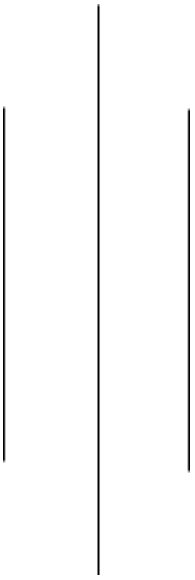
A
Report
on
Image Analysis and Computer Vision
Spring 2021
(CS 898BA)



Title: Fingerprint Spoof Detector Based on LBP and HOG

Submitted To
Dr. Ajita Rattani
(Course Instructor)

Submitted By
Suresh Pokharel
WSUID: X254W356
Department of Electrical Engineering and Computer Science
Wichita State University

23 March, 2021

Introduction:

This project uses a supervised machine learning algorithm called Support Vector Machine to classify the fingerprint images whether they are from real source or fabricated. The different two types of features: Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) features to train the model. Then, classification performance is compared using different evaluation metrics.

Objectives:

1. To develop a two-class Support Vector Machines (SVM) classifier to distinguish live fingerprints images from spoof samples using Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) features.
2. To compare the SVM classification performance between using LBP and HOG features.

Dataset:

The provided dataset has two classes of fingerprint images: live and spoof. Each class has 200 training and 200 test images.

Methods:

1. Import required libraries
2. Read all the tests and train images from the defined path separately and store them in the lists.
3. Choose the feature type: 'lbp' or 'hog'.
4. Convert images into HoG features or LBP features and flatten into 1 dimensional array.

   hog(live_img, feature_vector = True)

   This returns hog features of *live_img. feature_vectore* = *true* returns the features in 1D form.

   spoof_img = rgb2gray(spoof_img)
   spoof_train_features.append(local_binary_pattern(spoof_img, P, R, METHOD).flatten())

   To find LBP features, the image is converted into grayscale. Actually, this conversion was not needed as we already have grayscale. But, this is done to make the data compatible for further processes. After

that, P is the number of circularly symmetric neighbours and R is radius. These values are referenced from official documentation examples of scikit learn. Finally, the features are converted to flatten.

5. Assign the label ‘1’ for live images and ‘0’ for spoofed images.
6. Stack the features and label together, then shuffle.
7. Prepare train and test data with corresponding labels.
8. Train the SVM with default parameters of scikit learn.
9. Perform tests and evaluate.
10. Change the feature type in step 3 and repeat.

Results:

The results obtained from testing the model are given below. The results may slightly differ according to the values of hyperparameter considered.

Metrics	**SVM Performance**	
	HoG	**LBP**
Accuracy	0. 845	0.9225
Precision	0.8593	0.9303
Recall	0.845	0.9225

***