University of Engineering & Technology, Lahore

Department of Electrical Engineering

Study of Trained ANN & Random Forest Networks for Image Classification using SIFT, SURF & HOG Techniques

---

Prepared & Designed by

Aziz Haider (2020-EE-172)

EE-439: Introduction to Machine Learning

Assignment Report || April 09, 2023

Contents of Report

Abstract

In this report, we investigate the problem of classifying seven different classes of images. The classes include *Cats, Dogs, and Horses*. To address this problem, we explore the use of feature extraction methods such as Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and Histogram of Oriented Gradients (HOG). We train two different classifiers, namely Artificial Neural Network (ANN) and Random Forest, using the extracted features.

For the feature extraction step, we implemented a **MATLAB** code to extract SIFT, SURF, and HOG features from the image dataset. The dataset was divided into a *70-30 ratio of training and test data*. SIFT, SURF, and HOG features were extracted for each class.

After the feature extraction step, we trained two different classifiers, *ANN* and *Random Forest*, using the extracted features in **Python** programming language. For both classifiers, we used the training data to train the model and the test data to evaluate its performance.

In conclusion, the proposed approach demonstrates the effectiveness of using feature extraction methods such as SIFT, SURF, and HOG, in combination with machine learning classifiers such as ANN and Random Forest, for classifying different classes of images.

Introduction

The purpose of feature extraction is to transform the raw data or image into a format that can be easily understood by machine learning algorithms. In image processing, feature extraction involves identifying and extracting relevant information or features from the image that can be used for classification, recognition, or other tasks.

Feature Extraction Methods

SIFT, SURF, and HOG are methods used to extract features from images. These features can be used to identify and match images. All three methods are good at detecting features in an image that are robust to changes in lighting and partial occlusions.

1. **SIFT (Scale-Invariant Feature Transform):** SIFT is a feature extraction method that is widely used in computer vision and image processing. SIFT extracts distinctive features from an image, which can be used to identify and match images. Keypoints are identified

by taking Difference of Gaussian (DoG). By taking convolution of the input image and the DoG, scale space is calculated. Lastly, keypoints are localized and orientations are assigned at each keypoint location based on image gradient.

2. **SURF (Speeded Up Robust Features):** SURF is an improved version of SIFT which uses integral images and it requires less computation. It is also scale-invariant, and it uses a faster approximation of the scale-space representation. SURF uses a Hessian matrix to detect interest points in the image and extracts features using Haar wavelet responses.
3. **HOG (Histogram of Oriented Gradients):** HOG extracts features based on the gradient orientation of the image. It works by dividing the image into small cells and calculating the gradient orientation within each cell. The gradient orientation is then quantized into a histogram, and the histograms are concatenated to form the final feature vector.

Materials and Methodology

Following section discusses the materials and methodology used in detail.

Data Collection

Data has been taken from kaggle.com/datasets/pavansanagapati/images-dataset which included image sets for seven different classes {bike, car, cat, horse, human, dog, flower}. For the problem at stake, we were asked to take only 3 classes i.e. cats, dogs and horses.

Segmentations

We have divided the data into 2 parts.

1. **Training Data** which is 70% of data provided. This will be used for training of the model.
2. **Test Data** which is 30% of data provided. This will be used to determine the accuracy of the trained model.

Class	Total Data Instances	Training Data Instances	Test Data Instances
Cat	202	141	61
Dog	202	141	61
Horse	202	141	61

Data Processing

MATLAB version R2022b was used for the purpose of Feature Extraction. Since Python could not be used for feature extraction as the built-in libraries aren't supported because they are patented and they couldn't be used. Only SIFT feature extraction method is available in Pyhton so MATLAB was decided as the final approach for all three i.e. SURF, SIFT, and HOG. All the images are resized to 512 x 512 pixels and gray-scaled in the loop before features are extracted. Training data was loaded into the system using *imageDatastore* command of MATLAB. Labels for all the images were converted to numeric index for model training and an empty 2D array was defined for features which made sure that all features were of same dimension.

```
% Load Training Data
trainData = imageDatastore('C:\Users\ApriZon\Desktop\ML Assignment\Training Data',
'IncludeSubfolders', true, 'LabelSource', 'foldernames');
labels = grp2idx(trainData.Labels);
num_images = length(trainData.Files); % set the total number of images
num_features = 1500; % set the fixed number of features to extract for each image
ExtractedFeatures= zeros(num_images, num_features); % Create empty 2D array
```

