Detectia stresului din imagini

1st Chelea Diana-Maria
1306B

2nd Spiridon Bianca
1306B

***Abstract*—Stresul constituie un factor ce afecteaza aspectele cotidiene si influenteaza atat sanatatea mintala, cat si gandirea. Acest proiect abordeaza tehnici de machine learning, utilizand toolkit-ul Dlib, pentru a detecta expresiile faciale din imagini, fiind capabil sa precizeze nivelul de stres al persoanelor. O alta ramura a ML folosita este Deep Learning, care consta in utilizarea de retele neuronale artificiale pentru a prelua caracteristici, adica structuri alcatuite din mai multe straturi, vizibile si ascunse, care transforma imaginea initiala. Partea de cod foloseste libraria Python facerecognition, OpenCV si mediul de programare Colab. Pentru inceput, vom prelua un set de imagini de pe site-uri web, iar algoritmul este implementat prin tehnici precum Histogram of Oriented Gradients, care ajuta la identificarea unui obiect in functie de contururi, bazandu-se pe gradientul imaginii, sau Haar Cascades, metoda care analizeaza variatiile pixelilor din subregiuni ale imaginii pentru a determina anumite caracteristici specifice.**

***Index Terms*—stres, recunoastere faciala, OpenCV, ML, retele neuronale, Haar Cascades, caracteristici**

I. Introducere

Acest fenomen omniprezent in existenta umana este o realitate inevitabila si complexa care ne insoteste in fiecare aspect al vietii noastre. Este important sa intelegem ca cresterea sau intensificarea stresului poate avea un efect negativ, chiar si atunci cand este considerat adaptiv sau normal. Intelegerea si gestionarea acestuia sunt esentiale pe masura ce societatea noastra se schimba si avem ritmuri din ce in ce mai alerte si solicitante. Interesul in cercetare a atins cote considerabile, desfasurandu-se studii in domenii precum neurostiinte, psihologie si medicina. In acest scop, s-au conturat diverse solutii. Aceste solutii variaza de la strategii de adaptare personala, pana la implementarea unor tehnologii ce pot detecta stresul. Scopul acestei lucrari este de a dezvolta si a implementa un program pentru detectarea stresului in imagini, punand accent pe analiza expresiilor faciale. Prin utilizarea tehnologiilor de recunoastere faciala si al algoritmilor deep learning, intentionam sa identificam caracteristici faciale specifice, asociate starii de stres. Ne propunem sa dezvoltam un instrument precis, capabil sa ofere o evaluare obiectiva a stresului. Prin imbinarea tehnologiei cu intelegerea profunda a comportamentului uman, aceasta lucrare va ajuta in cercetarea stiintifica, dar si in implementarea unor solutii practice pentru prevenirea stresului. Proiectarea si implementarea acestui program ar include: procesul de colectare a datelor, preprocesarea fotografiilor: redimensionarea si normalizarea imaginilor pentru a garanta coerenta si eficienta analizei, implementarea algoritmului de identificare a fetei, selectarea unui model de deep learning (CNN) pentru extragerea caracteristicilor fetei, utilizarea unui model machine learning pentru a urmari coordonatele sprancenei, testarea algoritmului si analiza expresiilor faciale in contextul stresului.

Identify applicable funding agency here. If none, delete this.

II. State-of-the-art

Diversele abordari si implementari care au fost dezvoltate și aplicate in ultimii ani au imbunatatit semnificativ procesul de identificare si evaluare a stresului prezent in imagini. Ca exemplu, acestea includ:

A. Retele neuronale CNN

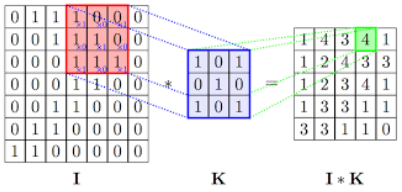
Retelele neuronale extrag caracteristicile fetelor din imagini pentru recunoașterea facială. Acest lucru le permite să distinga diferite persoane din imagini noi. CNN contine: input layer, hidden layers si output layer, toate conectate intre ele. Primul nivel accepta intrari in diferite forme, straturile ascunse le prelucreaza,iar layerul de iesire contine rezultatul final.

