Ion Bianca-Andreea
333AA

Traffic Sign Classification

-Implementare-

Temă: clasificarea semnelor de circulație (trecere de pietoni, cedează trecerea, stop).

Dataset:
View dataset

- Creat prin combinarea imaginilor dintr-un set de date cu mai multe semne de circulație și imagini descărcate de pe google.
- Etichetarea s-a făcut manual
- Alcătuit din 3 foldere: *crosswalk*(117 imagini), *give-way*(108 imagini), *stop*(110 imagini)
- Preprocesare:
  - Toate imaginile au extensia *.png*
  - Imaginile sunt modificate pentru a avea rezoluția *200*x*200*
  - Imaginile sunt convertite într-un format ușor de procesat de un algoritm de machine learning: *.flatten()*
- Impărțirea în seturile de antrenare, validare și testare se face cu ajutorul funcției *train_test_split()*, 80% din dataset fiind păstrat pentru antrenare, 10% pentru validare și 10% pentru testare; împărțirea se face prin apelarea de 2 ori a funcției *train_test_split()* pentru a evita pierderea de date

Algoritm:

- Pentru antrenare am utilizat algoritmul *SVC (Support Vector Classification)* – a supervised learning algorithm.
- *SVM*-urile sunt una dintre cele mai robuste metode de predicție, fiind bazate pe cadrele de învățare statistică sau teoria VC propusă de Vapnik și Chervonenkis
- Scopul acestui algoritm este de a găsi un hiperplan într-un spațiu N-dimensional (N - numărul de caracteristici) care clasifică punctele de date în diferite clase.
- Vectori suport – punctele care sunt cele mai aproapiate de hiperplan și marginea maximă.
- Margine – distanța dintre hiperplan și cel mai apropiat punct de date din oricare dintre clase.
- Marginea ar trebui să fie cât mai mare pentru a obține un model mai robust și mai generalizabil.
- Avantaje:
  - Eficient în spații cu dimensiuni mari
  - Versatil cu diferite funcții ale nucleului
- Dezavantaje
  - Sensibil la alegerea nucleului și a parametrilor
  - Ocupă memorie pentru seturi mari de date
- *SVC(kernel, c, gamma)*
  - Kernel – rbf, linear, poly
  - C – parametru de regularizare
  - Gamma – coeficient kernel

Ion Bianca-Andreea
333AA

Librării:

- openCV
- cv2
- numpy
- sklearn
  - model_selection ➔ train_test_split
  - svm ➔ SVC
  - metrics ➔ accuracy_score, classfication_report, confusion_matrix
- matplotlib

Acuratețe:

Pentru parametrii aleși, *SVC(kernel = 'linear', C = 1.0)* am obținut un procent de acuratețe de:

- Validare: 75.76%
- Testare: 70.59%

Raport de clasificare pentru setul de testare:

- Precision – capacitatea clasificatorului de a nu eticheta un eșantion negativ ca fiind pozitiv. $\frac{T_p}{Tp+F_p}$
- Recall – capacitatea clasificatorului de a găsi toate probele pozitive. $\frac{T_p}{Tp+F_n}$
- F1-Score – o medie armonică ponderată a preciziei și a sensibilității, unde un scor F-beta atinge cea mai bună valoare la 1 și cel mai slab scor la 0.
- Support - numărul de apariții ale fiecărei clase în *y_true*.

