Texture Feature Classification Using Fashion-MNIST

Hansika N.M, Saisaravanan S, Chandrini Chandrasekaran Kalaivanan
SSN College of Engineering

***Abstract*—Classification of images is a major study area in computer vision, and is very significant when analyzing fashion, or for use in visual recognition systems. Much of the current work around doing classification of images utilizing deep learning models leaves classical techniques for machine learning behind when combined with sufficient feature extraction of an image (getting useful information from an image).**

Thus, this project will focus on the texture classification of the database of images known as Fashion-MNIST; this database is extensively used for running benchmarks using machine learning algorithms. The first step in our process will involve normalizing the images we have by converting the pixel values of the images into a standard value across all images being analyzed. Next, we will extract texture features from the images, using two types of descriptors, which will allow us to create a feature vector for each image based upon the two types of texture features we extracted. Naive Bayes and Random Forest will be our two classifiers, which will allow for evaluation of the performance of both classifiers on the texture features we extracted.

Our experimental results will indicate that Random Forest outperformed Naive Bayes in classifying images based on the texture features we extracted. This indicates that Random Forest can capture more complex patterns of texture than Naive Bayes. This project presents the importance of feature engineering in classical machine learning/feature extraction methods for classifying images.

***Index Terms*—Texture Feature Extraction, Image Normalization, Fashion-MNIST, Histogram Features, Gradient Magnitude, Naïve Bayes, Random Forest, Image Classification, Machine Learning**

I. Introduction

The use of texture in image/photograph classification is an area that has received much attention in both the fields of image processing as well as pattern recognition because it contains significant amounts of information regarding the visual characteristics of images. The analysis of texture can assist in distinguishing images, based on how they vary in terms of intensity and surface patterns; therefore, a technique to analyze or interpret textures is also widely used in various areas of application (i.e., in medicine, in remote sensing, and in identifying material properties).

The majority of the time when an individual is interpreting the characteristics of texture patterns by hand, they will typically be doing so in an inefficient and subjective manner, which has led to the exploration of various machine learning techniques that can automate the process of interpreting textures. For this project, normalized texture features will then be extracted from images and employed to train machine learning classifiers (e.g., Naïve Bayes and Random Forest), and subsequently, the performance of these classifiers will also be analyzed to evaluate the usefulness and effectiveness of using texture features as a means of classifying images.

II. Literature Survey

The earliest studies into texture-based image classification were based on statistical features constructed manually, such as Gray Level Histogram (GLH) and Gray Level Co-occurrence Matrix (GLCM), to define the distribution of gray level intensities and the spatial relationship between those intensities in images. The traditional classifiers used to classify the various textures were limited by their sensitivity to noise and illumination changes, so the performance of the statistical features used for classification was poor.

Researchers began to investigate gradient-based and filter-based approaches to capture the local texture variations and edges in images. Features extracted using gradients, Gabor filters, and wavelet transforms improved the robustness of the methods and provided increased discrimination between textures that have similar intensity characteristics. The structural information captured with the gradient-based and filter-based methods could not be captured by the use of the statistical methods.

With the introduction of machine learning, texture classification methods such as Naïve Bayes and Random Forest became popular methods for texture classification. Naïve Bayes offered ease of use and efficiency when implementing histogram-based features for classification; Random Forest improved the accuracy of classification by managing the high-dimensional feature space and complex decision boundaries associated with many texture classifications. Recent research indicates that combining multiple texture features with appropriate classifiers can produce more reliable and accurate classifications of textures.

III. Methodology

A. Dataset Organization

The study utilizes Fashion-MNIST, which is a recognized benchmark in image classifiers. The Fashion-MNIST branch is a collection of 70,000 monochromatic photographs of clothing at 28×28 pixels each, separated into 10 classes representing different categories (t-shirts, pants, dresses, shoes, bags, etc.).

The Fashion-MNIST data split contains:

- 60,000 training data
- 10,000 test data

Since each image generated is a single-channel (monochrome) array, processing is computationally manageable due to the low-resolution nature of each image while storing discernible texture and dimensions. Since the purpose of the study is to evaluate different textural based feature extraction methods, Fashion-MNIST provides an appropriate balance between ease of interpretation and complexity of the feature set.

