Politechnika Śląska w Gliwicach

Wydział Automatyki, Elektroniki i Informatyki

Podstawy Programowania Komputerów

04.02.2018r.

|  |  |
| --- | --- |
| Autor | Karol Pander |
| Prowadzący | Dr inż. Jakub Nalepa |
| Rok akademicki | 2017/2018 |
| Kierunek | Informatyka |
| Rodzaj studiów | SSI |
| Termin laboratorium | Czwartek, 13:00-14:00 |
| Grupa | 5 |
| Sekcja | 14 |

Link do kodu:

https://github.com/nieczyta/Karol-Pander-gr05-repo/blob/master/SVM

1. Treść zadania

Implement an ensemble of SVMs (LibSVM library can be used for SVMs) for a given classification/prediction problem. Details to be discussed.

1. Analiza zadania

Program odczytuje cyfrę zapisaną ręcznie w pliku graficznym.

1. Specyfikacja zewnętrzna

Po wpisaniu ścieżki do pliku w konsoli wyświetlona zostanie cyfra, którą program odczytał.

1. Specyfikacja wewnętrzna
   1. Typy użyte w programie:

* Lista składająca się z elementów typu struct trainCells\_str.
* Lista składająca się z elementów typu struct trainLabels\_str.
* Lista składająca się z elementów typu struct TrainHOG\_str.
* Lista składająca się z elementów typu struct descriptors.
* Struktura struct Test.
  1. Funkcje zdefiniowane w programie:
* string help() - pobiera ścieżkę do pliku.
* void loadTrain(string &filename, trainCells\_str \*htrC, trainLabels\_str \*htrL, int trainCellsCounter) - pobiera cyfry z pliku "digits.png".
* void loadTest(string &filename, struct Test \*test) - pobiera cyfrę z pliku podanego przez użytkownika.
* void trainHOG(HOGDescriptor hog, trainCells\_str \*htrC, trainHOG\_str \*htrH, int trainCellsCounter, int trainHOGCounter) - tworzy Histogram of Oriented Gradients z cyfr pobranych w funkcji loadTrain.
* void testHOG(Test \*test) - tworzy Histogram of Oriented Gradients z cyfry pobranej w funkcji loadTest.
* void trainMatrix(trainHOG\_str \*htrH, Mat &trainMat, int trainHOGCounter) - tworzy matryce z cyfr pobranych w funkcji loadTrain.
* void testMatrix(Test \*test, Mat &testMat) - tworzy matrycę z cyfry pobranej w funkcji loadTest.
* vector <int> vecTrainLabels(trainLabels\_str \*htrL) - tworzy vector <int> trainLabel z elementów listy.
* void Train(Ptr<SVM> svm, Mat &trainMat, trainLabels\_str \*htrL, vector <int> trainLabel) - trenuje SVM i zachowuje wyniki treningu.
* void Predict(Ptr<SVM> svm, Mat &testResponse, Mat &testMat) - odczytuje zapisaną cyfrę.
* Ptr<SVM> set\_svm() - tworzy SVM.