Machine Learning 1:
Programming Assignment (10 points)
(Due March 22, 2024)

Shen-Shyang Ho (Dr.)

March 1, 2024

1. We use the Fruits-360 dataset (https://github.com/Horea94/Fruit-Images-Dataset). There are a total of 131 classes (fruits/vegetables). The number of training images for each class ranges from 246 to 492. We will use classes with more than 400 training images.

   Each student will

   (a) be assigned 10 classes to work.

   (b) download images from folders **Training** and **Test** for the 10 classes to work on.

2. Since we have students from different disciplines, you can use **python, R, or Matlab**, etc.

3. **Traditional features construction**

   (a) Feature extraction from EACH image from the training data using image feature descriptor **SIFT (Scale-invariant feature transform)** (see https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html) with 128 dimensions for each keypoint (Note that you should obtain many keypoints from each image).

   - Plot the keypoints on one image from your training dataset (see Fig 1 for one such figure). (0.5 point)

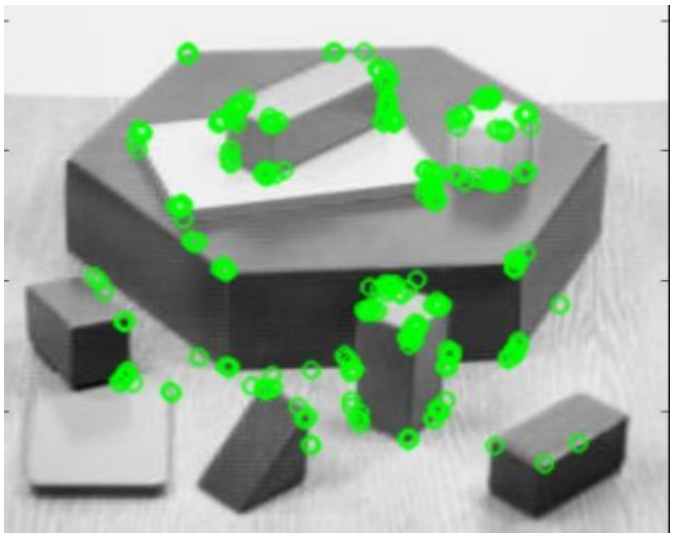

Figure 1:

(b) Create a new keypoints dataset KP that consists of all the keypoints from all the training images.

(c) Perform K-mean clustering such that $K = 100$ on KP.

(d) Use the learned K-mean clusters (see https://scikit-learn.org/stable/modules/clustering.html) to construct a 100-D vector for each image.

Example: For Image A, we have 20 keypoints in Cluster 1 and 10 keypoints in cluster 2, and no keypoints in the other clusters, then the vector representing Image A is

$$(20, 10, 0, 0, 0, \ldots, 0, 0)$$

(e) Create a new 100-D dataset D consisting of vectors constructed from the training images using Step (d).

(f) Dimensionality reduction (using Principal Component Analysis, PCA) (see https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html for PCA. https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_iris.html for code example.)

i. Perform Principal Component Analysis (PCA) dimensionality reduction Dataset D to 2 dimensions. (Note: You should not use the class labels)

ii. Plot the 2D points using 10 different colors/symbols for data from the 10 classes (see Figure 2 for an example of the plot without normalization).(1 point)

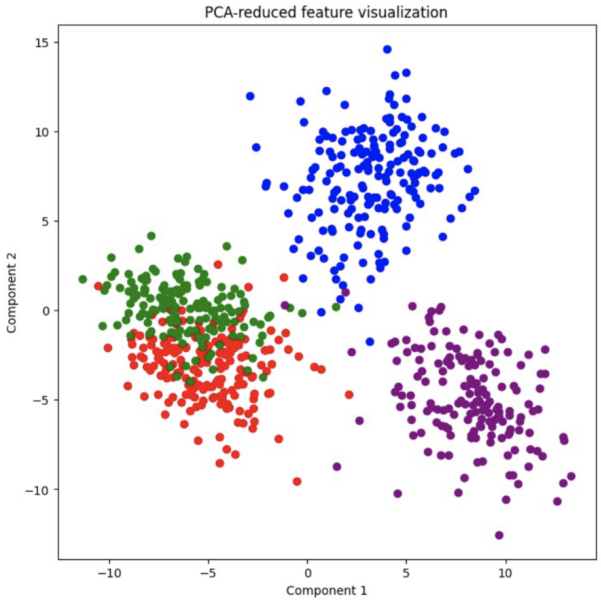

Figure 2:

4. **Processing Test Images**

For each test image

- perform keypoint extraction (3(a)).
- Use the 100 cluster centers obtained in 3(c) **based on training data** to obtain a 100D vector for the test image similar to 3(d).

All the processed test images will be the test dataset used in Question 5.

5. **"Traditional" Machine Learning Model - Support Vector Machine (SVM)**

   (a) We perform model selection to learn a linear SVM (see https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) prediction model (i.e., kernel = 'linear').

   (b) Train the SVM model using different C parameters. In particular, you will use the following values: 0.01, 0.1, 1.0, 10, 100.