Finally, the Fashion-MNIST is used without modification to the proportionate distribution of the classes, to ensure standardized benchmarking.

B. Pre-processing and Input Construction

1) Image Normalization: To maintain numerical stability when extracting features from an image or training a model, all images must be converted from the whole pixel space (integer values) into floating point by normalizing them to the interval $(0, 1)$. In addition to preventing the complete override of one intensity scale by another, normalization will create a consistent method of computation based on the gradient for each image regardless of its value.

Formally,

$$I_{\text{normalized}} = \frac{I}{255} \tag{1}$$

where I represents the original pixel intensity values.

2) Histogram-Based Texture Features: We calculate every image's intensity histogram to determine the distribution of global textures.

For each image:

1) The intensity of pixels in an image is divided into 32 different levels between 0 and 1.
2) The histogram is created.
3) The histogram is normalized.

A single representation will show:

- The overall brightness values of all the pixels
- How the intensities of the pixels are distributed
- The overall shape of the textures

Features of the histogram are stable against minor spatial displacement, but spatial structural characteristics are not included.

3) Gradient-Based Structural Features: We utilize the Sobel operator to identify gradients on an image for its structure and edge-based aspects.

Steps:

- Calculation of horizontal and vertical gradients for the image
- Magnitude of gradients calculated using

$$G = \sqrt{G_x^2 + G_y^2} \tag{2}$$

After the above is completed, the magnitude image of gradients will be transformed to a flattened vector as a feature representation.

The model captures:

- Density of edges
- Shape boundaries
- Structural complexity
- Fine-textured variations

Unlike histogram features, gradient features maintain geographical structural properties that are critical for distinguishing visually alike classes of clothing (i.e., shirt vs t-shirt).

4) Feature Fusion: By horizontally stacking histogram feature data and the features from the gradient magnitude, both global (overall brightness of the image) and local (structural variation) contributions are blended together to create an overall image feature representation with 816 values per image.

Using this approach, models leveraging both types of feature data can utilize a single representation of data to achieve improved performance through category separation, without requiring the use of deep convolutional neural networks for feature construction.

C. Model Training and Fine-Tuning

1) Gaussian Naïve Bayes: Assuming the features are independent, Gaussian Naïve Bayes models the features independently using a Gaussian distribution. As simple as it may be, it forms a reasonable starting point for assessing feature separability.

The advantages of using the Naïve Bayes classifier are:

- Fast training
- Low computational resource usage
- Handles high-dimensional input

The disadvantages of using a Naïve Bayes classifier include:

- May perform poorly if there is a lot of correlation among features (e.g., gradient-based representations) because of the independence assumption

2) Random Forest Classifier: In order to address the constraints imposed by linear and independence-based models, we utilize a Random Forest classifier using 200 decision trees. Random Forest was selected for the following reasons:

- High-dimensional feature spaces can be efficiently processed
- Tolerant to noise
- Able to describe non-linear interactions between features
- Reduces overfitting via ensemble averaging

Each tree is created using a bootstrap sample of the training data, and the final prediction is provided through the majority vote of all the trees.

No elaborate hyper-parameters were tuned because the focus was on evaluating the effectiveness of the features rather than optimizing the model.

D. Performance Evaluation and Inference

1) Evaluation Metrics: Model performance is evaluated using:

- Accuracy
- Macro F1-score

Accuracy measures overall classification correctness, while Macro F1-score ensures balanced evaluation across all classes, preventing dominance by majority classes.

2) Confusion Matrix Analysis: Class-wise performance is evaluated using a confusion matrix. Specifically, it enables:

- Identification of frequently misclassified classes
- Observation of structural misclassification patterns
- Evaluation of robustness of each class individually

The Random Forest classifier performed significantly better than Naïve Bayes in terms of overall accuracy and macro F1-score. This indicates that modeling non-linear relationships between fused texture features is important for texture-based classification.

