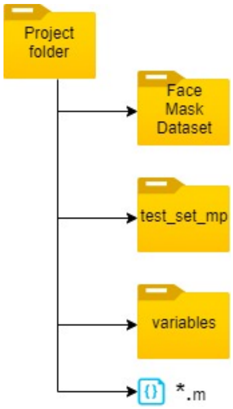
1 Machine learning course project univr

Project for the ML course at univr
Used dataset : `https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset`

2 Instructions to run the code

Create a folder for the project, then download the dataset from the link. Extract all .m files, the folders ”variables” and ”test set mp” then move the dataset and the extracted files/folders in the project folder.

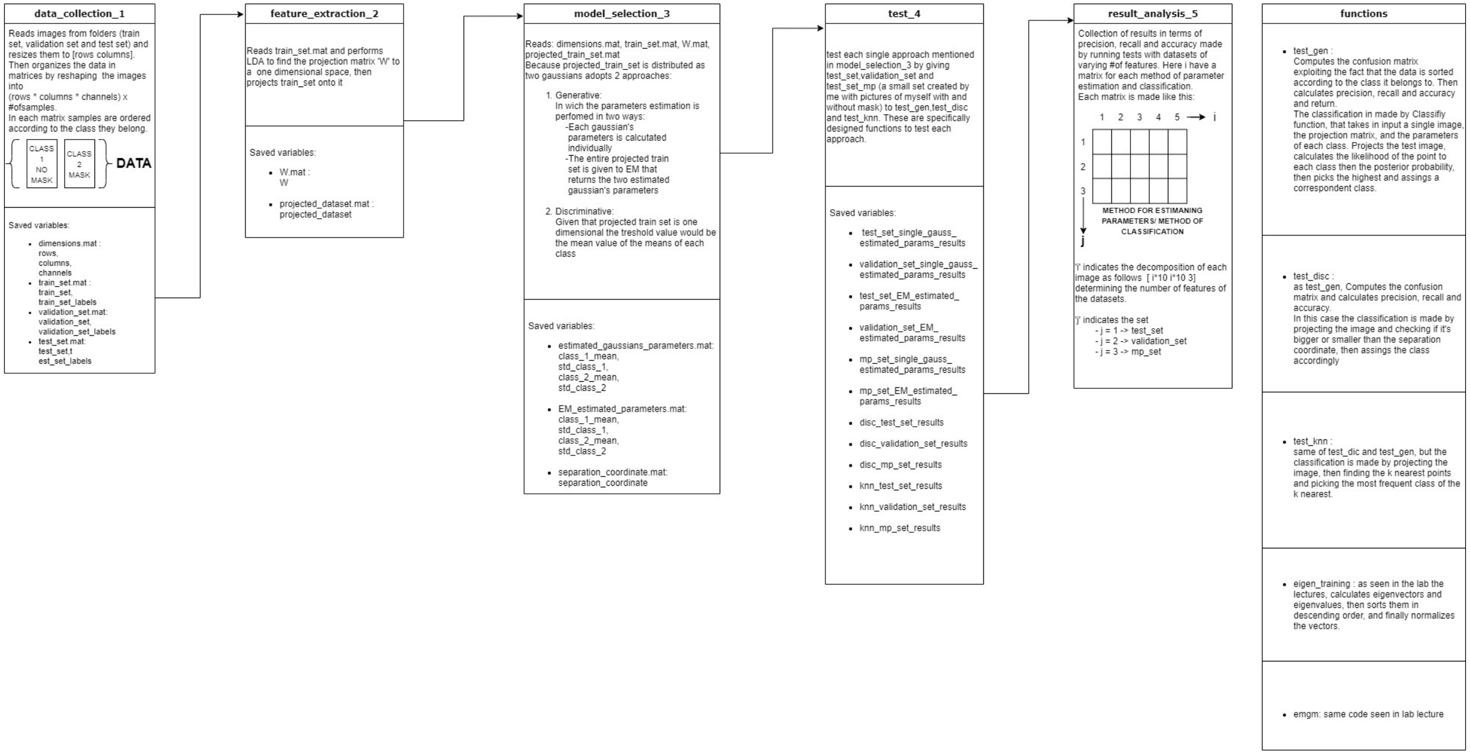


Open ”data collection 1.m” and modify the variable ”project folder” according to the path of your project folder. The execution order is:

1. data collection 1
2. feature extraction 2
3. model selection 3
4. test 4

”result analysis 5.m” is independent from the others because it uses a workspace already provided.

3 Project structure



data_collection_1

Reads images from folders (train set, validation set and test set) and resizes them to [rows columns].
Then organizes the data in matrices by reshaping the images into
(rows * columns * channels) x #ofsamples.
In each matrix samples are ordered according to the class they belong.

