Multiclass Apparel Identification Based on HOG Feature Extractor using SVM and Softmax

Prof.Vijayalaxmi Kadroli
Department of Information Technology,
Terna Engineering College, Nerul
Navi Mumbai, India

Yash Kerkar
Department of Information Technology,
Terna Engineering College, Nerul
Navi Mumbai, India

Amit Haral
Department of Information Technology,
Terna Engineering College, Nerul
Navi Mumbai, India

Kapil Jain
Department of Information Technology,
Terna Engineering College, Nerul
Navi Mumbai, India

Abstract— **Image classification and recognition plays an important role in many applications, like online shopping, driverless cars, automation, similar item retrieval queries, etc. In this project we have presented the identification of fashion items in an image. Given an image our model can identify whether it contains any fashion item or not. It can identify items like shirt, shoes, t-shirt, trousers, handbag and 6 other items. Our model consists of two things which are a feature extractor and a classifier. Based on research and experimental work we have selected HOG (Histogram of Oriented Gradients) as feature extraction method and two classifiers which are SVM (Support Vector Machine) and Softmax. It is a very tough task to select appropriate model for classification. It requires training and testing various models and techniques. However, we are able to achieve excellent results using our model.**

Keywords—Multiclass Apparel Classification, Machine Learning, Object Identification, Image Classification, SVM Classifier, Softmax Classifier.

I. INTRODUCTION

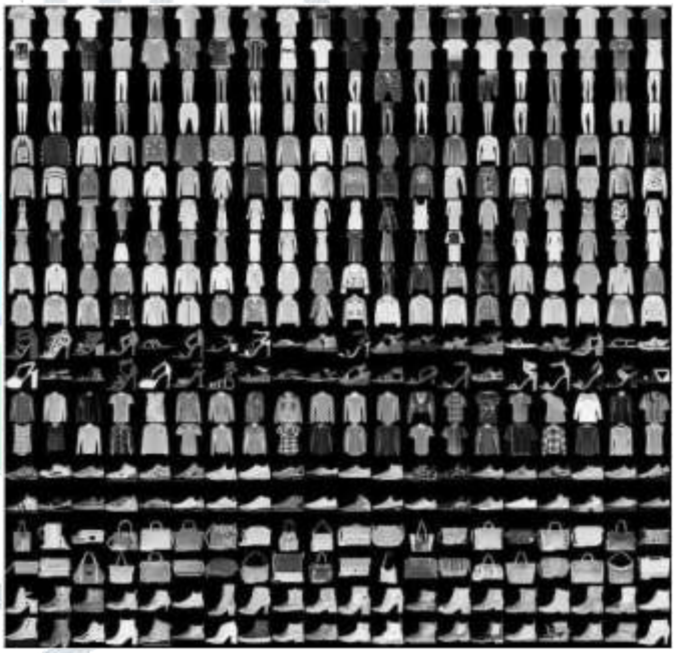
The most popular applications in computer vision are object classification and object recognition. Object classification means to classify an image into its right class. Classification comes handy in lot of automation tasks like inventory management, items retrieval for queries and also driverless cars, online shopping etc. There are many models for classification like logistic regression, Random forest, decision tree, SVM, Softmax etc. There are also various pre-processing algorithms like LBP, HOG, SIFT, smooth etc. Though the performance of HOG feature descriptor is excellent compared to others. In this project we have classified the images in Fashion-MNIST dataset using HOG feature extractor and SVM / Softmax classifier.

Fashion-MNIST (F-MNIST) is a dataset consisting of 70000 fashion images. This dataset is developed by Zalando Research Company. The data is been divided into two parts which are – 1) 60,000 images for training 2) 10,000 images for testing. The dataset consists of 10 different classes, some of which are –T-shirts, trousers, shoes, coat, handbag etc. All images in the dataset are greyscaled.

II. LITERATURE SURVEY

The images in the dataset are very small of the size 28 by 28 pixel. They are greyscaled which means every pixel in the image is represented by only one value ranging between 0-255 which represent shades of grey. Though the images are small the dataset is large enough for efficient training.