The built-in extractFeatures function, used below, will create a feature matrix in MATLAB, where each row represents an image from the *imageDatastore* object type, and each column represents the features extracted from the image. The number of columns in the feature matrix will depend on the type of feature extraction method used.

1. **SURF Feature Extraction:** Built-in function `detectSURFFeatures` was used in addition to `extractFeatures` to extract the features.

```
for i=1:num_images
    img = readimage(trainData, i);
    % Resize image to [512,512] and convert to grayscale
    img = imresize(rgb2gray(img), [512, 512]);
    % Detect and extract SURF features
    points = detectSURFFeatures(img);
    [features, valid_points] = extractFeatures(img, points);
    % Flatten features and adjust dimension if necessary
    features = features(:)';
    if length(features) > num_features
        features = features(1:num_features);
    elseif length(features) < num_features
        features = [features zeros(1, num_features-length(features))];
```

```
    end
    ExtractedFeatures(i,:) = features; % Store features in 2D array
end
```

2. **SIFT Feature Extraction:** Built-in function `detectSIFTFeatures` was used in addition to `extractFeatures` to extract the features.

```
for i=1:num_images
    img = readimage(trainData, i);
    % Resize image to [512,512] and convert to grayscale
    img = imresize(rgb2gray(img), [512, 512]);
    % Detect and extract SURF features
    points = detectSIFTFeatures(img);
    [features, valid_points] = extractFeatures(img, points);
    % Flatten features and adjust dimension if necessary
    features = features(:)';
    if length(features) > num_features
        features = features(1:num_features);
    elseif length(features) < num_features
        features = [features zeros(1, num_features-length(features))];
    end
    ExtractedFeatures(i,:) = features; % Store features in 2D array
end
```

3. **HOG Feature Extraction:** Built-in function `extractHOGFeatures` to extract the features.

```
for i=1:num_images
    img = readimage(trainData, i);
    % Resize image to [512,512] and convert to grayscale
    img = imresize(rgb2gray(img), [512, 512]);
    features = extractHOGFeatures(img);
    % Flatten features and adjust dimension if necessary
    features = features(:)';
    if length(features) > num_features
        features = features(1:num_features);
    elseif length(features) < num_features
        features = [features zeros(1, num_features-length(features))];
    end
    ExtractedFeatures(i,:) = features; % Store features in 2D array
end
```

The features extracted from each process along with their respective labels were exported using the following commands in *.mat* format. The following code is shown as to how the HOG features along with the labels were saved in another repository.

```
save('C:\Users\ApriZon\Desktop\ML Assignment\HOG\HOG_train.mat',
'ExtractedFeatures');
save('C:\Users\ApriZon\Desktop\ML Assignment\HOG\labels_train.mat', 'labels');
```

This approach was used to extract the features from the test data too. Test data features and their corresponding labels were saved in the folder along with the training data features extracted and labels.

