

Documentație proiect sisteme de vedere artificială

Echipa nr. 2

Acsinte Andreea

Munteanu Maria-Daniela

Grupa 1301A

Cuprins

1. Prezentarea temei.....	3
2. Etape.....	3
2.1 Realizarea unui GUI pentru încărcarea imaginilor.....	3
2.2 Crearea unui alfabet al semnelor unei mâini.....	3
2.3 Recuperarea mesajului înregistrat.....	4
2.4 Vizualizarea rezultatelor în GUI.....	6
3. Dificultăți întâmpinate în realizarea temei.....	7
4. Concluzii.....	8

1. Prezentarea temei

Obiectivul principal al acestei teme este de a realiza un sistem de vedere artificială care să permită identificarea mesajului unui alfabet definit pe baza semnelor unei mâini.

2. Etape

2.1. Realizarea unui GUI pentru încărcarea imaginilor

Prima etapă a acestui proiect a fost realizarea unui GUI pentru încărcarea imaginilor.

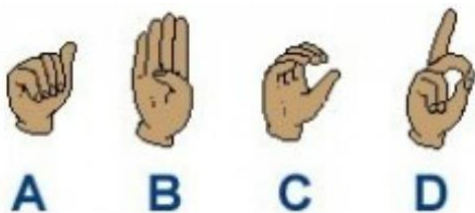
S-a utilizat biblioteca tkinter pentru a crea fereastra grafică în *Testing.py*, care va interacționa cu utilizatorul atunci când rulăm programul, iar *Data Colection.py* este codul cu ajutorul căruia am salvat imaginile pentru antrenarea rețelei neuronale.

După achiziția imaginii aceasta a fost prelucrată pentru a avea o dimensiune fixată. Acest lucru este necesar în trimiterea datelor clasificatorului. Am ales ca imaginea să aibă dimensiunea 300x300.

Rețeaua neuronală reprezintă un model matematic și computațional utilizat pentru a rezolva probleme complexe prin recunoașterea de modele și înțelegerea datelor. Procesul prin care o rețea învață să rezolve o problemă specifică se numește antrenare. Antrenarea se realizează prin prezentarea repetată a datelor de antrenament și ajustarea ponderilor în funcție de erorile generate de diferența dintre ieșirile prezise și cele dorite. Procesul constă în alimentarea rețelei neuronale cu imagini etichetate cu punctele cheie ale mâinii, și ajustarea parametrilor rețelei astfel încât să învețe să recunoască și să le localizeze în mod corect. Antrenarea implică un set imens de date și algoritmi de optimizare care ajustează ponderile rețelei pentru a minimiza erorile dintre predicțiile rețelei și etichetele reale. După ce este antrenată, aceasta poate fi folosită pentru a prezice sau a clasifica noi exemple de date care nu au fost văzute anterior.

2.2. Crearea unui alfabet al semnelor unei mâini

Ne-am propus ca în această aplicație să definim primele patru litere din limbajul semnelor.

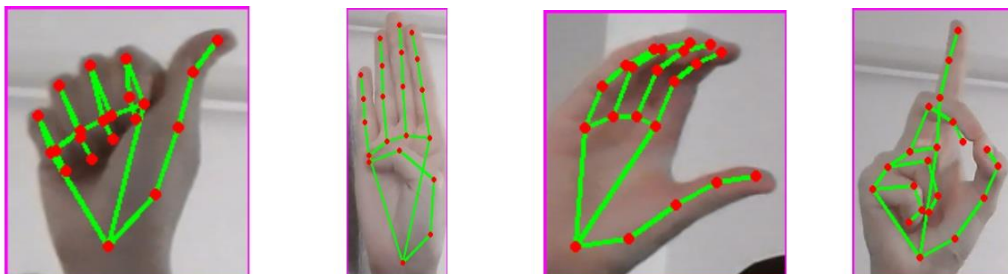


Imaginile obținute la pasul anterior au fost repartizate în foldere distincte A, B, C, D, creându-se astfel setul de antrenament. Utilizându-se biblioteca de învățare automată (TensorFlow, tipul modelului de conversie Keras) se antrenează modelul de clasificare pe baza acestor semne ale mâinii. Modelul antrenat a fost salvat împreună cu etichetele asociate cu fiecare semn într-un fișier pentru a fi utilizat ulterior.

TensorFlow este o bibliotecă care oferă un cadru flexibil pentru construirea și antrenarea rețelelor neuronale. Aceasta se bazează pe conceptul de flux de date, în care operațiile sunt reprezentate ca grafuri direcționate, numite grafuri de calcul. În aceste grafuri, tensorii (matrici

multidimensionale) circulă între nodurile grafului, realizând operații matematice și transformări de date. TensorFlow este o platformă puternică pentru calcul numeric.

Keras este o bibliotecă ușor de utilizat pentru a construi modele de învățare automată. Scopul principal al Keras este de a oferi o interfață simplă și intuitivă pentru construirea, antrenarea și evaluarea modelelor de învățare automată.