E. Inference Pipeline

The following steps occur during the inference process:

1) The input image is normalized
2) Histogram and gradient features are extracted
3) The features are concatenated
4) A class label prediction is made using the trained classifier
5) The final output is returned from the model prediction

This inference pipeline is lightweight and computationally efficient, making it suitable for low-resource or real-time environments.

IV. Results and Discussion

Using the Fashion-MNIST dataset as the evaluation metric, the proposed texture classification scheme was validated. There are sixty thousand training images and ten thousand testing images in the Fashion-MNIST database all of which are sized 28×28 pixels. After images were normalized, texture descriptors were extracted from each image via histogram approaches and gradient-based approaches, producing a total feature vector with length 816. All of the classifiers utilize Naive Bayes and Random Forest Classification methods. The results obtained from the Naive Bayes model were 62.77% accuracy with a macro F1 score of 0.5952. The Random Forest classifier has a much better result in terms of classification accuracy at 88.47% with a macro F1 score of 0.8836. Additionally, when looking at the confusion matrix created for the Random Forest classifier there is very strong diagonal dominance, meaning that lots of the sample cases were correctly classified. However, some classes are misclassified because they fall under similarly appearing classes visually, including T-shirt/top versus shirt, and pullover versus coat. This confusion occurs because the Fashion-MNIST images are grayscale and have a number of texture features that are very similar between these two categories of clothing. The randomforests superior performance can be attributed to the following: • Ability to model non-linear decision boundaries • Robustness to correlated features; • Using an ensemble voting method to decrease variance; and • More effectively handling high-dimensional, high-quality features. On the other hand, Naive Bayes assumes features are independent, which is not true for texture descriptors, resulting in poorer performance. Overall, this experiment demonstrates that the use of hand-crafted texture features through an ensemble learning method provides a great baseline benchmark for image classification and does not require deep learning models. Fig. 1 shows the confusion matrix of the Random Forest classifier.

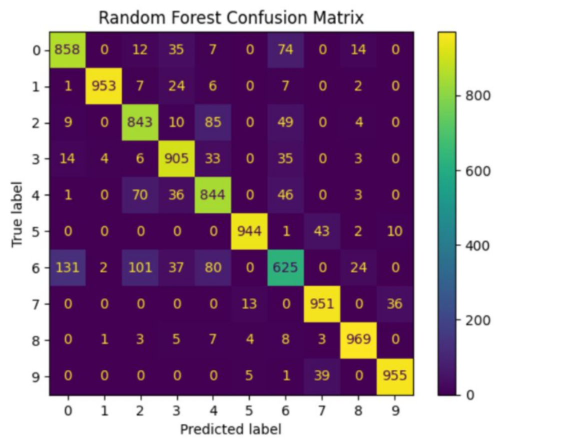

Fig. 1. Confusion matrix of the Random Forest classifier on the Fashion-MNIST test set.

Based on the confusion matrix, there is strong diagonal dominance, which means that most classes are correctly predicted. There are small confusions between classes that look similar, namely T-shirt/top and Shirt, and Pullover and Coat.

V. Conclusion

A full image classification and texture feature extraction procedure was performed on the Fashion-MNIST dataset, where the images were standardised before converting them into a set of discriminative feature vectors, using both histogram and gradient descriptors. To determine which classifier would yield the best classification accuracy, two traditional classifiers (Naïve Bayes and Random Forest) were tested against the dataset.

Random Forest outperformed Naïve Bayes with an overall classification accuracy of 88.47%. This outcome shows that applying traditional machine learning classification techniques, coupled with effective extraction of texture features from images, can produce competitive results to classify grayscale fashion images.

While the final classification results were reasonably accurate, confusion remains for classes that have similar visual characteristics, indicating the limitation of hand-crafted textural feature vectors. Future work may include exploring advanced descriptor methods (such as HOG or LBP), feature selection algorithms, or leveraging deep-learning based approaches to further improve classification accuracies.

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] E. Xhaferra, E. Cina, and L. Toti, "Classification of Standard Fashion MNIST Dataset Using Deep Learning Based CNN Algorithms," in *Proc. IEEE*, 2023.

[3] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.
[5] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[6] S. S. Kadam, A. C. Adamuthe, and A. B. Patil, "CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset," 2021.