

ECE661: Homework 8

Fall 2020

Due Date: Oct 26, 2020 (11:59 PM)

Turn in typed solutions via BrightSpace. Additional instructions can be found at BrightSpace. For this homework you don't have any theory questions, however your analysis on your plots and/or results is important. Similar to HW7, you will be showing image classification results using different approach for a larger dataset.

1 Introduction

The different algorithms you have learned so far for measuring image texture are popular in the mainstream approaches to computer vision — that is, computer vision of the sort that is often encountered in robotic applications on earth and in space.

However, as you all know, during the last few years, deep-learning (DL) based approaches have taken on a life of their own. Typically, if the main focus of an application is image classification and you have very large image datasets available for training a classifier, these days you are likely to use a DL based approach for solving the problem.

The DL based algorithms have popularized a completely different approach to measuring texture — an approach that uses what is known as the Gram Matrix representation of a texture. The goal of this homework is to make you familiar with the idea of Gram matrices as used in this manner and doing so within the framework of mainstream computer vision.

That brings us to the definition of a Gram Matrix for Texture Characterization.

You construct a Gram Matrix for an image by convolving the image with C different convolutional operators, each of size $M \times M$. Typically, $M = 3$, and the floating-point weights in each convolutional operator are set randomly. So that normalization does not become an issue, you make sure that the weights in each operator add up to zero. Additionally, you make sure that each weight is drawn randomly from the interval $[-1, 1]$.

In what follows, we will refer to C as the *channels*.

Let's say that the size of the output with each channel is $K \times K$. You are allowed to downsample the output in each channel. Let's say your input images are of size 256×256 . Through downsampling, you could set $K = 16$,

implying that the output in each channel will be of size 16×16 . For the explanation that follows, I will assume that you set $K = 16$.

Next, you vectorize each channel output. That is, for each of the C outputs, you would represent it by a 256-element vector. Subsequently, you take inner products of all pairs of such C vectors.

The output of the above operation will be a symmetric matrix of size $C \times C$. This is known as the Gram Matrix representation of the texture. Since the matrix is symmetric, it is good enough if you retain just its upper triangular part.

Now you can represent each image by a point in a $C^2/2$ dimensional feature space. Using this feature space, you can again carry out image classification as you did for HW7. For the classification task, you can use Support Vector Machine (SVM) implementation that is available in OpenCV or scikit-learn python package.

2 Understanding the dataset

For this homework you will be using the outdoor multi-class weather data [1]. This dataset has a little over 1000 images divided into four weather categories. The four categories are: cloudy, rain, shine and sunrise. The last 10 images from each category are used to create the test set for this homework. You will show your final confusion matrix on these images. Some sample images are shown in Fig. 1. Note that the size varies from image to image, your implementation should handle this and convert the final feature maps to your required downsampled size, e.g, 16×16 .



Figure 1: Sample input images.

3 Notes

- Split the given training images into the training and validation sets. You can use 70 % - 30 % or 75 % - 25 % splitting criteria for training

and validation.

- Here is one recommended way to perform the classification task using Gram matrix. Run an outer ‘for’ loop for n trials, say $n = 100$. Adjust this parameter depending on the efficiency of your implementation and you can also include additional termination conditions. For each iteration, run the following steps
 1. Compute C different $K \times K$ feature maps using randomly generated convolution operators (downsampling is included in this step).
 2. Generate your Gram-matrix-based $C^2/2$ dimensional feature vectors for both training and validation images
 3. Train a SVM classifier using OpenCV or scikit-learn.
 4. Evaluate the classification accuracy on the validation set, check if the new features improve the accuracy.
 5. Save your best convolutional operators and SVM model for reproducibility.
- Show your confusion matrix for the test images using your best model parameters. Present your analysis by showing some false positives or false negatives, and your hypothesis for the misclassifications.
- If you’re using OpenCV’s SVM classifier, save your model in .xml file format.
- For scikit-learn use either ‘pickle’ or ‘joblib’ python module for storing your SVM model parameters.

3.1 Performance Measures

Similar to HW7, present your results in the form of confusion matrix. For reproducibility purpose, also include your convolutional operators that produced the best classification accuracy on your validation images. Also include the discussion on the model accuracy on training, validation, and test sets.

4 Extra Credit Task

Run your code from HW7 on this new dataset and present a comparative analysis of the results you obtain using your HW7 method and Gram matrix

based approach. Note that your observations and discussion on the accuracy and confusion matrices are critical for this task.

5 Submission Instructions

Include a typed report explaining how did you solve the given programming tasks.

1. Turn in a zipped file, it should include (a) a typed self-contained pdf report with source code and results and (b) source code files (.py, .cpp,.c) (c) your SVM model parameters in .xml or .pkl file format . Rename your .zip file as hw8_<First Name><Last Name>.zip and follow the same file naming convention for your pdf report too.
2. Your pdf must include a description of
 - Explain the Gram-matrix-based classification approach in your own words. How many features maps produced good classification results? (C parameter and SVM feature vector size)
 - Your plots or results should accompany supporting analysis, e.g., the effect on accuracy for the different values of C .
 - With OpenCV, you can store your best SVM parameter in .xml file format. For scikit-learn, you can store your model parameters in .pkl format
 - Your output confusion matrix with your observations on the overall accuracy.
 - Your source code. Make sure that your source code files are adequately commented and cleaned up.

References

- [1] Gbeminiyi Ajayi. Multi-class Weather Dataset for Image Classification. URL <http://dx.doi.org/10.17632/4drtyfjtfy.1>.