Saved variables:

- dimensions.mat :
  rows,
  columns,
  channels
- train_set.mat :
  train_set,
  train_set_labels
- validation_set.mat:
  validation_set,
  validation_set_labels
- test_set.mat:
  test_set,t
  est_set_labels

feature_extraction_2

Reads train_set.mat and performs LDA to find the projection matrix "W" to a one dimensional space, then projects train_set onto it

Saved variables:

- W.mat :
  W
- projected_dataset.mat :
  projected_dataset

model_selection_3

Reads: dimensions.mat, train_set.mat, W.mat, projected_train_set.mat
Because projected_train_set is distributed as two gaussians adopts 2 approaches:

1. Generative:
   In wich the parameters estimation is perfomed in two ways:
   -Each gaussian's parameters is calcutated individually
   -The entire projected train set is given to EM that returns the two estimated gaussian's parameters
2. Discriminative:
   Given that projected train set is one dimensional the treshold value would be the mean value of the means of each class

Saved variables:

- estimated_gaussians_parameters.mat:
  class_1_mean,
  std_class_1,
  class_2_mean,
  std_class_2
- EM_estimated_parameters.mat:
  class_1_mean,
  std_class_1,
  class_2_mean,
  std_class_2
- separation_coordinate.mat:
  separation_coordinate

test_4

test each single approach mentioned in model_selection_3 by giving test_set,validation_set and test_set_mp (a small set created by me with pictures of myself with and without mask) to test_gen,test_disc and test_knn. These are specifically designed functions to test each approach.

Saved variables:

- test_set_single_gauss_estimated_params_results
- validation_set_single_gauss_estimated_params_results
- test_set_EM_estimated_params_results
- validation_set_EM_estimated_params_results
- mp_set_single_gauss_estimated_params_results
- mp_set_EM_estimated_params_results
- disc_test_set_results
- disc_validation_set_results
- disc_mp_set_results
- knn_test_set_results
- knn_validation_set_results
- knn_mp_set_results

result_analysis_5

Collection of results in terms of precision, recall and accuracy made by running tests with datasets of varying #of features. Here i have a matrix for each method of parameter estimation and classification.
Each matrix is made like this:

'i' indicates the decomposition of each image as follows [i*10 i*10 3] determining the number of features of the datasets.

'j' indicates the set
- j = 1 -> test_set
- j = 2 -> validation_set
- j = 3 -> mp_set

functions

- test_gen :
  Computes the confusion matrix exploiting the fact that the data is sorted according to the class it belongs to. Then calculates precision, recall and accuracy and return.
  The classification in made by Classifiy function, that takes in input a single image, the projection matrix, and the parameters of each class. Projects the test image, calculates the likelihood of the point to each class then the posterior probability, then picks the highest and assings a correspondent class.
- test_disc :
  as test_gen, Computes the confusion matrix and calculates precision, recall and accuracy.
  In this case the classification is made by projecting the image and checking if it's bigger or smaller than the separation coordinate, then assings the class accordingly
- test_knn :
  same of test_dic and test_gen, but the classification is made by projecting the image, then finding the k nearest points and picking the most frequent class of the k nearest.
- eigen_training : as seen in the lab the lectures, calculates eigenvectors and eigenvalues, then sorts them in descending order, and finally normalizes the vectors.
- emgm: same code seen in lab lecture

4 Short report

The code was developed modularly to separate and emphasize each phase of the pipeline, also giving a certain degree of flexibility to the approaches to solve the problem.

4.1 data collection

This section reads all the images in the data set and organizes them in matrices. In particular:

- All images are resized according to the value of the variables "rows" and "columns" that have the same value. Then each image is reshaped into a **rows x columns x channels** column vector, then it's stored in a matrix.
- Each matrix has its columns ordered according to the class they belong to.

  DATA { [NO MASK] [MASK] }

  Each matrix has its label vector in wich 1 indicates membership to the class NO MASK and 2 to the class MASK. The matrices are : train set, validation set and test set.

4.2 feature extraction

This section reads train set and performs LDA to find the projection matrix **W** to a one dimensional space that maximize the separation between the classes, then projects train set onto it.

4.3 model selection

Given that projected train set is distributed as two gaussians I adopted two approaches:

1. **Generative**
   So the goal here is to find the parameters of each class. To do this I applied two methods:
   - Single class parameters estimation, that is simply the calculation of the mean and the standard deviation of each class.
   - EM parameters estimation, in wich projected train set is fed to expectation maximization algorithm that returns the two means and the two variances.
2. **Discriminative**
   Here we are looking for a treshold that divides the two classes, so I took the mean of the means of each class as the separation coordinate.

4.4 test

The sets used for testing are: test set, validation set and mp set that is a set of 10 pictures of myself without mask and 10 with mask.

- **Generative approach**
  Each image is projected by **W** into the one dimensional space. Using the estimated gaussians parameters I compute the likelihood of the image with respect to each class. Finally I compute the two posterior probabilities, pick the highest and assign the class.

- **Discriminative approach**
  Each image is projected by **W**. If the projected point is bigger than the treshold the assigned class is 2, else 1.

- **KNN**
  Each image is projected by **W**. The classification is made by using K nearest neighbors algorithm in wich I compute the K points nearest and pick the most frequent class as the class of the test image.

Each test function computes the confusion matrix and calculates precision, recall and accuracy then returns them.

4.5 result analysis

This section plots - for each model or classification method - the results of the tests in terms of precision, recall and accuracy as the number of features varies (N.B: the number of features is **rows x columns x channels**)