Final Project Title

Object Detection and Recognition

1. Introduction

An essential part of the behavior of humans is their ability to recognize objects. Humans are able to recognize large numbers of other humans, letters, digits, and so on.

The object recognition problem can be defined as a labeling problem based on models of known objects. Formally, given an image containing one or more objects of interest (and background) and a set of labels corresponding to a set of models known to the system, the system should assign correct labels to regions, or a set of regions, in the image.

2. Objective

The goal of this project is to build an object recognition system that can pick out and identify objects from an inputted camera image, as shown in Figure 1, based on the registered objects.

Figure 1. Desired performance of the project.

3. System Architecture

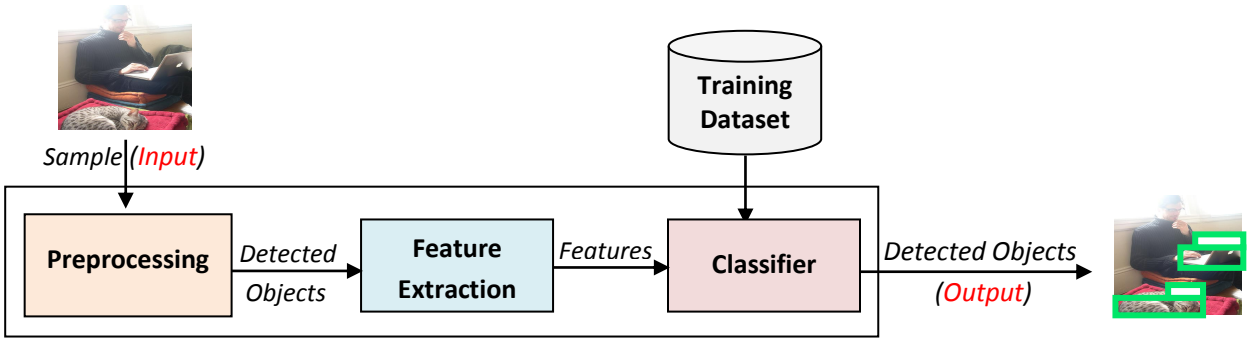

Figure 2. System Architecture

4. Dataset

- The dataset is real-world data, gathered from ImageNet [1]. **ImageNet** is an image database organized according to the WordNet hierarchy.
- The training dataset has 5 objects (classes), as shown in Figure 3, 5 different models each.
- The test dataset has 14 different images, each image may contain one or more objects as shown in Figure 4.
- Each training\testing image is given as colored image in .jpg format.

Figure 3. Objects (Cat, Laptop, Apple, Car, and Helicopter) [1]

Figure 4. A test image contains Helicopter, Cat, and Apple.

5. Data Preprocessing

- Perform any useful preprocessing routines on images, such as
  a. Converting them to grayscale, and
  b. Resizing them to 50 * 50.
- As a test image, contains more than one object, can be entered into the system. It is required to isolate objects from the background. For that, the test dataset of segemented images is provided, as shown in Figure 5. You only need to crop the detected objects from the segmented images.
- Cropped objects are the detected objects (samples) to be classified in the next steps.

(a) (b)

Figure 5. (a) original image (.jpg), and (b) the corresponding segmented image (.png).

6. Feature Extraction

Extract features of each image using statistical Principle Component Analysis (***PCA***) algorithm. Based on the cumulative eigenvalues, plot the variance curve to decide the best number of principle components (***PCs***) as shown in Figure 6.

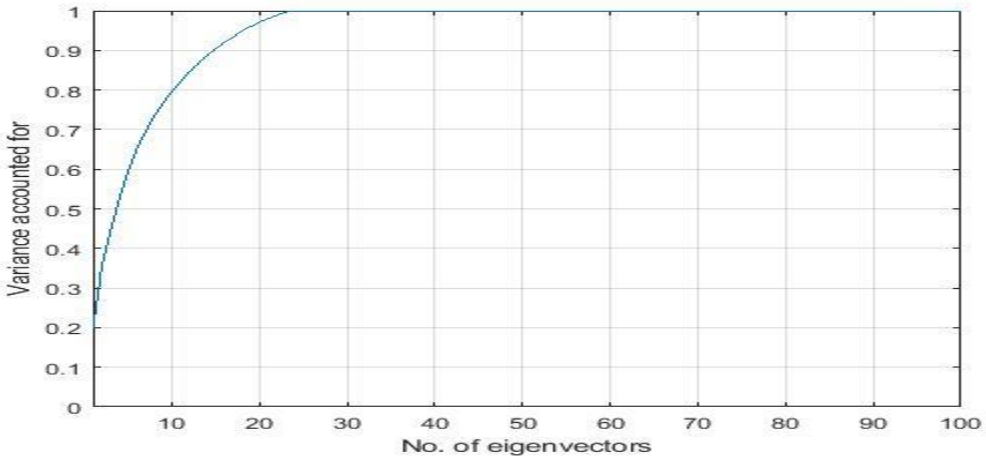

Figure 6. Variance Curve.