1. Convolution layer: rezultatele filtrarii imaginii initiale (peste imaginea de intrare se aplica filtre, pentru a evidentia obiecte).

Formula dupa care se calculeaza convolutia:

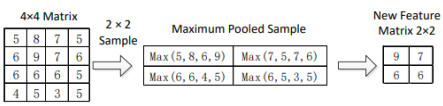
$$s(i,j) = (X \times W)(i,j) + b = \sum_{k-1}^{n\_in} (X_k \times W_k)(i,j) + b$$

Convolutie:



2.Pooling layer: reprezinta layerul obtinut prin micsorarea imaginii de intrare, ajuta la pastrarea caracteristicilor relevante. De exemplu, max pooling reduce dimensiunea selectand valoarea maxima dintr-o regiune specifica.

Pooling:



Scopul convolutiei si al pooling-ului este de a reduce progresiv volumul datelor si de a extrage elemente care devin tot mai abstracte in straturile ulterioare ale retelei.

3.ReLU: reprezinta layerul Rectified Linear Unit. Astfel, daca valorile din matrice sunt mai mici decat 0, clar pixelii nu sunt relevanti pentru cautare, deci ii vom ignora, atribuindu-le valoarea 0.

B. The Histogram of Oriented Gradients

The Histogram of Oriented Gradients este o tehnica de extragere a caracteristicilor din imagini, utilizata pentru detectia de obiecte sau recunoastrea lor in imagini . Aceasta tehnica numara aparitiile orientarii gradientului intr-un segment specific al unei imagini. Descriptorul HOG se concentreaza pe forma sau structura unui obiect. Deoarece calculeaza caracteristicile utilizand atat magnitudinea, cat si unghiul gradientului, aceasta tehnica este mai eficienta decat orice descriptor de margini , creand histograme pentru zonele imaginii . Pasii de calculare ai histogramei sunt :

- redimensionarea imaginii la o forma standard de 128x64 pixeli
- calcularea gradientului, a magnitudinii si unghiul gradientului

  Sobel

$$G_x \rightarrow \begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y \rightarrow \begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

  Prewitt

$$G_x \rightarrow \begin{bmatrix} -1 & 0 & 1 \\ -1 & \boxed{0} & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

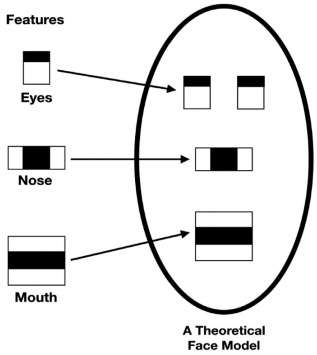
$$G_y \rightarrow \begin{bmatrix} -1 & -1 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- dupa ce obtinem gradientul pentru fiecare pixel, matricile sunt impartite in blocuri si se creeaza histograme ale gradientilor
- urmeaza normalizarea in recunoasterea faciala, care este utilizata pentru a pregati cadrele si a crea un cadru uniform pentru machine learning
- formarea histogramei finale a gradientilor orientate: un vector de caracteristici ce contine informatii despre pozitia ochilor, nasului , textura pielii , etc. Acest vector este rezultatul extragerii si sumarizarii informatiilor relevante din imaginea fetei .
- ultimul pas este clasificarea si detectia, unde se foloseste un algoritm de invatare pentru probleme de clasificare si regresie .

