Texture Feature Classification on Fashion-MNIST: A Comparative Study of Machine Learning Algorithms

Arjun Balaji, Rahul Suresh
Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar (SSN) College of Engineering, Kalavakkam, Chennai - 603110

***Abstract*—Robust feature extraction techniques are essential for bridging the gap between raw pixel data and semantic meaning in image classification. Using the Fashion-MNIST dataset, this project investigates texture-based feature classification as a more sophisticated substitute for conventional digit recognition tasks. We employ an end-to-end machine learning pipeline that includes model training, image normalization, Histogram of Oriented Gradients (HOG) feature extraction, and a thorough performance assessment. Gaussian Naïve Bayes and Random Forest are two basic classical algorithms that we evaluate. Our findings show that classification accuracy and computational efficiency are clearly traded off. Despite having a much longer training time, the Random Forest classifier, an ensemble approach, outperformed the Naïve Bayes model with an accuracy of 82.49% versus 70.90%. This report provides details on the mathematical foundations of the methods utilized, the experimental setup as well as an analysis of the class-wise performance limitations.**

***Index Terms*—Fashion-MNIST, Histogram of Oriented Gradients, Feature Extraction, Random Forest, Naïve Bayes, Texture Classification, Ensemble Learning**

I. Introduction

Image sorting might seem simple these days, but it still grabs a lot of attention in visual computing. Newer systems love their deep CNN layers, finding patterns on their own without anyone handcrafting the rules. But honestly, the older methods still hold up. They let you see exactly how features get pulled out of raw pixels, trimmed down, and fed into models step by step.

Fashion-MNIST shakes things up in this space. Instead of the old benchmark with digits, now you get jackets, shoes, and bags, all squeezed into those same dim little 28x28 frames. No more crisp lines like before. Here, it's all about soft outlines, hints of fabric, and rough patches. It's not about counting loops or checking for curves anymore; it's about how things bend and stretch across the image.

Raw pixel values are messy and get thrown off by noise or small shifts. So, this approach focuses on features built from texture, not just the pixels themselves. That shift gives you patterns that stay stable, even when the image changes a bit.

Here's what this work brings to the table: It starts right at the pixel grid, moving through layers that turn raw input into steady patterns. With each step, features pop out that don't care about rotation or scale. The process keeps the important stuff while smoothing out the chaos, building up a solid structure from the ground up.

It pulls out strong texture and shape features using the Histogram of Oriented Gradients (HOG) algorithm.

And then, it pits Gaussian Naïve Bayes against Random Forest, both in theory and in real tests. One leans on probability and assumptions, the other on combining lots of decisions. They take different routes: Naïve Bayes watches distribution patterns, while Random Forest lets a bunch of trees vote. By testing both on the same ground, you can see exactly where each one struggles or outshines the other.

II. Literature Survey

Image classification has changed a lot over the years. We've moved from carefully designing features by hand to letting algorithms figure out what matters on their own. Before deep learning took over, people spent a lot of time building tools to pick out things like color, texture, or shape from images. Then they'd pass those measurements into classic machine learning models.

Take HOG, for example. Dalal and Triggs came up with the Histogram of Oriented Gradients back in 2005. At first, it was all about finding people in photos. HOG made a big splash because it showed you don't need to know the exact spot of every edge or gradient; just knowing how those edges and gradients are spread out in small patches tells you a lot about what's in the image. It worked surprisingly well, even if the lighting or angle changed, so it became a go-to method in early computer vision.

Then in 2017, Xiao and colleagues introduced the Fashion-MNIST dataset. The old MNIST digit dataset had basically become too easy; simple models could nail over 99

When it comes to actually classifying images, ensemble methods like Random Forests were a popular choice; Breiman introduced them, and they caught on quickly. People would often combine features like HOG with Random Forests for better results. Bosch and colleagues showed that when you pair strong texture features with Random Forests, you get image classifiers that are both accurate and less likely to overfit compared to single decision trees or basic probabilistic models. This combo held up well, even as datasets and expectations grew.

III. Methodology

The proposed system follows a sequential four-stage pipeline to process the data, extract features and train the classifiers.

A. Dataset Description and Normalization

The 70,000 grayscale photos in the Fashion-MNIST dataset are divided into ten mutually exclusive classes of apparel (e.g., T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot). To guarantee objective assessment, the dataset is pre-partitioned into 60,000 training images and 10,000 testing images.

The dataset contains raw pixel values ranging from 0 (black) to 255 (white). All images were normalized in order to improve model convergence, guarantee numerical stability, and keep features with wider numeric ranges from controlling the objective functions. The range of $[0.0, 1.0]$ was used to min-max scale the pixel intensities p:

$$p_{norm} = \frac{p - p_{min}}{p_{max} - p_{min}} = \frac{p}{255}$$

B. Feature Extraction: Histogram of Oriented Gradients

We used HOG to extract texture-based features rather than raw pixel matrices, which have high dimensionality and lack spatial invariance. By allocating gradient orientations in localized spatial grids, HOG is able to capture local object appearances.

