Politechnika Śląska w Gliwicach

Wydział Automatyki, Elektroniki i Informatyki

Podstawy Programowania Komputerów

08.02.2018r.

|  |  |
| --- | --- |
| 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" dzieląc plik na części 20x20 pikseli i zapisuje każdą z nich w matrycy, wraz z odpowiadającą jej wartością, jako osobny element listy trainCells.
* void loadTest(string &filename, struct Test \*test) - pobiera cyfrę z pliku podanego przez użytkownika do matrycy i zapisuje ją w strukturze struct Test.
* void trainHOG(HOGDescriptor hog, trainCells\_str \*htrC, trainHOG\_str \*htrH, int trainCellsCounter, int trainHOGCounter) - Używa zdefiniowanego wcześniej desktyptora hog typu HOGDescriptor, aby wypełnić listę descriptors jednowymiarowymi wektorami, zawierającymi piksele poszczególnych cyfr, utworzonymi przy użyciu funkcji compute. Elementy listy zostaną później użyte w celu stworzenia matrycy treningowej.
* void testHOG(Test \*test) - używa zdefiniowanego wcześniej deskryptora hog typu HOGDescriptor, aby stworzyć jednowymiarowy wektor, przy użyciu funkcji compute i zapisać go w strukturze struct Test
* void trainMatrix(trainHOG\_str \*htrH, Mat &trainMat, int trainHOGCounter) - tworzy matrycę treningową, która zostanie wykorzystana w funkcji Train.
* void testMatrix(Test \*test, Mat &testMat) - tworzy matrycę z cyfry pobranej w funkcji loadTest.
* vector <int> vecTrainLabels(trainLabels\_str \*htrL) - tworzy wektor <int> trainLabel z elementów listy TrainLabels.
* Ptr<SVM> set\_svm() - tworzy SVM.
* void Train(Ptr<SVM> svm, Mat &trainMat, trainLabels\_str \*htrL, vector <int> trainLabel) - tworzy dane treningowe (td) przy użyciu funkcji TrainData::create z wcześniej stworzonej matrycy treningowej, oraz wektora trainLabel przechowującego faktyczne wartości. Następnie trenuje svm używając funkcji train, która operuje na danych treningowych td, umożliwiając dopasowanie prawidłowej wartości do podanego obrazu cyfry.
* void Predict(Ptr<SVM> svm, Mat &testResponse, Mat &testMat) - odczytuje wartość cyfry z matrycy utworzonej w funkcji testMatrix.