# Natural Image Classification

Aditya Pulekar
Computer Science Graduate student at Rochester Institute of Technology, New York
acp8882@rit.edu

Chirag Kular
Computer Science Graduate student at Rochester Institute of Technology, New York
ck4957@rit.edu

## ABSTRACT

Our project aims at classifying the images into three categories, namely, airplanes, cars and motorcycles using an Artificial Neural Network. Similar to humans, machines, too, require to analyze the features of an image to determine its content. Hence, for helping the system with the classification, we used algorithms such as Harris Corner Detection for detection of interest points in an image and consequently applied algorithms such as Histogram of Oriented Gradients (HoG), Speeded-Up Robust Features (SURF) and Daubechies D4 Wavelet Transform to generate the subsequent feature vectors. Our project utilized 160 out of the 500 images of Cars, 160 out of the 800 images of Motorcycles and 160 out of the 900 images of airplanes for training purposes. For the testing purposes, we selected 30 images from each of the categories which were not selected for the training. The system takes a test image as an input, classifies it into one of the three categories and annotates it.

## Categories and Subject Descriptors

I.4.m [**Image Processing and Computer Vision**]: Miscellaneous; D.0 [**Software**]: General;

## General Terms

Image Classification, Artificial Neural Network

## Keywords

Histogram of Gradients, SURF, Daubechies Wavelet Transform, Back propagation, Artificial Neural Network, Feature Extraction

## 1. INTRODUCTION

In today's world which have billions of images and to search for a specific image is a tedious task. There are timestamps associated with the images, but it is impossible for humans to remember the time when a particular image is created or clicked. So to make our life easier, we have classification algorithms in the computer vision which have the ability to identify different characteristics and extract features which in turn help in classifying the images. These classification algorithms combine with the machine learning algorithms helps us to search from more quickly from a image database. It works in two step process where the first step is training in which a training class is created based on image features and the second step is testing. In this paper, with the help artificial neural network we will train them with three categories of image data set and use feature extract algorithms to classify the images into one of these category and train the ANN accordingly.

## 2. IMPLEMENTATION

### 2.1 Daubechies D4 Wavelet Transform

A particular type of discrete wavelet transform called the Daubechies Wavelet Transform was invented by Ingrid Daubechies which was named after him. It consists of four scaling and wavelet function coefficients. The scaling function coefficients are as follows:

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \quad (1)$$

And the wavelet function coefficients are as follows:

$$g_0 = h_3, g_1 = -h_2, g_2 = h_1, g_3 = -h_0 \quad (2)$$

The scaling function is applied to the incoming N data values and computes the N/2 smoothed values. The N element input vector stores this smoothed values in the lower half of the input. While the upper half of the input stores the N/2 differences values obtained by applying the wavelet function. Multiply the four coefficients and data values to compute the scaling and wavelet functions. The functions are as follows:

**Scaling Function:**

$p[k] = h_0 * signal[2k] + h_1 * signal[2k+1] + h_2 * signal[2k+2] + h_3 * signal[2k+3]$

**Wavelet Function:**

$q[k] = g_0 * signal[2k] + g_1 * signal[2k+1] + g_2 * signal[2k+2] + g_3 * signal[2k+3]$

For each step, the wavelet transform calculates wavelet and scaling function and it is repeated after every two iterations.

Algorithm for Daubechies D4 Wavelet Transform is as follows:

1. Initialize four scaling coefficients (h0, h1, h2, h3) and four wavelet coefficients (g0, g1, g2, g3).
2. Iterate over the rows in the image matrix.

3. Initialize a temporary array of the same length as the row dimension of the image matrix.

4. Apply the scaling function to the data values of the current row and store the resulting values in the first (lower) half of the temporary array.

5. Apply the wavelet function to the data values of the current row and store the resulting values in the second (upper) half of the temporary array.

6. Replace the intensity values of the current row with the temporary array values.

7. Repeat steps 2 to 6 unless all the rows in the input image matrix are transformed.

## 2.2 Harris Corner Detection

Corner detection is one of the approach used in computer vision for certain kind of feature extraction from the image. This corner detection and interest point detection are the two topics which goes hand in hand. When two edges moving in different direction are intersected, the resultant point is defined as corner. The most simplified approach for corner detection is correlation however this approach is expensive in terms of computational power. The first algorithm for corner detection was given by Moravec where he tested each pixel in the image to check whether it is a corner. Moreover, this method was improved by Harris and Stephen. They performed the non-maximum suppression on the points which gave larger corner response in the computed matrix of sub-window of image. The corner response is measured by auto-correlation of a matrix.

## 2.3 Histogram of oriented Gradients (HoG)

HoG is one of the feature descriptor in computer vision for object detection which was proposed by Navneet Dalal and Bill Triggs. It is very similar to SIFT descriptor however, the inner calculation on grid of uniformly spaced cells is done differently for better accuracy. In HoG, the distribution of intensity gradients is described on the basis of local object appearance and shape within an image. A histogram of gradient is calculated for the pixels in a cell which are comprised of small connected regions in a image. It is mostly suited for detection of humans in images.