7. Classifiers

Implement the following two learning algorithms for recognizing objects in **ONE** package:

No.	***NN Architecture***	***Learning Algorithm***
1	Multilayer Perceptron (***MLP***)	Back-Propagation
2	Radial-Basis Function (***RBF***)	Least Mean Square

- For each ***NN Architecture***, the number of neurons in input layer is ***d*** neurons depending on the number of ***PCs*** (i.e. ***d=PCs***), and in output layer is 5 neurons as shown in Figure 7. For hidden layer, try using different numbers of layers and neurons, and different network parameters to achieve a maximum performance.

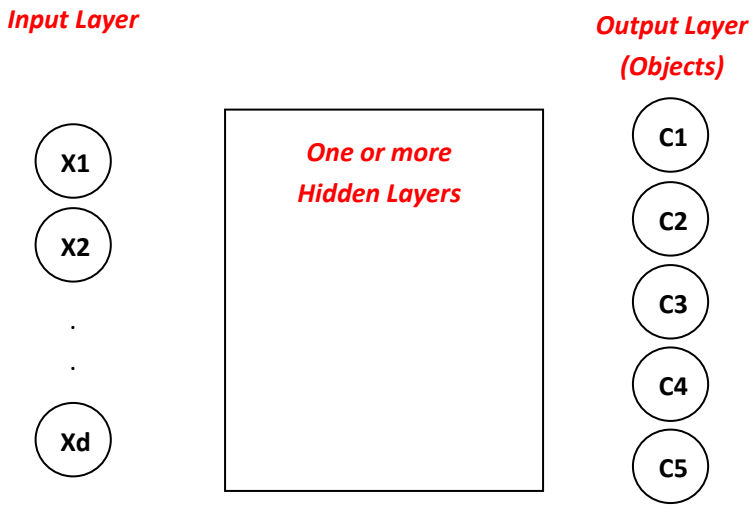

Figure 7. NN Architecture.

8. Ouput

- Identifying the detected objects on the image by drawing a rectangle around each object on the image, as shown in Figure 8.
  - For drawing the rectangle, use the location of objects. So, you need to find the minimum X and Y, and the maximium X and Y. Minimum X and Y represent the lower-left point, (maximum X – minimum X) is the rectangle width and (maximum Y – minimum Y) is the rectangle height.

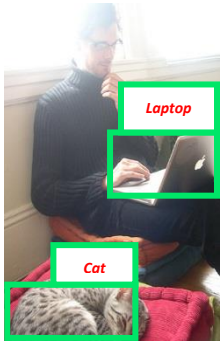

Figure 8. Output.

9. System Performance Evaluation

Separately, evaluate the performance of the two classifiers by the following techniques:

1) **Confusion Matrix**
   use it to describe the performance of a classifier on *test images*, by representing actual and predicted classifications done by that classifier.

2) **Overall Accuracy**
   use it to obtain the accuracy of a classifier from confusion matrix, by counting the number of *correct classifications* (i.e classified the same in the actual and predicted classifications), and then dividing this by the total number of *classifications*.

10. Bonus

Implement ***Principal Component Analysis (PCA)*** *using* ***Generalized Hebbian Algorithm*** as a feature extraction algorithm that finds minimum set of proper features, out of the 50*50 features, trying in improving classification performance of the project's objective.

Deliverable Specifications

11. Requirements

- **For MLP**, the user should be able to
    a. Enter number of hidden layers
    b. Enter number of neurons in each hidden layer
    c. Enter learning rate (***eta***)
    d. Enter number of epochs (***m***)
    e. Add bias or not (***Checkbox***)
    f. Choose to use *Sigmoid* or *Hyperbolic Tangent Sigmoid* as the activation function
    g. Choose the stopping criteria:
        a. fix the number of epochs,
        b. threshold the output mean squared error (in this case, the user should enter the ***mse_threshold***), or
        c. cross validation (validation subset is a random 20% of the training set. Test NN every 50 epochs)

- **For RBF**, the user should be able to
    a. Enter number of hidden neurons
    b. Enter the ***mse_threshold***
    c. Enter learning rate (***eta***)
    d. Enter number of epochs (***m***)

- The user must be able to insert an input (image) to the system, and the system has to identify objects on the inputted image.
- Use the test images to **test each classifier**. Find their performance using the **Overall Accuracy (OA)** and the **Confusion Matrix**.
- A comparative study showing the difference in applying the two classification algorithms based on the two evaluation measures mentioned above. Thus, a report template will be provided to you for filling it.
- Also, the report must be provided showing the different NN architectures and different parameters you used, and their effect on the training and testing results.

12. Rules

- The same groups of lab. tasks (any change in group members is not allowed).
- **All** the group members must be aware of the project (anyone can be asked in any part of the project).
- **IT'S NOT ALLOWED TO USE ANY AVAILABLE SOURCE CODE or LIBRARIES IN IMPLEMENTING CLASSIFICATION ALGORITHMS and BONUS.**
- You can use any programming language such as **(C++, C#, Java, Python, or Matlab)**

13. Deliverables

- Report that describes the project, system evaluation and comparative study.
- A CD contains the source code, and the report.

14. Deadlines

Delivering the whole project will be during the Neural Networks practical exam time.

15. Support

There will be support labs for answering your inquiries.

16. References

[1] ImageNet, http://image-net.org/index

Good luck! ☺