   (c) Plot a graph to show the SVM performance on the training data and test data similar to Figure 3:

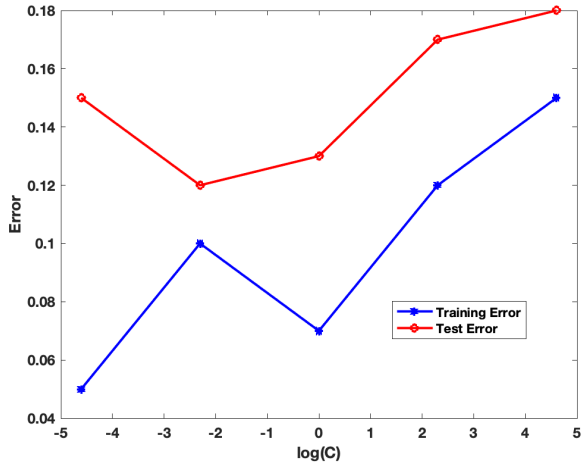

Figure 3:

   (d) Repeat Step (b) and (c) for kernel = 'rbf', 'poly', 'sigmoid'. (2 points - include 0.5 for linear kernel.)

   (e) Plot a graph to compare the best performance for SVM using the four kernels on the training data and test data similar to Figure 4 (note that you identify the best results for each kernel from results in Step (d)) (0.5 points):

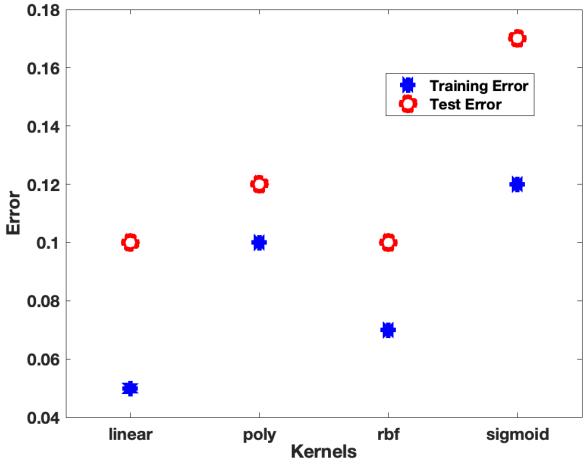

Figure 4:

For Question 6, 7, 8, the training and test sets will be the original image datasets.

6. **Deep Learning - Training a Simple Convolution Neural Network Model**

   Construct a 10-class prediction model using a convolutional neural network with the following simple architecture:

   i 2 Convolutional Layer with 8 3×3 filters.

   ii 1 max pooling with 2×2 pool size

   iii **Flatten** the Tensor

   iv 2 hidden layer with 16 nodes for fully connected neural network

   v Output layer has 10 nodes (since 10 classes) using 'softmax' activation function.

   (Use 'Relu' for all layers except the output layer.) for 20 epochs using 'adam' optimizer and 'categorical cross entropy' loss function. You can perform more epochs (> 20) if you want to. Use **Test data for validation**. For batch size, you can pick a size that will not slow down the training process on your machine. (see https://keras.io/examples/vision/mnist_convnet/)

   (a) Plot a graph to show the learning curves (i.e., x-axis: number of epochs; y-axis: training and validation accuracy - 2 curves) (1 points)

7. **Transfer Learning via Feature Extraction:**

   (a) Use the pre-trained "ResNet18" (freeze all layers, i.e., does not do training) and add a last layer (for training), train your 10-class prediction model for 10 epochs. (see https://pyimagesearch.com/2021/10/11/pytorch-transfer-learning-and-image-classification/) (0.5 point)

   (b) Plot a graph to show the learning curves (i.e., x-axis: number of epochs; y-axis: training and validation (use the test set) accuracy - 2 curves) (1 point)

   (c) We will use the features extracted from the last convolution layer of a "ResNet18" using "forward_hook" in PyTorch (https://pytorch.org/docs/stable/generated/torch.nn.modules.module.register_module_forward_hook.html) OR "feature_extraction" in TorchVision (https://pytorch.org/vision/stable/feature_extraction.html) for Step (d) and (e). (You can follow https://kozodoi.me/blog/20210527/extracting-features.)

   (d) Perform Step (c) on the training and test datasets.

   (e) Train a SVM prediction model using the features extracted (in Step (d)) from the training set using the kernel and C parameters that give the best test result in Question 4(e). What is the training and test accuracy? (1 point)

8. **Transfer Learning via Fine-Tuning: Training ResNet18 Using a PreTrained ResNet18 Model as the starting point**

   (a) Use the pre-trained ResNet18 as the starting model (initialization), train your 10-class prediction model for 10 epochs. (https://pyimagesearch.com/2021/10/11/pytorch-transfer-learning-and-

   (b) Plot a graph to show the learning curves (i.e., x-axis: number of epochs; y-axis: training and validation (use the test set) accuracy - 2 curves) (1 points)

9. **Conclusions:**

(a) For 6(a), 7(c), 8(b), what are the test accuracies (identify the epoch from your curves) before overfitting occurs? (0.5 point)

(b) For all the approaches you have tried in this assignment, which approach gives you the best test accuracy/error? (0.5 point)

(c) Are there any performance result(s) that surprise(s) you? Why? (0.5 point)