HoG Algorithm:

1. **Gradient Computation:** Ensure that in image preprocessing, calculation of color and gamma values are normalized. However, the impact of this step is very little on performance. On the other hand, gradient values can be calculated as the first step. This is done by applying one dimensional mask in horizontal and vertical directions. Also, image's color and intensity data needs filtering.

2. **Orientation binning:** The second step involves the creation of cell histograms. For an orientation based histogram, each pixel within the cell does a weighted vote based on gradient values. These cells can take either a rectangular shape or radial shape and depending on signed or unsigned gradient, histogram channels are evenly spread over 0 to 180 degrees.

3. **Descriptor block formation:** For the changes in illumination and contrast, we locally normalize the gradient strengths for which large cell grouping is required to form blocks and these blocks needs to be connected spatially. From all of these block regions, the components of the normalized cell histograms are concatenated to form the vector known as HoG descriptor. These blocks occur multiple times in the resultant descriptor and are basically of rectangular or circular blocks. The rectangular block is represented by number of cells per block, number of pixel per cell and number of channels per histogram which are the three parameters. For the detection of human, the most optimal parameters consist of four cells in each blocks forming nine histogram channels.

4. **Block Normalization:** All the blocks from the previous step needs to be normalized. Dalal and Triggs worked on four different methods to do the normalization since it improved the performance over the non-normalized data.

## 2.4 Speeded-Up Robust Features (SURF)

It is another local feature detector and descriptor in the computer vision for the object recognition or classification and it takes the concept from the SIFT. The SURF algorithm works on principle of SIFT however their step details are different.

SURF Algorithm:

**Interest Point detection:** In this approach, the gaussian smoothing is approximated by the square-shaped filters. The process of filtering the integral image with a square is quite faster. To detect interest points, a Hessian Matrix based blob detector is used and local changes around these points are measured by the determinant of the Hessian matrix. Points with the maximal determinant value are chosen and these values are also considered in scale selection.

**Scale space representation and location of interest points:** The detected interest points can be found at different scales compared to images which we see at different scales. So SURF basically divides these scale space into octaves which are response map series covering scale doubling and different size of box filters helps in implementing the scale spaces. Application of Non-maximum suppression in a neighborhood helps in localizing the interest points in the image.

**Local neighborhood descriptor:** The descriptor's job is to find image's unique and robust feature description. Every interest points identified has a description and their dimensionality has an effect on point-matching accuracy and its computational complexity. A short descriptor has a advantage of giving robust feature and on the other hand have a disadvantage of giving false value resulting in insufficient discrimination. So we try to fix this false value based on data around the interest point. After that we build a square region and SURF descriptor are extracted from it.

**Matching:** The matching pairs can be found based on the descriptors obtained from different images.

## 3. ARTIFICIAL NEURAL NETWORK

### 3.1 Back Propagation Network (BPN)

For the training of artificial neural networks, the method most commonly used in parallel with optimization method is known as backward propagation of errors (also known as Backpropagation). The gradient value is computed by this method for a loss function w.r.t all the weight given in the network. Backpropagation falls under supervised learning since we need a particular output for a given input that in turn will calculate the loss function gradient. The Backpropagation algorithm is partitioned in two stages viz Propagation and Updation of Weights:

**Stage 1: Propagation**

A. The input of a desired pattern is passed through neural network to give output activations. This is basically a forward propagation.

B. The output from the previous step is propagated through neural network to compute the differences between desired and actual output (i.e. deltas). This is backward propagation.

**Stage 2: Updation of Weights**

A. Input activation and the deltas of output of each weight are multiplied to calculate the gradient of the weight

B. The gradient ratio is subtracted from the weight and this ratio is called learning rate. The training of neuron is much faster if the ratio is high and when the ratio is low, the training is more accurate. The above two stages are repeated to improve the performance of the neural network.

## 4. RESULTS

Our database consisted of the following image categories:

| Image Category | Image Type | No. of training images | No. of testing images |
|---|---|---|---|
| Category 1 | Car | 160 | 30 |
| Category 2 | Motorcycle | 160 | 30 |
| Category 3 | Air-Plane | 160 | 30 |

Table 1.1 All Image categories

The classification accuracy of an artificial neural network is dependent upon the quality of its training. Training of an artificial neural network is achieved by feeding it the feature vectors corresponding to each image that is used for its training. Training of our back propagation network (a multi-layer perceptron artificial neural network) was achieved by generating feature vectors for training images using algorithms such as Histogram of Oriented Gradients (HoG), Daubechies D4 Wavelet Transform and SpeededUp Robust Features (SURF).

### 4.1 Training and Testing with HoG and SURF features (together)

Let us look at the performance of our Back Propagation Network (BPN) with the images used during training (160 for each category).

| | Car | Motorcycle | Airplane | Accuracy (%) |
|---|---|---|---|---|
| **Car** | 121 | 21 | 18 | 75.63 |
| **Motorcycle** | 19 | 131 | 10 | 81.88 |
| **Airplane** | 23 | 18 | 119 | 74.38 |

Table 1.2 Performance Matrix for BPN (with training images)

In the above performance matrix, the diagonal elements of the matrix determine the accuracy percentage for every category. For example, in case of the first category (car), out of 160 images, 91 images containing cars were correctly classified and the rest 69 images were incorrectly classified to contain either a motorcycle or an airplane. Hence, the net classification accuracy for category 1 was 56.88%. Similarly, the classification accuracy of the back propagation network for the training images was calculated for other categories. The net training accuracy was evaluated to be 77.30%.