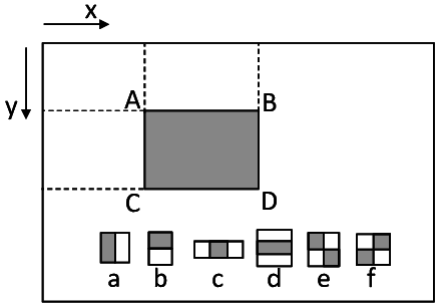
C. Haar Cascades

Detectarea obiectelor utilizand Haar feature-based cascade classifiers reprezinta o metoda machine learning unde o functie cascade este antrenata pe mai multe imagini pozitive si negative. Desi sunt antrenati sa identifice un singur tip de obiect, acesti clasificatori pot fi folositi in paralel, pentru a detecta de exemplu ochii si fata.

- **Algoritmul are nevoie de un set de imagini pozitive** (in care se regasesc expresii faciale) si unul de **imagini negative** (in care nu exista oameni) pentru a colecta Haar features (trasaturile). Acestea sunt extrase folosind blocuri de tipul:

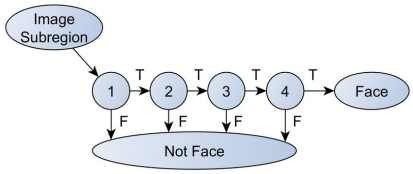


  Haar features sunt calculate prin scaderea sumei intensitatilor pixelilor din zona alba din cea a pixelilor din zona neagra. Deoarece acestea sunt greu de determinat pentru o imagine de mari dimensiuni, algoritmul foloseste integral images.
- **Creare integral images**: In loc de calcularea pentru fiecare pixel in parte, se vor crea subregiuni ale imaginii pentru care vom calcula trasaturile.



- **Adaboost alege cele mai bune trasaturi**: Se va folosi o combinatie de weak classifiers (metode de clasificare nu foarte precise, dar care pot sa dea rezultate importante folosite impreuna) pentru a crea un strong classifier pe care algoritmul il poate folosi pentru a detecta obiectele. Pentru a crea acesti clasificatori slabi, algoritmul va aplica ferestre peste imagine, calculand pentru fiecare in parte caracteristicile Haar. Acestea vor fi comparate cu un prag anterior, care separa fetele de restul obiectelor.

- **Combinarea acestor clasificatori slabi intr-un clasificator puternic** se realizeaza prin intermediul cascade classifiers.
- **Implementare Cascade Classifier**: este alcatuita dintr-o serie de stari, unde fiecare nivel este o colectie de weak classifiers. Fiecare nivel decide daca obiectul este sau nu o expresie faciala – positive: s-a identificat o fata, negative:cauta mai departe in urmatorii classifiers. Este esential sa avem o rata cat mai mica de raspunsuri negative, deoarece afecteaza grav algoritmul de detectie.



D. AdaBoost

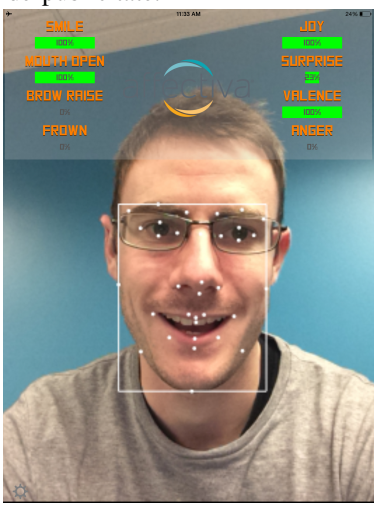
AdaBoost este un algoritm de invatare automata utilizat pentru clasificare si regresie . Acest algoritm construieste un model puternic prin combinarea unor modele de baza(weak learners). Metoda asociază o pondere fiecărui exemplu de antrenare.Iniţial toate instanţele au aceeaşi pondere de 1/n. Algoritmul se bazeaza pe ideea de a gasi clasificatorul slab cel mai bun pe baza erorii ponderate(cea mai mica valoare), apoi modificam importanta exemplelor. Exemplele care au fost clasificate gresit vor avea o pondere mai mare, ceea ce le va face mai importante pentru urmatorii clasificatori slabi, in timp ce exemplele care au fost clasificate corect vor avea o pondere mai mica. In urmatorul pas vom descoperi un alt clasificator ineficient, deoarece trebuie sa clasifice corect instantele cu pondere mai mare, avand o parte mai mare din eroarea ponderata. Scorul final este influentat de fiecare clasificator slab. Aceasta influenta este evaluata in functie de cat de bine performeaza pe setul de antrenare.