Figure 1: Random Images from Fashion MNIST dataset



HOG was developed by Dalal and Triggs (2005) [12] for the human detection and it is one of the most popular and successful feature extractors in pattern recognition and computer vision. Refering to research paper presented by Ebrahimzadeh and Jampour very high accuracy was achieved on Hand written digits dataset using SVM + HOG. In one of the papers presented by Khan, H.A (2017) he has introduced a new method called Multiple Cell Size (MCS) for improving feature vectors of HOG. By using MCS approach along with HOG excellent results have been achieved on classifications problems.Also the combination of HOG and SVM works very well for classification problems. Improvements based on Chain Code Histogram (CCH) for recognition of handwritten digits was proposed by Qian, Y. and Xichang (2013) improves the speed of training and recognition and this reduces the feature dimension.

III. PROPOSED METHODOLOGY

A. *Pre-processing and Feature Extraction*

There are various methods for pre-processing like smooth, dilate, max etc. We have chosen HOG as pre-processor because experimental results of HOG show excellent performance. Before advancing to training the SVM classifier and evaluating the results a preprocessing task is introduced to decrease noise artifacts produced while collecting samples of images.

Pre-processing provides better feature vectors than just the raw pixels. Preprocessing is a very important task because it reduces the noise or the not so useful information from the image. Here we have used HOG feature extractor only. A feature vector is produced for every image after applying HOG on it.

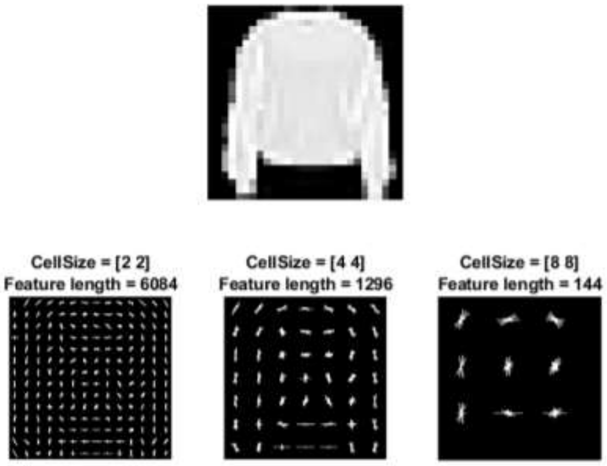
B. *Histogram Of Oriented Gradients (HOG)*

The concept of HOG is that it captures the shape of the object in the image using the edges and corners of the object. HOG is a simple yet effective feature descriptor for images. HOG performs very well compared to SIFT and LBP which are also feature extractors. It is also proved that HOG feature vectors are very useful for object detection. Using HOG the shape and appearance of the main object in the image can be captured.

It divides the image into small cells like n-by-n and computes the edge directions. Normalizing the histograms can also help in increasing accuracy. HOG basically finds the distribution of directions of gradients in the feature vector. In the first step we find the Gradients(x and y derivative) of the image, for which the magnitude is large around the edges and corners (regions of intensity change) because edges pack in more information about the shape of an object rather than than flat regions.

In figure 2 the features extracted from a single image based on different cell sizes is given. Looking at the images we can easily conclude that cell size 2-by-2 contains more information than cell size 8-by-8. But 2-by-2 cell increases the size of the feature vector. A good choice is 4-by-4 cells. By using 4-by-4 cell size we are also able to capture good amount of information along with a small sized feature vector. After applying HOG on images using 4-by-4 cell size we get feature vector of size 1296 for each image. We now have a new dataset in which these features represent the image.

Figure 2: Extracted Features of an Image



C. *Support Vector Machines (SVM)*

SVM is one of the most popular and powerful classifiers in machine learning. SVM is famous because of the kernel trick. Kernels are used to solve a non-linear problem with a linear classifier. Kernels work very fast and efficiently with SVM using the kernel trick. Though here we are not using a kernel as the dataset is not very complex. Various applications of SVM include pattern recognition, text recognition, face recognition etc. In this part we utilize SVM for building the model.

The main goal of any classifier is to separate classes with a decision boundary so that that a point which lies on either side of the boundary is classified into its proper class. But SVM not only finds this boundary but also tries to keep a good gap between the classes so that we get a boundary which generalizes well for the dataset.

Figure 3: Loss function for SVM

$$L_i = \sum_{j \neq y_i} \max(0, w_j^T x_i - w_{y_i}^T x_i + \Delta)$$

It works by calculating the score of each class for an image by multiplying weight vector (w) with feature vector (x). After that it tries to keep the score of the true class above scores of wrong classes plus a threshold (triangle in figure 3). This threshold produces a gap in the decision boundary. It uses **hinge loss**. This formulation of Multiclass SVM follows the Weston-Watkins formulation (1999) [9].We have tested multiclass SVM with two variations in data which are the original pixel vector and the HOG feature vector.

