International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT183821

Multiclass Classification Using Random Forest Classifier

Sermista Talla*, Pavani Venigalla, Ashmitha Shaik, Meghana Vuyyuru
Computer Science and Engineering Department, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

ABSTRACT
A multiclass classification using Random Forest Classifier is proposed in this paper. The Random forest classifier is commonly used for solving the multiclass classification tasks in machine learning. The Random forest classifier predicts a valid classification results with minimum training time when compared to other classifier algorithms. For pattern classification tasks on an image dataset are performed in order to evaluate the performance of the proposed classifier. First, we consider some sample datasets of flower species and birds. Second, we train the object class models by using datasets to develop the random forest classifier for image classification. Third, we test the object class models and find the accuracy .In this we use Haralick,Hu moments to extract global features to quantify an image. In the experiments we test our method using FLOWERS-17,CUB_200_2011 dataset and compare the results with other methods like logistic regression, k-nearest neighbour, support vector machine.

Keywords : Random Forest Classifier, Image Classification, Haralick, Hu Moments, Histogram.

I. INTRODUCTION

The image classification task is one of the ongoing important topics in various computer vision tasks. The ability of a machine learning model to classify or label an image into its respective class with the help of learned features from hundreds of images is called as Image Classification. Multiclass Classification refers to the task of extracting one or more classes. Automatic processing of these images requires efficient classification techniques. In general automatic data classification tasks require two processes:an appropriate feature extraction process and an accurate classifier model process.

For image classification tasks, a feature extraction process can be considered on the basis of information provided in an image. Features are the information or list of numbers that are extracted from an image. These can called as real-valued numbers (integers, float or binary). There are a wider range of feature extraction algorithms in Computer Vision.

When deciding about the features that could quantify plants and flowers, we could possibly think of Color, Texture and Shape as the primary ones. This is an obvious choice to globally quantify and represent the plant or flower image.But this approach is less likely to produce good results, if we choose only one feature vector, as these species have many attributes in common like sunflower will be similar to daffodil in terms of color and so on.

So, we need to quantify the image by combining different feature descriptors so that it describes the image more effectively.The global feature descriptors are the feature descriptors that quantifies an image globally. These don't have the concept of interest points and thus, takes in the entire image for processing.These are the feature descriptors that quantifies local regions of an image. Interest points are determined in the entire image and image patches/regions surrounding those interest points are considered for analysis.

II. METHODS AND MATERIAL

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

The pseudocode for random forest algorithm can split into two stages.

1. **Random forest** creation pseudocode.
2. **Pseudocode to perform prediction** from the created random forest classifier.

Random Forest pseudocode:

- Randomly select **"k"** features from total **"m"** features.
- Where **k << m**
- Among the **"k"** features, calculate the node **"d"** using the best split point.
- Split the node into **daughter nodes** using the **best split**.
- Repeat **1 to 3** steps until "l" number of nodes has been reached.
- Build forest by repeating steps **1 to 4** for "n" number times to create **"n" number of trees**.

Random forest prediction pseudocode:

- Takes the **test features** and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
- Calculate the **votes** for each predicted target.
- Consider the **high voted** predicted target as the **final prediction** from the random forest algorithm.

Feature Selection

Most researchers apply standard feature selection in their approach to improve computational performance with a handful using more sophisticated approaches.

Types:

We have two different types of feature descriptors, they are local and global. In this paper, we just concatenate each feature vector to form a single global feature vector.

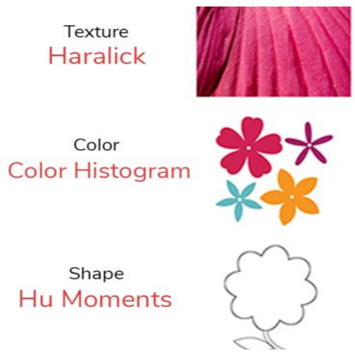

Figure 1 : Global Features to quantify a flower image

Functions for global feature descriptors

1.**Hu Moments**

To extract Hu Moments features from the image, we use cv2.HuMoments() function provided by OpenCV. The argument to this function is the moments of the image cv2.moments()flatenned. It means we compute the moments of the image and convert it to a vector using flatten(). Before doing that, we convert our color image into a grayscale image as moments expect images to be grayscale.