E. SVM

Un instrument de învățare automată eficient cunoscut sub numele de SVM (Support Vector Machine) are capacitatea de a gestiona seturi de date extrem de complexe, de recunoaștere facială, insa nu este practic in sistemele real-time. Practic, SVM cauta un hiperplan pe care il va plasa astfel incat distanta dintre 2 elemente sa fie maxima. Atunci cand introducem o noua data (o noua persoana), algoritmul va sti sa o plaseze in multimea corecta. CNN este mai precis decat SVM, fiind utilizat in prezent.

III. Solutii comerciale

- Tehnologia Affectiva poate fi integrata in aplicatii care doresc sa includa functionalitati de recunoastere a emotiilor. Prin utilizarea unei camere web, Affectiva poate identifica atat fata, cat si punctele cheie ale acesteia, pentru a clasifica expresiile faciale in sapte emotii principale(furie, dezgust, frica, bcurie, tristete si surpriza). Aplicatia este destinata agentiilor de publicitate care doresc sa testeze reactia consumatorilor la videoclipuri, reclame si emisiuni tv. Rezultatele tehnologiei Affectiva sunt extrem de promitatoare avand in vedere parteneriatele cu Coca Cola si Mars. Este folosita de peste 70 la suta dintre companiile de publicitate.



- O alta aplicatie cunoscuta care utilizeaza recunoasterea faciala pentru detectarea si recunoastere emotiilor este Microsoft Emotion API. Aceasta poate identifica emotiile umane, cum ar fi bucurie, tristete, furie, dispret si altele, folosind expresiile faciale. Dezvoltatorii pot obtine rezultate privind emotiile din fetele media prin incarcarea imaginilor sau transmiterea fluxurilor video catre API-ul Microsoft Emotion.

IV. Descrierea metodelor

Pentru implementarea codului nostru, am folosit limbajul Python, impreuna cu libraria facerecognition (din dlib, foloseste CNN), functii din OpenCV si mediul de programare Colab. Am creat un nou folder pentru a pune imaginile pe care le-am selectat de pe site-uri web pentru a incepe procesarea. Am ales multiple fotografii cu aceleasi persoane, din ipostaze diferite sau care afiseaza diverse emotii, pentru a pune in evidenta posibilitatea recunoasterii si gruparii imaginilor. La rulare, codul va incadra intr-un chenar rosu expresia faciala identificata in poza, urmand ca la final sa afiseze, in ordine, imaginile care contin aceeasi persoana.

Dintre functiile din libraria cv2 amintim:

- **cv2.imread**: citeste imaginea, primeste ca argument path-ul
- **cv2.rectangle**: pentru afisarea chenarului
- **cv2.imshow**: pentru afisarea imaginii

Algoritm detectare fete in imagini:

- 1. facerecognition.facelocations: returneaza un array care contine lista de coordonate pentru fiecare fata din lista de imagini
- 2. salvam coordonatele pentru fiecare fata din imagine intr-un tuplu (top,right,bottom,left); acestea vor fi folosite la trasarea conturului cu rosu pentru a evidentia zona capului