We also tested our system with images, belonging to the same three categories, which were not used during the training of the BPN. Let us look at the performance results of the BPN for testing images.

| | Car | Motorcycle | Airplane | Accuracy (%) |
|---|---|---|---|---|
| **Car** | 12 | 7 | 11 | 40.00 |
| **Motorcycle** | 7 | 17 | 6 | 56.67 |
| **Airplane** | 12 | 6 | 12 | 40.00 |

Table 1.3 Performance Matrix for BPN (with testing images)

As it can be seen from table 1.3, the accuracy for each of the categories dipped since the BPN was not trained with a very large dataset, covering diverse images, for each category. The accuracy for category 1 and 3 (i.e. car and airplane) was closer to each other since the training dataset for both the categories consisted of images with similar texture backgrounds. Unlike the images belonging to category 2 (motorcycle), category 1 and 3 images had a more natural setting as their backgrounds. Majority of the category 2 images in the training set had a white (non-natural) background thereby differentiating the motorcycles in the images in the best possible way. The net testing accuracy was evaluated to be 45.56%.

## 4.2 Training and Testing with Daubechies D4 Wavelet Transform features

The classification accuracy of the BPN reduced slightly on using the Daubechies Wavelet Transform to generate feature vectors for the images. Let us look at the performance of our Back Propagation Network (BPN) with the images used during training (160 for each category).

Table 1.4 Performance Matrix for BPN (with training images)

| | Car | Motor -cycle | Airplane | Accuracy (%) |
|---|---|---|---|---|
| **Car** | 118 | 14 | 28 | 73.75 |
| **Motorcycle** | 17 | 121 | 21 | 75.62 |
| **Airplane** | 29 | 17 | 114 | 71.25 |

Table 1.5 Performance Matrix for BPN (with training images)

The net training accuracy for this method was 73.54%.

Similar to testing with the HoG and SURF features, we also used unseen testing images for the BPN in case of Daubechies D4 Wavelet Transform as well. Let us look at the performance results of the BPN for testing images.

| | Car | Motor -cycle | Airplane | Accuracy (%) |
|---|---|---|---|---|
| **Car** | 12 | 4 | 14 | 40.00 |
| **Motorcycle** | 10 | 13 | 7 | 43.33 |
| **Airplane** | 12 | 8 | 10 | 33.33 |

Table 1.6 Performance Matrix for BPN (with testing images)

As it can be seen from the table 1.6, unlike HoG and SURF Features, Daubechies D4 Wavelet transform tends to classify more of the category 1 images as category 3 images and vice versa in case of test images. Hence, Daubechies D4 Wavelet Transform does not represent a robust alternative to HoG and SURF features. The net testing accuracy for this method was 38.89%.

## 5. CONCLUSION

From this paper, we can conclude that using Histogram of Oriented Gradients and SURF features for training and testing a Back Propagation Network (BPN), with an image database of 480 images (160 images of each of the three categories) with lesscomplex backgrounds, represent a more robust option than Daubechies D4 Wavelet Transform features for a pairing with a back propagation network. The best natural image classification accuracy achieved for the training set was 77.30% and for the testing set was 45.56%. Both of the best classification accuracy percentages were obtained when the HoG and SURF feature vectors were used for training and testing purposes. It is to be noted that these observations have been recorded for a relatively small dataset. Moreover, the accuracy of the system can be increased by implementing a technique for wisely choosing from the obtained set of interest points in an image. Choosing such a limited set interest points which truly represent the interest points pertaining to the object can surely increase the training and testing accuracies for HoG and SURF features manifold.

## 6. ACKNOWLEDGMENTS

We would like to thank our professor Dr. Srinivas Sridharan for his technical guidance and support during the course of this project.

## 7. REFERENCES

[1] Girish Patil, Sonu Mourya, Karan Belsare and Shashikant Patil ”Wavelet Transform and Neural Network based Approach to Natural Image Classification” in International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128x Volume 5, Issue 4, April 2015.

[2] Szummer, Martin, and Rosalind W. Picard. ”Indoor-outdoor image classification.” Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on. IEEE, 1998.

[3] Vailaya, A., Figueiredo, M. A., Jain, A. K., & Zhang, H. J. (2001). Image classification for content-based indexing. Image Processing, IEEE Transactions on, 10(1), 117-130.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool: Speeded-Up Robust Features (SURF)

[5] Latex Source: https://www.overleaf.com/read/pfzfmprpxhph

[6] Dataset: http://www.robots.ox.ac.uk/~vgg/data/data-cats.html

[7] Speeded-Up Robust features (SURF): https://en.wikipedia.org/wiki/Speeded_up_robust_features

[8] Histogram of Oriented Gradients (HoG): https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients