

Proiect TIA

Detectarea umana in imagini

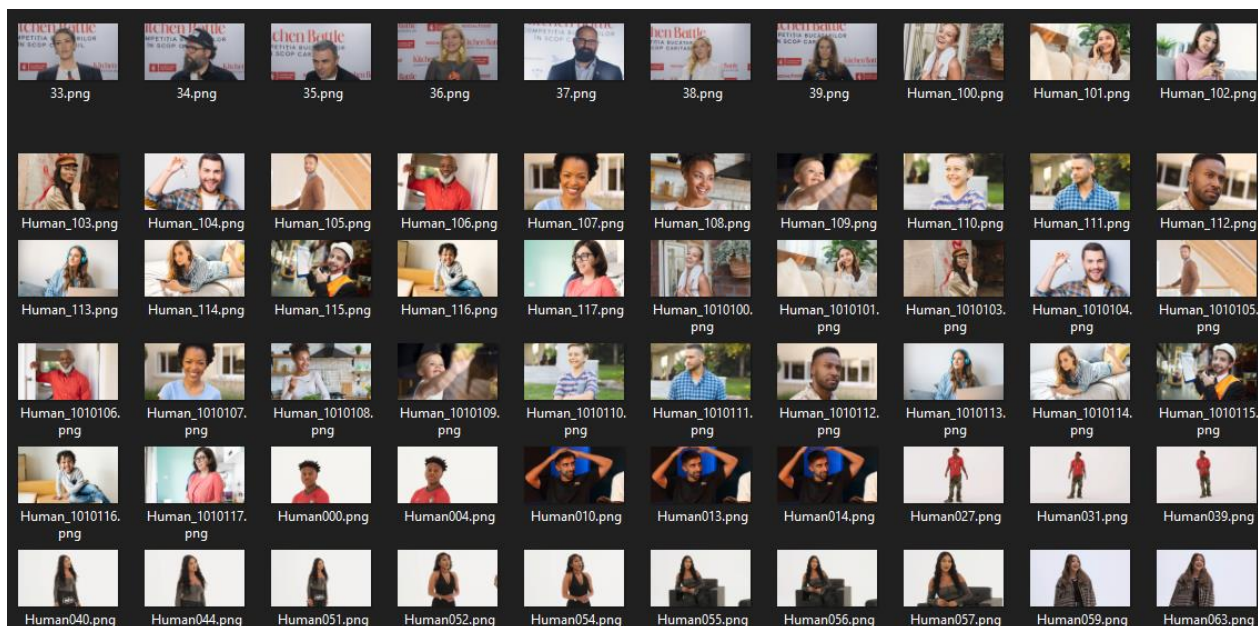
Blejan Alexandru

Link Dataset: <https://s.go.ro/2nfiomy8>

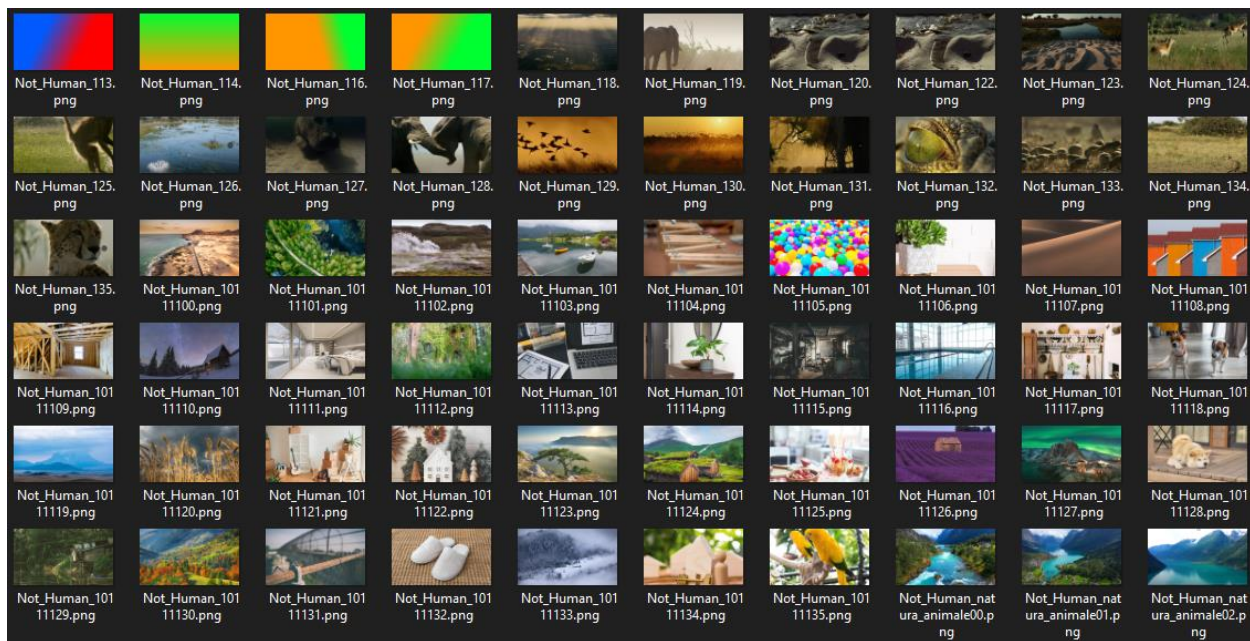
Proiectul consta in aplicarea si antrenarea unui algoritm de invatare automata supervizata pentru a recunoaste daca intr-o poza data este prezenta o persoana sau nu. Codul antreneaza un clasificator Support Vector Machine (SVM) si foloseste Histogram of Oriented Gradients (HOG) pentru a distinge imaginile care contin sau nu oameni. Setul de date este impartit in invatare, validare si testare, cu imagini sortate de mine, ce sunt incarcate si procesate in loturi. Exactitatea testului este calculata si afisata in terminal, iar imaginile de test sunt afisate. Se foloseste algoritmul SVM implementat in clasa SVC din libraria scikit-learn.