D. *Softmax Classifier :*

Softmax predicts the probability of all the classes given an image and these probabilities sum to one. The class with the highest probability is the predicted class. It is a multiclass version of Logistic Regression. In machine learning one of the most widely used supervised classification algorithms is Softmax which can be used for classification and regression task. Softmax has been applied in various fields like the field of pattern recognitions, face recognition, text recognition and so on. The experimental results of Softmax show similar performance to SVM. Soo in this part we utilize Softmax algorithm for classification. We have tested Softmax with two variations of data which are original pixel vector and HOG feature vector to correctly test its performance. The HOG feature vector is of size 1x1296 for a single image.

Figure 4: Probablity of a class in Softmax

$$S_{i,j} = \frac{e^{z_{i,j}}}{\sum_{l=1}^{L} e^{z_{i,l}}}$$

Figure 5: Loss function for Softmax

$$L_i = -\log(\hat{y}_{i,k}) \quad \textit{where } \mathbf{k} \textit{ is an index of "true" probability}$$

This function is also called as **Cross entropy Loss**. y hat is the predicted probability of the model. k here refers to the class which is the true class of that example. Soo we can calculate the loss by taking negative log of predicted probability of true class and sum over the batch. From a probability point of view we are minimizing negative log likelihood of the true class which also means we are trying the maximize probability of true class. The function takes shape of a curve going downwards. It nears infinity at 0 and it nears 0 at 1 that is if we predict the correct class with high probability the loss would be nearly zero whereas if we predict true class with less probability the loss would be very high.

E. *Training - Mini-Batch Gradient Descent:*

Gradient descent is an optimization technique used for minimizing loss functions using their gradients. The procedure of repeatedly evaluating the gradient and then performing a

Parameter update is called Gradient Descent. It is one of the best techniques for optimizations and has wide applications. It is at the core of neural networks. In gradient descent what we basically do in each step is update the weight(w) matrix in such a way that loss decreases. In this manner with some number of iterations the loss is minimized. The process is that we subtract weight with (step size) x (gradient of loss function) at each iteration. In Mini-batch Gradient descent we compute gradient over small batches of training set in each iteration such as batch of 200 images. This is computationally less expensive than Stochastic Gradient descent in which gradient is computed over whole dataset on every iteration also less time consuming. The samples of say 200 examples are selected randomly from the dataset in every iteration because it is proven that this performs better than going over whole dataset in order.

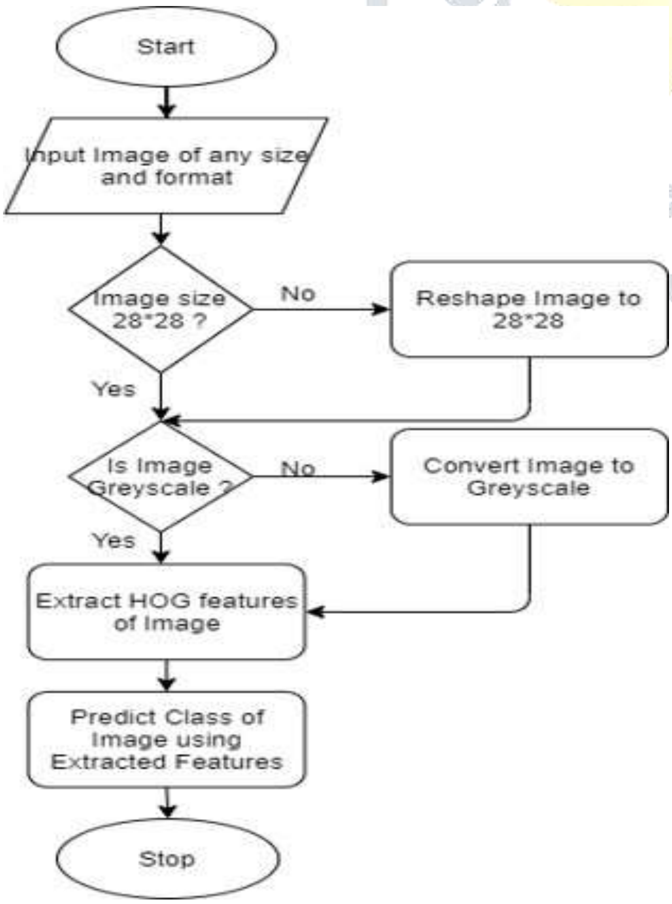
*F. **Validation Testing:***

This is an important step. Here we first divide the training set further into training and validation set. We train the model on training set and validate it on validation set. In this step we try out various parameter values such as regularization, learning rate, batch size etc. The final parameters are selected based on which parameters produce highest accuracy on validation set.

