

Tehnici de invatare automata

Raport - Tema 1

Clasificarea imaginilor cu copii si adulti

Mazilu Gabriela
Grupa 334AA

Prezentarea temei:

Proiectul implementat abordează problematica clasificării imaginilor în două categorii distincte, copii și adulți, prin antrenarea unui model bazat pe algoritmul SVM (SVC).

Setul de date:

Setul de date este unul personalizat, astfel încât l-am structurat în 3 foldere principale: **train** (setul de antrenament), **validate** (setul de validare), **test** (setul de testare). Fiecare la rândul lui conține 2 subfoldere ce indică etichetele clasificării ('adults', 'children').

Pentru clasa „children”:

Imagini pentru antrenare: 1518
Imagini pentru validare: 325
Imagini pentru testare: 325

Pentru clasa „adults”:

Imagini pentru antrenare: 1810
Imagini pentru validare: 388
Imagini pentru testare: 388

Această împărțire respectă distribuția de aproximativ 70% pentru antrenare, 15% pentru validare și 15% pentru testare. Astfel, nu a mai fost nevoie să apelez la funcția „train_test_split()” din biblioteca scikit-learn în cadrul implementării codului.

```
train_path = "C:\\Users\\Gabriela\\Desktop\\set de date\\train"
validate_path = "C:\\Users\\Gabriela\\Desktop\\set de date\\validate"
test_path = "C:\\Users\\Gabriela\\Desktop\\set de date\\test"

# Incarcarea datelor
X_train, y_train = load_images_and_hog_features(train_path)
X_val, y_val = load_images_and_hog_features(validate_path)
X_test, y_test = load_images_and_hog_features(test_path)
```

Algoritmul utilizat si etapa de preprocesare:

Algoritmul de Machine Learning utilizat : **Support Vector Machine (SVM) pentru clasificare (SVC)**

- SVM face parte din categoria **Algoritmilor de Invatare Supervizata**; este preferat pentru sarcina aleasa deoarece **poate gestiona bine seturi de date cu multe caracteristici**.
 - Acesta este folosit ca parte a unui pipeline pentru a clasifica imaginile in doua categorii: copii si adulti.
 - Se utilizeaza un kernel liniar pentru a gasi un hiperplan intr-un spatiu multidimensional care separa cel mai bine datele in doua clase
- Pentru etapa de preprocesare am utilizat **detectorul Haar Cascade** pentru localizarea și extragerea fețelor din imagini, urmată de redimensionarea acestora la dimensiunea de 64x64 pixeli. De asemenea, am aplicat extracția de caracteristici utilizând **metoda Histogram of Oriented Gradients (HOG)**, esențială pentru procesul de clasificare.

```
# Detectorul Haar Cascade pentru fete
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml')

# Functia pentru incarcarea imaginilor, detectarea fetei si extragerea caracteristicilor HOG
def load_images_and_hog_features(folder_path):
    hog_features = []
    labels = []
    class_names = os.listdir(folder_path)

    for class_name in class_names:
        class_path = os.path.join(folder_path, class_name)
        if os.path.isdir(class_path):
            for filename in os.listdir(class_path):
                if filename.endswith('.png'):
                    img_path = os.path.join(class_path, filename)
                    img = cv2.imread(img_path)
                    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

                    # Detectarea fetelor
                    faces = face_cascade.detectMultiScale(gray, 1.1, 4)
                    for (x, y, w, h) in faces:
                        # Croirea imaginii pentru a include doar fata
                        face_roi = gray[y:y+h, x:x+w]
                        face_resized = cv2.resize(face_roi, (64, 64))

                        # Calcularea caracteristicilor HOG
                        features = hog(face_resized, orientations=9, pixels_per_cell=(8, 8),
                                      cells_per_block=(2, 2), visualize=False)
                        hog_features.append(features)
                        labels.append(class_name)
                        break # Presupunem ca exista o singura fata relevanta per imagine

    return hog_features, labels
```

Parametrii:

Pentru Detectorul Haar Cascade:

‘scaleFactor’: 1.1 determina cat de mult imaginea este redusa la fiecare scala a imaginii

‘minNeighbors’:4 specifica cati vecini trebuie sa aiba fiecare candidat la detectarea fetei

Pentru HOG:

‘orientations’:9 specifica numarul de orientari ale bin-urilor gradientului in histograma

‘pixels_per_cell’(8,8) specifica dimensiunea celulei peste care se calculeaza histograma gradientului

‘visualize’: False specifica daca se doreste vizualizarea HOG

Pentru SVM:

‘kernel’: ‘linear’ specifica tipul kernel-ului

‘probability’: True permite calcularea probabilitatilor de apartenenta la clasa pentru fiecare instantă

Acești parametri sunt esențiali pentru funcționarea și performanța algoritmului de clasificare. Detectorul Haar Cascade localizează fețele în imagini, caracteristicile HOG descriu aceste fețe, iar SVM efectuează clasificarea finală bazată pe aceste caracteristici.