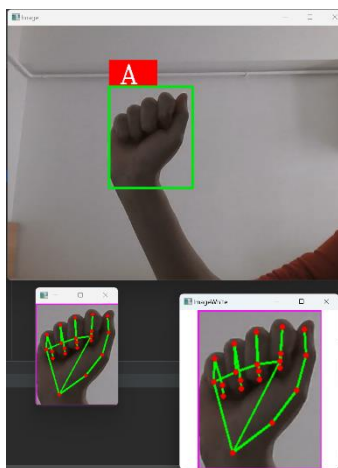
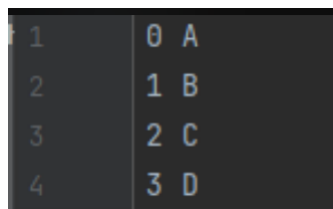


Pentru antrenarea modelului s-au încărcat imaginile la următorul link [Image Model - Teachable Machines](#). Am obținut două fișiere, unul cu modelul (pe care l-am folosit în acest proiect) și altul cu etichetele.

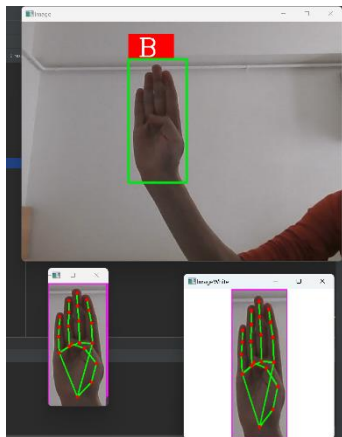
2.3. Recuperarea mesajului înregistrat

Recuperarea mesajului înregistrat de clasificator poate fi realizată prin identificarea predicțiilor făcute și asocierea lor cu etichetele corespunzătoare.

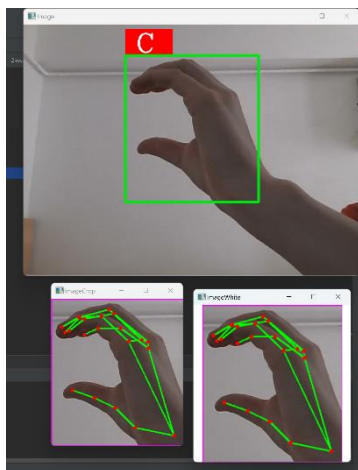
Pentru cazul literei A putem observa că cea mai mare probabilitate se asociază primei etichete, care corespunde indexului 0. Același lucru îl putem observa la toate celelalte litere.



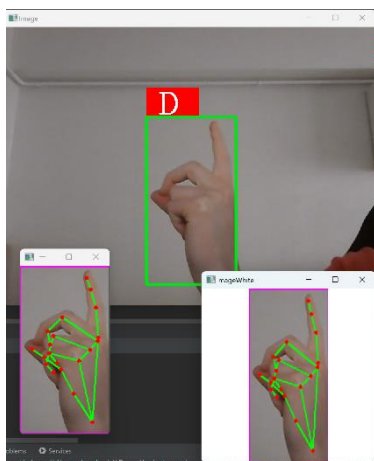
```
Testing
[0.99999976, 4.0699936e-07, 1.5566962e-07, 1.8466956e-06] 0
[0.99999595, 3.7385612e-07, 1.7665609e-07, 3.545888e-06] 0
[0.99999726, 3.4353928e-07, 1.044303e-07, 2.3062476e-06] 0
[0.99999595, 5.5873164e-07, 1.1993858e-07, 3.3520605e-06] 0
[0.99999607, 3.6022175e-07, 1.7737425e-07, 3.472208e-06] 0
```



```
Testing
[2.7018402e-06, 0.77777778, 0.00207000e-07, 1.1007101e-06] 1
[4.92896e-06, 0.99999344, 2.3310672e-07, 1.3846865e-06] 1
[3.1892507e-06, 0.9999962, 1.345849e-07, 5.3167156e-07] 1
[5.3927365e-06, 0.99998593, 1.9841332e-07, 8.43354e-06] 1
[9.237343e-06, 0.9999869, 1.9763958e-07, 3.6670726e-06] 1
[6.5327924e-07, 0.99999833, 1.2749356e-07, 8.41581e-07] 1
```



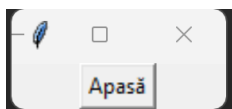
```
Testing
[4.7793790e-06, 4.0204700e-06, 0.77777778, 1.4714072e-06] 2
[5.4235247e-08, 5.7202495e-08, 0.9999999, 1.0288858e-08] 2
[2.3052154e-08, 3.273992e-08, 0.9999999, 5.431879e-08] 2
[8.4025864e-08, 6.217271e-08, 0.99999976, 2.757687e-08] 2
[2.350969e-08, 3.7286615e-08, 1.0, 2.1228503e-08] 2
[1.5238787e-07, 2.1622463e-07, 0.99999964, 8.151929e-08] 2
```