Model Training

Python version 3.11.1 was used for model training using the features and labels extracted in the previous step. Following libraries were installed before the process started. Module 'numpy' was used for array processing & manipulation. Module '`scikit-learn`' was used for ANN (Artificial Neural Network) & Random Forest training and testing. Module 'scipy' was used to import the features and labels.

```
pip install scikit-learn
pip install numpy
pip install scipy
```

Once these libraries are installed in the system, import them so they can be used in the script.

```
import numpy as np
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
import scipy.io
```

Import the features & labels from the *.mat* created by MATLAB using 'scipy' module. This will create an array of the object type *ndarray* in python. The following code shows how the feature extracted from HOG were loaded in the memory.

```
mat = scipy.io.loadmat('C:/Users/ApriZon/Desktop/ML Assignment/HOG/HOG_train.mat')
X_train = mat['ExtractedFeatures']

mat = scipy.io.loadmat('C:/Users/ApriZon/Desktop/ML Assignment/HOG/labels_train.mat')
y_train = mat['labels'].ravel()
```

```
mat = scipy.io.loadmat('C:/Users/ApriZon/Desktop/ML Assignment/HOG/HOG_test.mat')
X_test = mat['ExtractedFeatures']

mat = scipy.io.loadmat('C:/Users/ApriZon/Desktop/ML Assignment/HOG/labels_test.mat')
y_test = mat['labels'].ravel()
```

1. Artificial Neural Network (ANN)

Define the ANN network using MPLClassifier, which is a sub module of '`scikit-learn`'. ANN model consists of two layers which are one hidden layer with 100 neurons and one output layer. By default 'solver=adam' and epochs were set to 5000.

```
ann = MLPClassifier(hidden_layer_sizes=(100,),max_iter=5000)
ann.fit(X_train, y_train)
```

Accuracy of the model can be defined by using testing unseen images on the model and comparing its predicted value with the true value.

```
accuracy = ann.score(X_test, y_test)
print("Accuracy of ANN on test set: {:.2%}".format(accuracy))
```

2. Random Forest (RF)

An instance of RandomForestClassifier, which is a part of the '`scikit-learn`' module, is defined and the parameters provided state that the random forest is built using 100 decision trees. Other parameters have been kept at default.

```
rf = RandomForestClassifier(n_estimators=100)
rf.fit(X_train, y_train)
```

After training the model, the accuracy of the model is evaluated on the test data using the *score* function of the random forest classifier. It calculates the mean accuracy of the predictions made by the model on the test data.

```
accuracy = rf.score(X_test, y_test)
print("Accuracy of Random Forest on test set: {:.2%}".format(accuracy))
```

Results

The results of the study showed that the effectiveness of different feature extraction methods varied when used with machine learning classifiers to classify different classes of images. Three different feature extraction methods were used, namely SIFT, SURF, and HOG, to extract features from images of three different classes, namely Cat, Dog, and Horse.

Different classifiers, namely Artificial Neural Network (ANN) and Random Forest, using the extracted features, were trained. For the ANN classifier, a 2-layer sigmoid model was trained and the following accuracies were obtained: 40.11% for SURF, 33.52% for SIFT, and 59.34% for HOG. For the Random Forest classifier, the following accuracies were obtained: 37.91% for SURF, 32.42% for SIFT, and 55.49% for HOG.

Training Model	Feature Extraction Method	Accuracy
ANN (2-layer sigmoid)	SURF	40.11%
	SIFT	33.52%
	HOG	59.34%
Random Forest	SURF	37.91%
	SIFT	32.42%
	HOG	55.49%

Conclusion

Our study demonstrates that the choice of feature extraction method is an important factor in achieving high accuracy in image classification tasks. In our study, we found that HOG feature extraction method performed the best when used with both ANN and Random Forest classifiers. This may be due to the ability of HOG to capture the shape and orientation of objects in an image, which is important for distinguishing between different classes of images.

References

S. Routray, A.K. Ray, and C. Mishra, "Analysis of Various Image Feature Extraction Methods against Noisy Image: SIFT, SURF and HOG," International Journal of Image, Graphics and Signal Processing, vol. 11, no. 8, pp. 35-42, 2019.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27-48.

Raza, S., Saeed, U., Nazir, M., & Bukhari, S. A. (2017). Image feature extraction using deep learning: A review.

Kim, K. I., Kim, T. H., & Savarese, S. (2007). 3D object recognition using multi-scale keypoints. International Journal of Computer Vision, 75(2), 303-317.