2.Haralick Textures

To extract Haralick Texture features from the image, we make use of mahotas library. The function which we will be using here is mahotas.features.haralick(). Before that, we convert our color image into a grayscale image as haralick feature descriptor expect images to be grayscale.

3.Color Histogram

To extract Color Histogram features from the image, we use cv2.calcHist() function provided by OpenCV. The arguments it expects are the image, channels, mask, histSize (bins) and ranges for each channel [typically 0-256). We then normalize the histogram using normalize() function of OpenCV and return a flattened version of this normalized matrix using flatten().

After extracting, concatenating and saving global features and labels from our training dataset, now we should train our system. To do that, we need to create our Machine Learning models. For creating our machine learning model's, we take the help of scikit-learn.

III. III. RESULTS AND DISCUSSION

Here the train_test_split function provided by scikit-learn is used to split our training dataset into train_data and test_data. By this way, we train the models with the train_data and test the trained model with the unseen test_data. The split size is decided by the test_size parameter.

Here we used the technique called K-Fold Cross Validation method , a model-validation technique which is the best way to predict ML model's accuracy. In short, if we choose K = 10, then we split the entire data into 9 parts for training and 1 part for testing uniquely over each round upto 10 times. Random Forest (RF) gives the maximum accuracy of **65.57%**. This is mainly due to the number of images we use per class. We need large amounts of data to get better accuracy. For example, for a single class, we atleast need around 500-1000 images which is indeed a time-consuming task.

The following diagram shows us about the machine learning algorithm comparison chart with the various classifiers and makes the comparison with each one of them, to know which classifier gives us the better results.

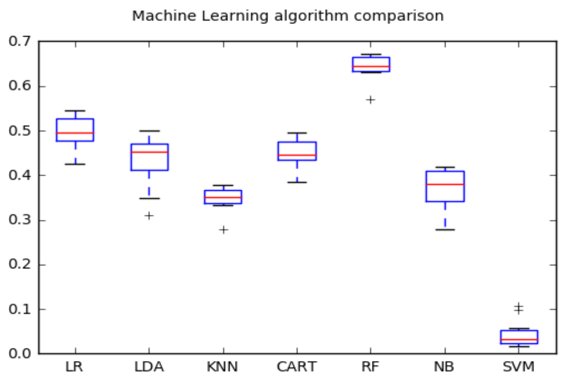

Figure 2 : Comparison chart of different machine learning classifiers used (Y-axis: Accuracy)

After training each our machine learning model, then we check the cross validation results. The results of classification can be seen in the below figures.

By using the below models, we predicted the flower and bird for which we got the correct results.

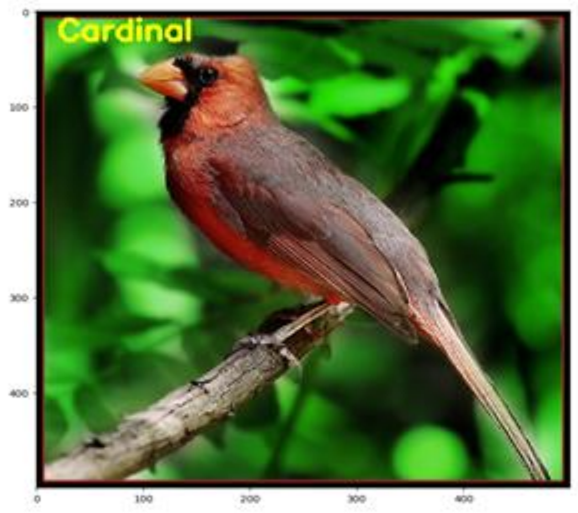

Figure 3 : Prediction 1-Cardinal(correct)