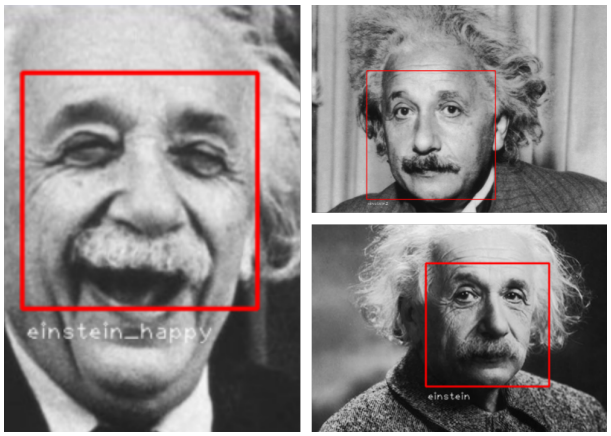
Algoritm recunoastere fete si afisarea imaginilor in ordine:

- 1. vom intera prin setul de date, adaugand intr-o lista knownnames toate numele imaginilor;
- 2. pentru a afla daca in imagini se repeta aceeasi oameni, comparam imaginile una cate una (prima cu a doua, cu a treia, samd). Vom folosi functia comparefaces, care primeste ca argumente cele 2 imagini. Aceasta functie returneaza un array de tip bool, comparand cele 2 arrayuri ale functiei facerecognition; atunci cand gaseste valori comune sau foarte asemanatoare pe aceeasi pozitie, returneaza True;
- 3. Daca s-au gasit 2 imagini cu aceeasi persoana, vom adauga numele acesteia intr-o lista doar o singura data;
  In ultima parte a codului, am evidentiat zonele de interes pentru algoritm, prin utilizarea functiei facelandmarks, care returneaza coordonatele pentru diferite caracteristici ale fetei (chin, left-eyebrow, nosebridge, etc), mai apoi colorandu-le cu functia fill din matplotlib.pyplot.

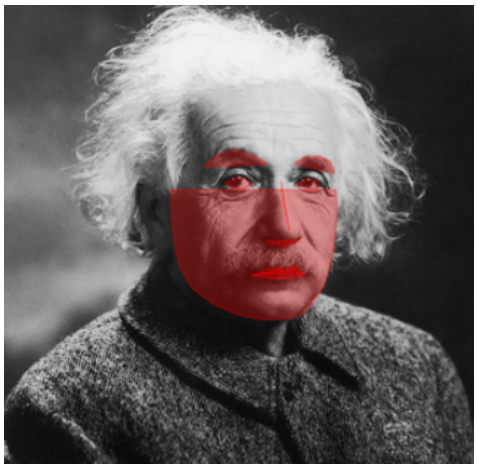
V. Rezultate preliminare

Rezultatele rularii:

Daca s-au gasit mai multe imagini cu aceeasi persoana, se vor afisa din nou doar imaginile in care apare persoana, impreuna cu numele acesteia.



Rezultatul functiei plotfacelandmarks



VI. Concluzii preliminare

Asadar, pana in acest stadiu al proiectului am reusit sa identificam expresiile faciale ale persoanelor din fotografii, utilizand in principal functii din libraria facerecognition. De asemenea, algoritmul poate identifica si daca o persoana se regaseste de mai multe ori in setul de date. Provocarile tehnice intalnite in realizarea proiectului au fost:

- alegearea unui set de date clar, cu persoane cu fata lizibila
- cautarea bibliotecilor si incercarea de a intelege concepetele de machine learning utilizate

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[3] S. Sriramprakash, Vadana D Prasanna, O.V. Ramana Murthy,Stress Detection in Working People,Procedia Computer Science,Volume 115,2017,Pages 359-366,ISSN 1877-0509

[4] Zhiming Xie et al 2019 J. Phys.: Conf. Ser. 1395 012006

[5] P Ramesh Naidu et al 2021 J. Phys.: Conf. Ser. 2089 012039

[6] Nijhawan, Tanya, Girija Attigeri, and T. Ananthakrishna. "Stress detection using natural language processing and machine learning over social interactions." Journal of Big Data 9.1 (2022): 1-24