Performanta modelului

Pentru a observa performanta modelului, am analizat:

- **Training Accuracy** = cat de bine performeaza modelul pe setul de antrenament
- **Validation Accuracy** = cat de bine performeaza modelul pe setul de validare
- **Test Accuracy** = capacitatea modelului de a clasifica corect date noi(setul de test)
- **Classification Report on Test Set** = include precizia, sensibilitatea si scorul F1
- **Macro Avg** = calculeaza media aritmetica a scorurilor, tratand ambele clase ca fiind la fel de importante
- **Weighted Avg** = reflecta performanta modelului in conditiile in care distributia claselor nu este uniforma

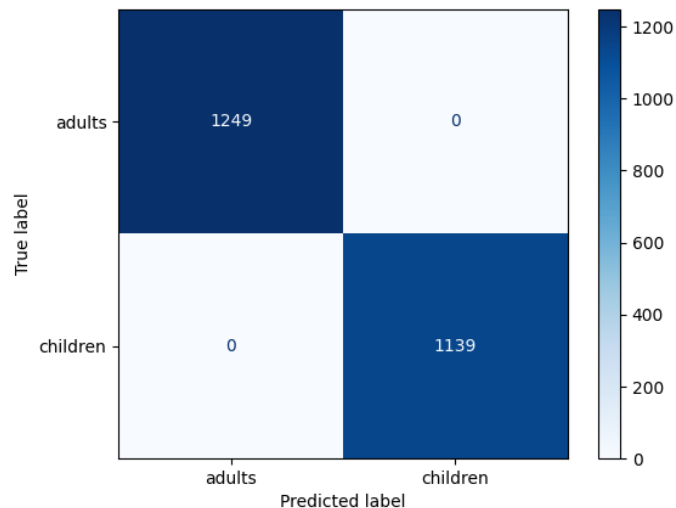
Classification Report on Test Set:				
	precision	recall	f1-score	support
adults	0.79	0.78	0.78	384
children	0.73	0.75	0.74	316
accuracy			0.76	700
macro avg	0.76	0.76	0.76	700
weighted avg	0.76	0.76	0.76	700

Classification Report on Training Set:				
	precision	recall	f1-score	support
adults	1.00	1.00	1.00	1249
children	1.00	1.00	1.00	1139
accuracy			1.00	2388
macro avg	1.00	1.00	1.00	2388
weighted avg	1.00	1.00	1.00	2388

- Matricile de confuzie:

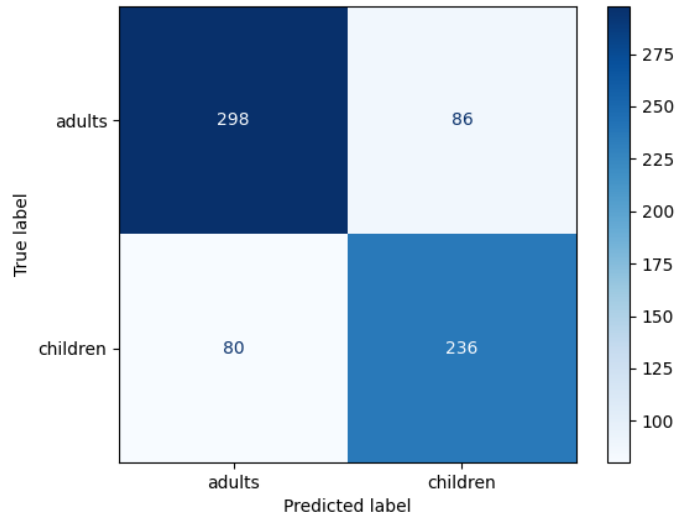
Matricea de confuzie prezintă numărul de predicții corecte și incorecte făcute de model, împărțite în categoriile adevărat pozitive (TP), fals pozitive (FP), adevărat negative (TN) și fals negative (FN).

Matricea de confuzie pentru setul de antrenare:



Modelul a clasificat perfect datele de antrenament, fara a exista erori sau clasificari gresite, indicand o acuratete de 100%. Acest lucru sugereaza ca modelul a invatat setul de antrenament extrem de bine, generand overfitting.