*G. **Testing:***

The final model is tested against the test dataset of F-MNIST. This data used for testing is never been seen by the model therefore this is the best way for evaluating the model. The results of various models can be seen ahead in results and accuracy part.

Figure 6: Workflow Diagram for classifying an image



IV. RESULTS AND ACCURACY:

Fashion-MNIST dataset consists of 70,000 images of various fashion items. Out of which 60,000 images are used for training and 10,000 images for testing. It consists of 10 types of fashion items which are:

Figure 6: Class Labels

Labels	*Description*
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

For evaluation we calculate various accuracies which are – SVM,, Softmax, ,SVM + HOG, Softmax + HOG and some accuracy results from literature. We are able to achieve an highest accuracy of 88% using Softmax + HOG and 2nd highest of 87% using SVM + HOG.

Table 1: Accuracy of various models

METHOD	*ACCURACY %*
Random Forest Classifier	*82*
Decision Tree	*79*
SGD Classifier	*81*
Linear SVC	*75*
SVM	*81*
Softmax	*84*
SVM + HOG	*88.3*
Softmax + HOG	*88.1*

Rather than only checking how many values were predicted right and how many wrong for checking accuracy there is a better way which is **Confusion Matrix**. The confusion matrix is a matrix which gives the number of True Positive, True Negative, False Positive and False Negative. Where row 0 and column 1 represents the images whose actual label was 0 but were predicted as 1. It gives a better understanding of accuracy. We have also converted it into heatmap so that it is easier to read.

Table 2: Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	True Negative	False Positive
Actual Positive	False Negative	True Positive

Table 3: Confusion Matrix for SVM + HOG

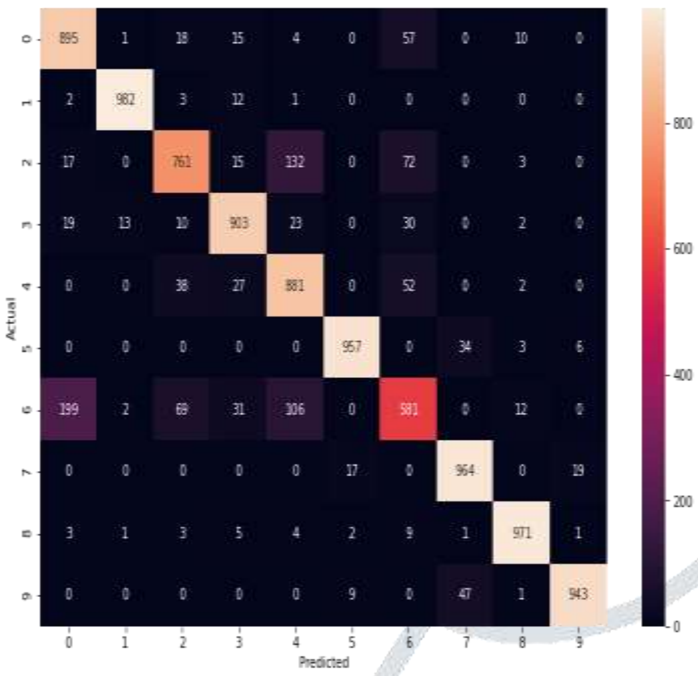
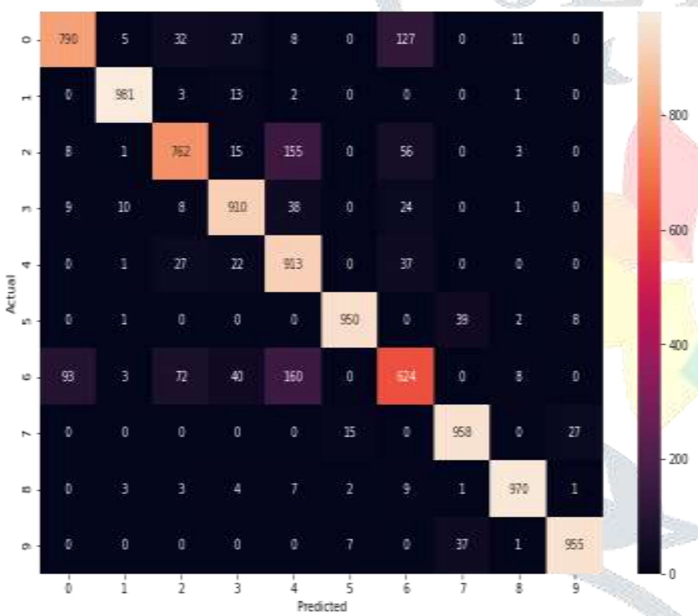