The first step in the mathematical formulation is to calculate the image's vertical (g_y) and horizontal (g_x) gradients. Usually, a 1-D centered derivative mask is used to compute the gradients for a pixel at (x, y) with intensity $I(x, y)$:

$$g_x = I(x+1, y) - I(x-1, y)$$
$$g_y = I(x, y+1) - I(x, y-1)$$

From these gradients, the magnitude m and orientation θ are computed:

$$m(x, y) = \sqrt{g_x^2 + g_y^2}$$
$$\theta(x, y) = \arctan\left(\frac{g_y}{g_x}\right)$$

Next, the image is separated into tiny spatial areas known as "cells" (e.g., 8×8 pixels). A local 1-D histogram of gradient directions, or orientations, is constructed for every cell's pixels. The local responses are contrast-normalized over broader spatial regions known as "blocks" in order to take illumination and contrast variations into account.

HOG was set up for this project with nine orientation bins, 8×8 pixels for the cell size, and 2×2 cells for the block size. The 28×28 raw pixel image is compressed into a highly dense, discriminative 1D feature vector in this particular configuration.

C. Model Architecture and Training

The extracted HOG feature vectors were used to train two distinct supervised learning models to observe the difference in the 2 learning paradigms.

1) Gaussian Naïve Bayes: Based on Bayes' Theorem, Naïve Bayes is a probabilistic, generative classifier. The "naïve" assumption is that, given the class label, all features are conditionally independent. The posterior probability is computed as follows for a feature vector $X = (x_1, x_2, ..., x_n)$ and a class label y:

$$P(y \mid X) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(X)}$$

We use the Gaussian variant of Naïve Bayes, which assumes that the continuous values associated with each class are distributed according to a normal (Gaussian) distribution, because HOG features are continuous.

2) Random Forest Classifier: During training, Random Forest, a discriminative ensemble learning technique, builds a large number of decision trees. It works on the basis of random feature selection and "bagging" (Bootstrap Aggregating). Random Forests rectify the tendency of individual decision trees to overfit to their training set by combining the predictions of multiple independent trees (by majority voting for classification).

For this experiment, the Random Forest model was instantiated with an ensemble of $N = 100$ estimators (trees) to capture highly complex, non-linear interactions between the HOG texture features.

IV. Results and Discussion

The 10,000-image testing set was used to assess the models. The overall accuracy and computational training time for both classifiers are summarized in Table I.

TABLE I
Model Performance Comparison

Algorithm	Testing Accuracy	Training Time (s)
Gaussian Naïve Bayes	70.90%	0.07
Random Forest (100 Trees)	82.49%	153.15

A. Analysis of Random Forest Performance

With an overall accuracy of 82.49%, the Random Forest classifier greatly outperformed the Naïve Bayes model. Examining the classification report in more detail shows that Random Forest produced very good precision and recall (often $\geq$ 0.90) for classes like trousers, bags, and ankle boots that have very different structural silhouettes. Random Forest successfully parsed the dense gradient maps generated by HOG due to its ability to map complex feature interactions and non-linear decision boundaries.

B. Analysis of Naïve Bayes Performance

The accuracy of the Gaussian Naïve Bayes model was 70.90%. Its underlying independence assumption theoretically predicts this decline in performance. The gradient histograms of neighboring HOG cells in an image are strongly correlated, and the shoulder's texture greatly influences the sleeve's texture. Naïve Bayes ignores this spatial correlation.

In particular, the model had a lot of trouble with visually similar classes; it frequently confused shirts with T-shirts, pullovers, and coats, resulting in an F1-score of just 0.35 for shirts.

C. Computational Trade-Off

Random Forest took more than two and a half minutes (153.15 seconds) to build its 100 decision trees, despite producing better predictive power. On the other hand, the Gaussian Naïve Bayes model demonstrated remarkable computational efficiency, fitting the distribution parameters in just 0.07 seconds. Predictive accuracy increases with computational complexity, highlighting a well-known trade-off in machine learning.

V. Conclusion and Future Work

Using HOG feature extraction, this project effectively designed and implemented an end-to-end texture classification pipeline on the difficult Fashion-MNIST dataset. A basic tenet of machine learning is highlighted by the comparative analysis: ensemble approaches such as Random Forest capture intricate feature interactions and yield reliable, highly accurate classifications, but they come at a high computational cost. Though their near-instantaneous training times make them unsuitable for severe real-time constraints, generative models such as Gaussian Naïve Bayes are constrained by rigorous independence assumptions that impair accuracy on complex visual data.

Future research might concentrate on optimizing the Random Forest architecture's hyperparameters, using Grid Search to determine the ideal tree depth and ensemble size. More gains in accuracy and training efficiency might also result from experimenting with different texture descriptors, like Local Binary Patterns (LBP), or dimensionality reduction strategies, like Principal Component Analysis (PCA), before classification.

References

[1] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[3] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[4] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.