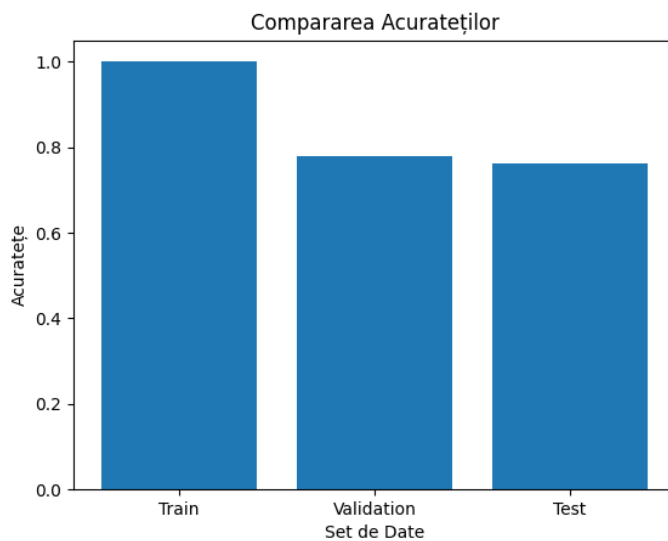
Matricea de confuzie pentru setul de test:



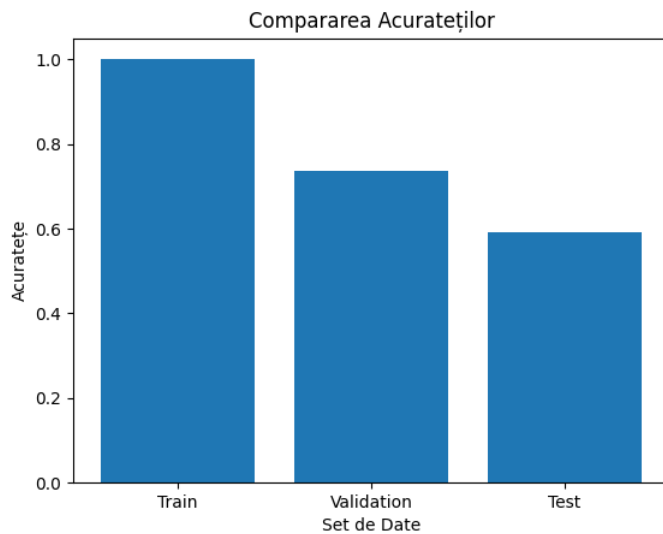
- 298 de adulti au fost clasificati correct (TP)
- 86 de adulti au fost clasificati gresit, drept copii (FN)
- 80 de copii au fost clasificati gresit, drept adulti (FP)
- 263 de copii au fost clasificati correct (TN)

Aceasta arata ca, desi majoritatea clasificarilor sunt corecte, exista o rata semnificativa de erori in clasificarea ambelor categorii.

- **Diagrama Acuratetilor pentru modelul implementat cu SVM, HOG si HAAR:**



- **Diagrama Acuratetilor pentru modelul implementat doar cu SVM:**



Deși ambele metode de implementare a clasificării prezintă un overfitting, se poate observa că utilizarea tehnicii Haar pentru detectarea feței și a descriptorului HOG pentru caracteristici aduc o îmbunătățire considerabilă performanței modelului.

Biblioteci Python:

```
import cv2
from skimage.feature import hog
from sklearn import svm
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix, ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
import os
import matplotlib.pyplot as plt
```

- Biblioteca OpenCV este utilizată pentru procesarea de imagini și computer vision
- Biblioteca scikit-image este utilizată pentru procesarea de imagini. Din aceasta, este importantă funcția 'hog' pentru extragerea Histogram of Oriented Gradients\
- Biblioteca scikit-learn pentru învățare automată. Sunt importante următoarele module:
 - 'svm' – modul pentru Support Vector Machine
 - 'metrics' – include funcții pentru calcularea metricilor de performanță a modelului
 - 'preprocessing' – include 'StandardScaler' pentru normalizarea caracteristicilor
 - 'pipeline' – pentru a construi un pipeline de procesare și clasificare
- Biblioteca os este utilizată pentru interacțiunea cu sistemul de operare
- Biblioteca matplotlib este utilizată pentru generarea de grafice și diagrame în Python

Concluzie

În urma evaluării rezultatelor asupra performanțelor pe care le are modelul implementat, s-a observat că algoritmul SVM pentru tema aleasă produce overfitting, deși acurătatea setului de testare este una multumitoare.

Pe baza analizei efectuate asupra performanței modelului de clasificare cu și fără utilizarea detectării feței Haar și a caracteristicilor HOG, am ajuns la concluzia că preprocesarea avansată a datelor îmbunătățește semnificativ capacitatea modelului de a învăța și de a distinge între categoriile „children” și „adults”.