Table 4: Confusion Matrix for Softmax + HOG



As it can be seen we have managed to attain maximum accuracy of 88.3% with SVM + HOG. It can also be seen that by using HOG there is about 7% increase in accuracy of SVM and 5% increase in accuracy of Softmax. Also it can be seen that softmax alone performs better then SVM by about 3% but with HOG they have similar results. Looking at the confusion matrix we can see that SVM has difficulty identifying Shirt and Pullover while Softmax has difficulty identifying tshirt, shirt and Pullover.

V. Conclusion:

We can conclude that we have provided an efficient way for apparel classification with an accuracy of 88% using HOG+SVM. Our model can be used in various applications which are listed in the paper. Our model can be integrated in any system by following the workflow diagram. These weights can also be used in neural network for transfer learning because the model is pretrained. In future, many modifications and improvements can be proposed on the pre-processing part and feature extraction and more combinations of features can be explored.

VI. Acknowledgment:

We express our deep gratitude to our project guide Prof. Vijayalakshmi Kadroli for providing timely assistant to our query and guidance that she gave owning to her experience in this field for past many year. She had indeed been a lighthouse for us in this journey.

VII. References:

[1] https://www.ijitee.org/wp-content/uploads/papers/v8i5/E3075038519.pdf
[2] https://www.learnopencv.com/histogram-of-oriented-gradients/
[3] https://cs231n.github.io/optimization-1/
[4] https://cs231n.github.io/linear-classify/#svm https://github.com/jermwatt/machine_learning_refined.org/wiki/Regression_analysis
[5] https://arxiv.org/pdf/cs/0205070.pdf
[6] https://www.aclweb.org/anthology/P12-3020.pdf
[7] https://www.oberlo.in/blog/twitter-statistics
[8] J Weston and C Watkins. Support vector machines for multi-class pattern recognition. In Proc. 7th European Symposium on Artificial Neural Networks, 1999.
[9] https://www.aclweb.org/anthology/W02-1011.pdf
[10] https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2857/3251
[11] J. Zhao and X. Gui, "Comparison research on text pre-processing methods on twitter sentiment analysis," .
[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.
[13] https://cs231n.github.io/optimization-1/
[14] Saif H, Fernandez M, He Y, & Alani H. "On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter". In Proc. 9th Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland, pp.80-81, 2014.
[15] M. Wongkar and A. Angdresey, "Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter," 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICIC47613.2019.8985884.
[16] L. Mandloi and R. Patel, "Twitter Sentiments Analysis Using Machine Learninig Methods," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5, doi: 10.1109/INCET49848.2020.9154183.
[17] V. Prakruthi, D. Sindhu and D. S. Anupama Kumar, "Real Time Sentiment Analysis Of Twitter Posts," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 29-34, doi: 10.1109/ Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 208-211, doi: 10.1109/ICECA.2018.8474783.
[18] R. I. Permatasari, M. A. Fauzi, P. P. Adikara and E. D. L. Sari, "Twitter Sentiment Analysis of Movie Reviews using Ensemble Features Based Naïve Bayes," 2018 International Conference on Sustainable Information Engineering and Technology

(SIET), Malang, Indonesia, 2018, pp. 92-95, doi: 10.1109/SIET.2018.8693195.

[19] Pak and P. Patrick. "Twitter as a corpus for sentiment analysis and opinion mining." LREc, vol. 10, no. 2010, pp. 1320-1326. 2010.

[20] Fauzi, M. Ali, and Anny Yuniarti. "Ensemble Method for Indonesian Twitter Hate Speech Detection." Indonesian Journal of Electrical Engineering and Computer Science, vol. 11, no. 1, pp. 294-299. 2018.

[21] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in Proceedings of the 52nd Annual Meeting of the Association for CSITSS.2018.8768774