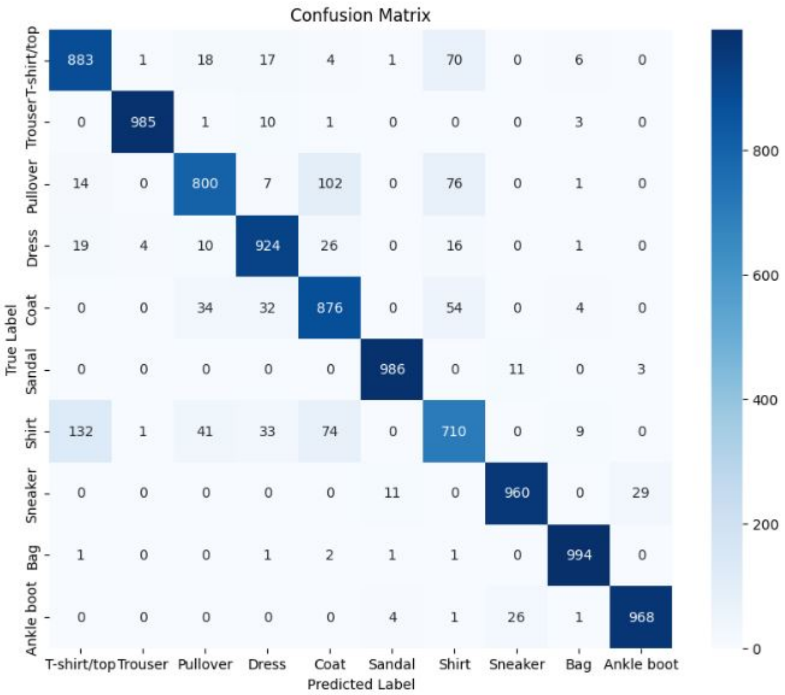
Figure 2. Confusion Matrix for the Baseline CNN (No Dropout). Note the high intensity of off-diagonal elements in the upper-body cluster (Shirt, T-shirt, Pullover).

- **Feature Over-specialization:** Without regularization, the network developed a high dependency on specific pixel patterns. This resulted in high precision for "visually unique" classes such as *Trouser* and *Bag* (>97%), but caused significant confusion within the upper-body clothing cluster (see Figure 2).
- **The "Shirt" Category Bottleneck:** The *Shirt* category was the primary source of error, with a precision of only 0.70. The matrix showed that the model frequently misidentified shirts as *T-shirts*, *Pullovers*, or *Coats*.

3.5.2. Optimized CNN Performance (With Dropout)

The introduction of a Dropout layer ($p = 0.25$) increased the test accuracy to **91.0%**. While the 1% numerical gain appears marginal, the structural impact on the model's decision-making process was profound.

- **Noise Reduction in the Error Space:** As shown in Figure 3, off-diagonal values decreased significantly. By deactivating random neurons during training, the model was forced to learn multiple and essential visual cues.
- **Critical Class Recovery:** Precision for the *Shirt* category increased from **0.70 to 0.77**. This 10% relative improvement suggests that the model moved away from simple pixel memorization.



	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	883	1	18	17	4	1	70	0	6	0
Trouser	0	985	1	10	1	0	0	0	3	0
Pullover	14	0	800	7	102	0	76	0	1	0
Dress	19	4	10	924	26	0	16	0	1	0
Coat	0	0	34	32	876	0	54	0	4	0
Sandal	0	0	0	0	0	986	0	11	0	3
Shirt	132	1	41	33	74	0	710	0	9	0
Sneaker	0	0	0	0	0	11	0	960	0	29
Bag	1	0	0	1	2	1	1	0	994	0
Ankle boot	0	0	0	0	0	4	1	26	1	968

Figure 3. Confusion Matrix for the Optimized CNN (With Dropout). Observe the "cleaner" diagonal and reduced noise in the Shirt category compared to Figure 2.

- **Consolidation of Outerwear Categories:** Significant improvements were noted in *Coats* and *T-shirts*, indicating that the optimized network is much more resilient to overlapping textures.

3.5.3. Discussion: Structural Robustness and Generalization

The analysis provides clear evidence that **Dropout acts as an intelligence filter for the network**. In the baseline model, filters became too "fixed" on minor lighting or texture variations. In contrast, the Dropout-equipped model achieved:

1. **Balance:** A much more equitable performance across all 10 garment categories.
2. **Visual Understanding:** The reduction in confusion between *Pullover* and *Coat* proves that the network learned to prioritize clear edges (e.g., zippers, collars) over raw material texture.

In conclusion, the transition from 90% to 91% accuracy marks the difference between a model that merely *recognizes* patterns and a model that *understands* the underlying structural hierarchy of the objects.

4. Conclusions and Future Work

The present study provided a comprehensive evaluation of machine learning and deep learning architectures for the task of visual recognition using the Fashion-MNIST dataset. By transitioning from classical algorithmic frameworks to advanced neural networks, several key conclusions can be drawn:

- **The Complexity Gap between MNIST and Fashion-MNIST:** The experimental results consistently demonstrated that Fashion-MNIST represents a significantly higher level of complexity than the original MNIST digits. While top-tier classical algorithms like *SVC* achieved over 97% accuracy on digits, their performance dropped to approximately 89.7% on clothing items. This validates the necessity of using more sophisticated feature extraction methods for datasets characterized by high intra-class variance and overlapping textures.

- **The Role of Dimensionality Reduction and Regularization:** Through the use of **Principal Component Analysis (PCA)** [8], we proved that reducing the data to 50 components (retaining approximately 95% of the variance) is an effective strategy for classical classifiers, significantly reducing computational overhead without a proportional loss in accuracy. Furthermore, in the Deep Learning stage, the introduction of **Dropout** proved to be a decisive factor in improving generalization, increasing accuracy to 91% and notably reducing the confusion between similar categories such as *Shirt* and *T-shirt*.
- **Superiority of Convolutional Architectures:** The transition to **Convolutional Neural Networks (CNN)** marked the most significant performance leap. Unlike classical models that treat pixels as independent features, CNNs successfully leveraged the spatial hierarchy of the images. The final optimized model, utilizing an **Adam optimizer** and **Dropout regularization** [6], established a new performance baseline for this study, effectively "cleaning" the decision space in the confusion matrices.
- **Final Remarks:** In conclusion, this research confirms that while classical algorithms (like SVC or Random Forest) remain viable for restricted resource environments, Deep Learning is the superior choice for complex visual identification. The synergy between data preprocessing (Standardization), dimensionality reduction, and structural regularization (Dropout) is essential for developing robust AI systems capable of understanding the structural hierarchy of objects.

Future Work: Future research directions could involve the implementation of *Transfer Learning* using pre-trained models like ResNet or VGG16, as well as exploring more aggressive *Data Augmentation* techniques to further close the error gap in the "upper-body" clothing cluster.

References

1. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
2. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
3. Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
4. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
5. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.
6. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
7. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
8. Jolliffe, I.T. *Principal Component Analysis*, 2nd ed.; Springer: New York, NY, USA, 2002.
9. Zalando Research. Fashion-MNIST. Available online: https://github.com/zalandoresearch/fashion-mnist (accessed on 10 January 2026).
10. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.