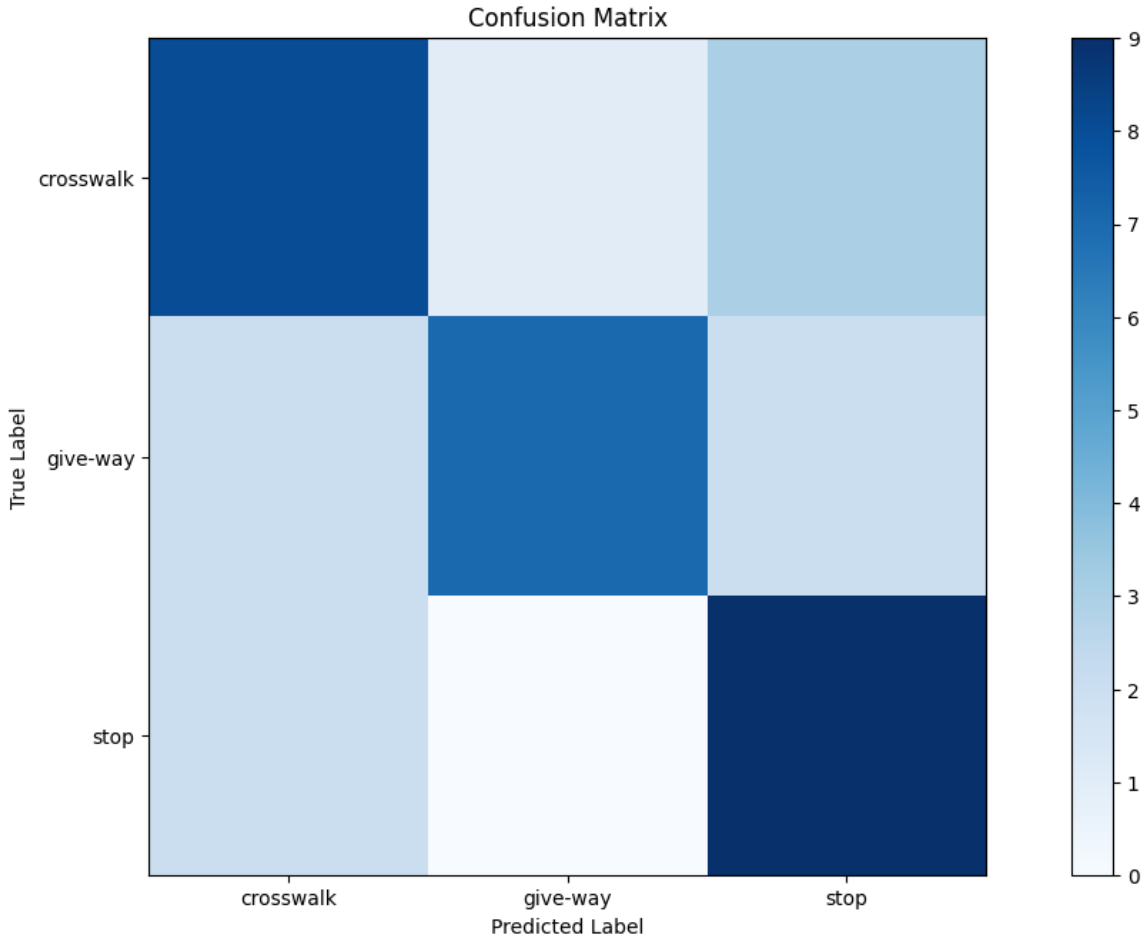
- Crosswalk
  - **Actual** – 12
  - **Predicted** – 8
  - Give-way – 1
  - Stop – 3
- Give-way
  - **Actual** – 11
  - **Predicted** – 7
  - Crosswalk – 2
  - Stop - 2
- Stop
  - **Actual** – 11
  - **Predicted** – 9
  - Crosswalk - 2

```
Classification Report for Test Set:
              precision    recall  f1-score   support

   crosswalk       0.67      0.67      0.67        12
    give-way       0.88      0.64      0.74        11
        stop       0.64      0.82      0.72        11

    accuracy                           0.71        34
   macro avg       0.73      0.71      0.71        34
weighted avg       0.73      0.71      0.71        34
```

Ion Bianca-Andreea
333AA

Matrice de confuzie:

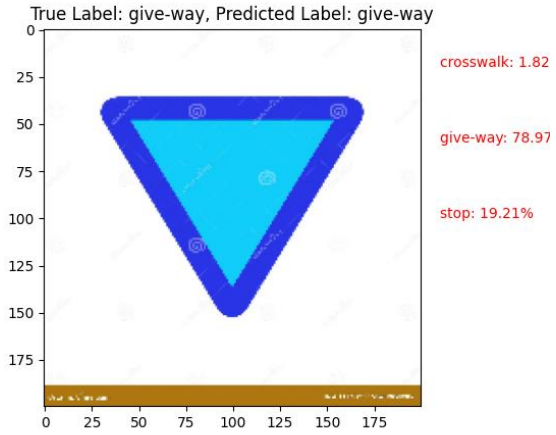
```
Confusion Matrix for Test Set:
 [[8 1 3]
 [2 7 2]
 [2 0 9]]
```