Setul de date

Setul de imagini este etichetat si sortat in doua parti: Human & Not_Human. Iar imaginile testate sunt diferite de cele din antrenament, dar sunt etichetate pentru calcularea acuratetii.



Imaginile cu persoane (168 de imagini) reprezinta frame-uri exportate manual din videoclipuri populare de pe youtube, imagini descarcate de pe internet si frame-uri exportate din interviuri la care am lucrat in proiecte personale. Anumite imagini sunt asemanatoare, le-am schimbat rotatia si paleta de culori.



Imaginile fara persoane (159 de imagini) sunt reprezentate de frame-uri exportate din videoclipuri din natura si orase in care nu apar persoane, dar sunt prezente si animale.

Numarul de imagini din ambele set-uri este apropiat, pentru ca algoritmul sa nu favorizeze o anumita eticheta.

Algoritmul si parametrii utilizati

Am utilizat algoritmul SVM cu nucleul (kernel) liniar, folosit pentru clasificarea imaginilor binare, si HOG pentru analizarea formei si informatiei din imagine.

```
87 clf = SVC(kernel='linear')
88 clf.fit(X_train, y_train)
```

Alegerea unui kernel liniar indica faptul ca granita de decizie dintre clase se presupune a fi un hiperplan liniara.

```
15 hog = cv2.HOGDescriptor()
```

Parametrii HOG-ului nu sunt specificati, i-am utilizat pe cei impliciti (default) ai clasei. Nu am considerat ca este nevoie de altii diferiti.

Rezultatul antrenarii si testarii

Am afisat in terminal exactitatea setului de validare si de testare.

```
90 y_verify_pred = clf.predict(X_verify)
91
92 accuracy_verify = accuracy_score(y_verify, y_verify_pred)
93 print(f"Validation Accuracy: {accuracy_verify}")
94
95 y_test_pred = clf.predict(data_test)
96
97 accuracy_test = accuracy_score(labels_test, y_test_pred)
98 print(f"Test Accuracy: {accuracy_test}")
```

```
Validation Accuracy: 0.8636363636363636
Test Accuracy: 0.875
```

Rezultatele testului sunt afisate folosind matplotlib:

Prediction: Person



Prediction: Not a Person



Prediction: Person



Prediction: Person



Prediction: Person



Prediction: Not a Person



Prediction: Person



Prediction: Not a Person



Librariile folosite

- `import cv2`
- `from skimage import exposure`
- `from sklearn.model_selection import train_test_split`
- `from sklearn.svm import SVC`
- `from sklearn.metrics import accuracy_score`
- `import os`
- `import numpy as np`
- `import matplotlib.pyplot as plt`

Aceste librari sunt necesare pentru procesarea imaginii, antrenare si validare, clasificare, evaluarea acuratetei, interactiunea cu sistemul de operare si afisarea vizuala a imaginilor testate.

Funcție importantă

Funcția primește imaginea, îi schimbă dimensiunea, o convertește în grayscale și returnează aplatizarea funcțiilor HOG.

```
# Function to extract HOG features from an image
def extract_hog_features(image):
    resized_image = cv2.resize(image, (64, 128)) # resize
    gray = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)
    hog = cv2.HOGDescriptor()
    features = hog.compute(gray)
    features = exposure.rescale_intensity(features, in_range=(0, 10))
    return features.flatten()
```

Observatii

- Schimbând parametrii SVC-ului cu Gamma sau C, îmi scade acurătatea validării și testării.
- Nu funcționează bine pentru animalele asemănătoare oamenilor.

Referințe

- <https://chat.openai.com/>
- <https://elements.envato.com/>
- <https://youtu.be/pb-j3svRQLI?si=d2MX3P8R7RVXr5kj>