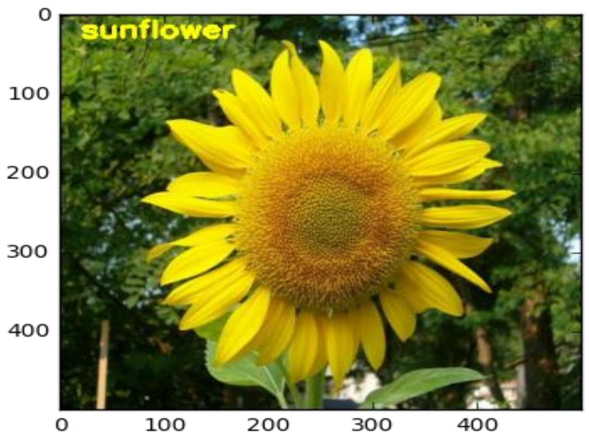

Figure 4 : Prediction 2-Sunflower(correct)

As we can see, our approach seems to do pretty good at recognizing flowers. But it also predicted wrong label like the last one. Instead of sunflower, our model predicted buttercup.

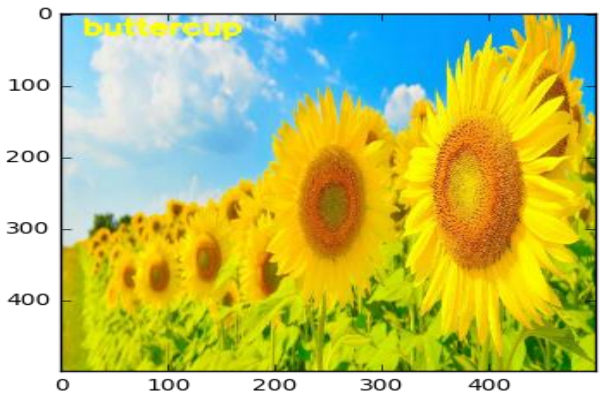

Figure 5 : Prediction 3-Buttercup(Wrong)

IV. CONCLUSION

A classification model for image data is proposed by using Random Forest classifier is proposed in this paper. Since the Random Forest classifier does not require any excessive training procedure commonly required in most of the artificial neural network architectures, the resulting classifier can yield an appropriate classification decision with very limited computational efforts. The proposed classifier utilizes the Random Forest classifier for minimizing the training time over the conventional classifiers while yielding accurate classification results by adopting the advantage of taking making many decision trees into consideration. In order to evaluate the performance of the proposed classifier, experiments on Flower-17 and CUB_200_2011 image data sets are carried out. The performance of the proposed classifier is compared with those conventional classifiers such as KNN, RFC, LDA and SVM in terms of training speed and classification accuracy. The advantage of training speed of the proposed classifier over the conventional classifiers is an advantageous feature in practical applications. Further research on how to overcome the assumption of the independence of the individual feature dimension in the Random Forest classifier will be a subject in future research.

V. REFERENCES

[1]. D. Lowd, P. Domingos, "Naive Bayes models for probability estimation," in proc. of the 22th International Conference on Machine Learning, 2005, pp. 529-536.

[2]. D. Lewis, "Naive Bayes at forty: The independence assumption in information retrieval," Lecture Notes in Computer Science, vol. 1398, pp. 4-15, June 2005.

[3]. G. Strang. "The discrete cosine transform", SIAM Rev, vol. 41, no. 1, pp. 135-147, 1999

[4]. Z. Guo, L. Zhang, D. Zhang. "A completed modeling of local binary pattern operator for texture classification", IEEE Transaction on Image Processing, vol. 19, no.6, pp.1657-1663, March 2010.

[5]. O. Tuzel, F. Porikli, P. Meer. "Region covariance: A fast descriptor for detection and classification", in proc. of Ninth European Conf. Computer Vision, vol. 2, 2006, pp. 589-600.

[6]. A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In Proc. ICCV, 2007.

[7]. N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In Proc. CVPR, volume 2, pages 886–893,2005

[8]. T. Saitoh, K. Aoki, and T. Kaneko. Automatic recognition of blooming flowers. In Proc. ICPR, volume 1, pages 27–30, 2004