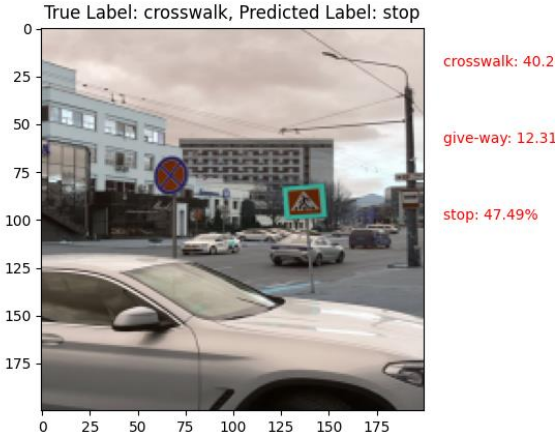


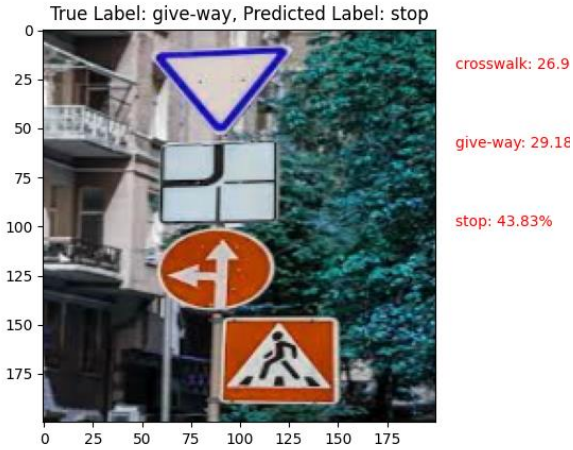
Ion Bianca-Andreea
333AA

Afișare rezultate:

View results







Ion Bianca-Andreea
333AA

Concluzii & Observații:

- Pentru *kernel = 'rbf', C = 0.1* (un nivel scăzut de regularizare, permitem clasificarea greșită a mai multor puncte de antrenament) și *gamma = 'scale'* am obținut o acuratețe a testelor de **50.00%**
- Pentru *kernel = 'rbf', C = 1.0* (un nivel moderat de regularizare) și *gamma = 'scale'* am obținut o acuratețe a testelor de **61.76%**
- Pentru *kernel = 'rbf', C = 1000* (un nivel ridicat de regularizare, impunem o marjă strictă pentru a preveni clasificările greșite) și *gamma = 'scale'* am obținut o acuratețe a testelor de **67.65%**
- Pentru *kernel = 'rbf', C = 1000, gamma = 0.1* am obținut o acuratețe a testelor de **32.35%**
- Pentru *kernel = 'linear', C = 1.0* am obținut o acuratețe a testelor de **70.59%**
- Pentru *kernel = 'poly', C = 1000* am obținut o acuratețe a testelor de **61.76%**

- Prin modificarea parametrilor funcției *SVC()* rezultă că setul de date poate fi separat de o linie dreaptă pentru a obține o performanță cât mai bună.
- În alte aplicații de detectare a semnelor de circulație a fost folosit algoritmul *Support Vector Machine*, deoarece prezintă un grad mare de flexibilitate în manipularea sarcinilor de clasificare de complexități variate.
- Modelele *SVM* funcționează similar cu rețelele neuronale clasice, dar, în comparație cu abordările tradiționale are rețelelor neuronale, teoria generalizării *SVM* permite modelelor să evite *overfitting*-ul datelor.
- Am ales să folosesc algoritmul SVC, deoarece este printre primele exemple din lista de algoritmi de învățare supervizată din curs, iar nivelul de robustețe este destul de ridicat.
- Am adaptat exemple găsite pe internet și exemplele din laborator pentru setul meu de date și pentru scopul temei alese. Am modificat parametrii funcției SVC pentru a obține cea mai bună acuratețe.

Referințe:

- SVM - Exemple
- sklearn.svm.SVC
- Support_vector_machine
- Support Vector Machine - YouTube
- Implementation_of_Various_Machine_Learning_Algorithms_for_Traffic_Sign_Detection_and_Recognition
- Road sign
- Confusion matrix
- Classification report
- SVM