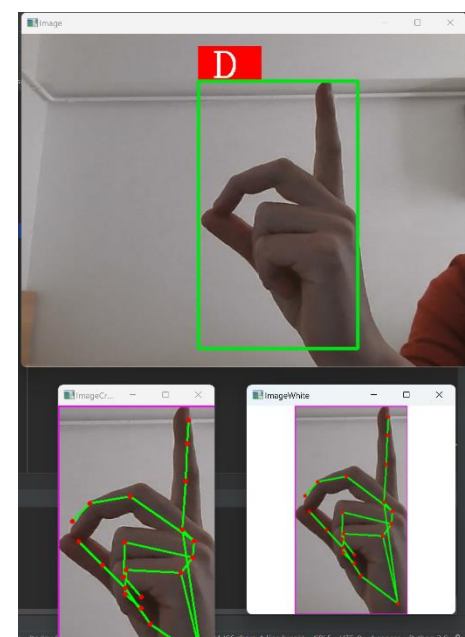
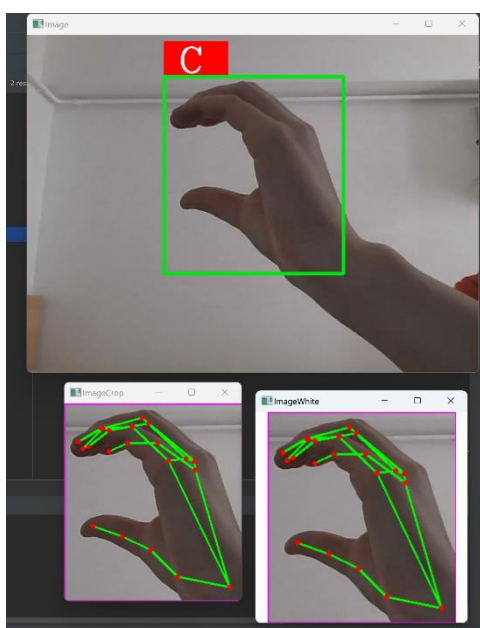
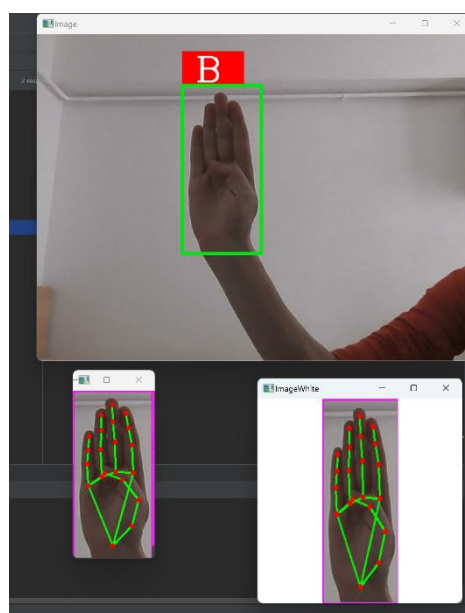
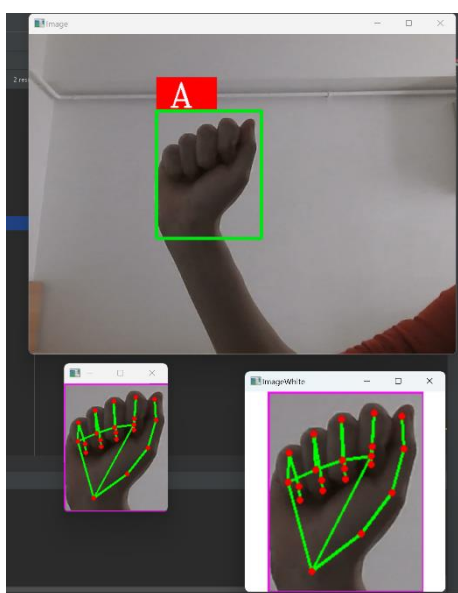
```
[2.77110407e-06, 0.021000707, 0.0010070000, 0.7771020] 0
[3.6322046e-05, 0.011815175, 0.00045967705, 0.98768884] 3
[6.1950625e-05, 0.02159286, 0.0012791553, 0.977066] 3
[0.00014939609, 0.062855534, 0.00095087406, 0.9360443] 3
[0.000101116966, 0.06784634, 0.0010307824, 0.9310217] 3
[6.3421765e-05, 0.018824598, 0.00052737637, 0.9805846] 3
```

2.4. Vizualizarea rezultatelor în GUI

Prin apăsarea butonului vor apărea ferestrele grafice conținând imaginea în timp real cu eticheta corespunzătoare fiecărui semn definit. La a doua apăsare se va opri execuția programului și vor fi închise toate ferestrele.

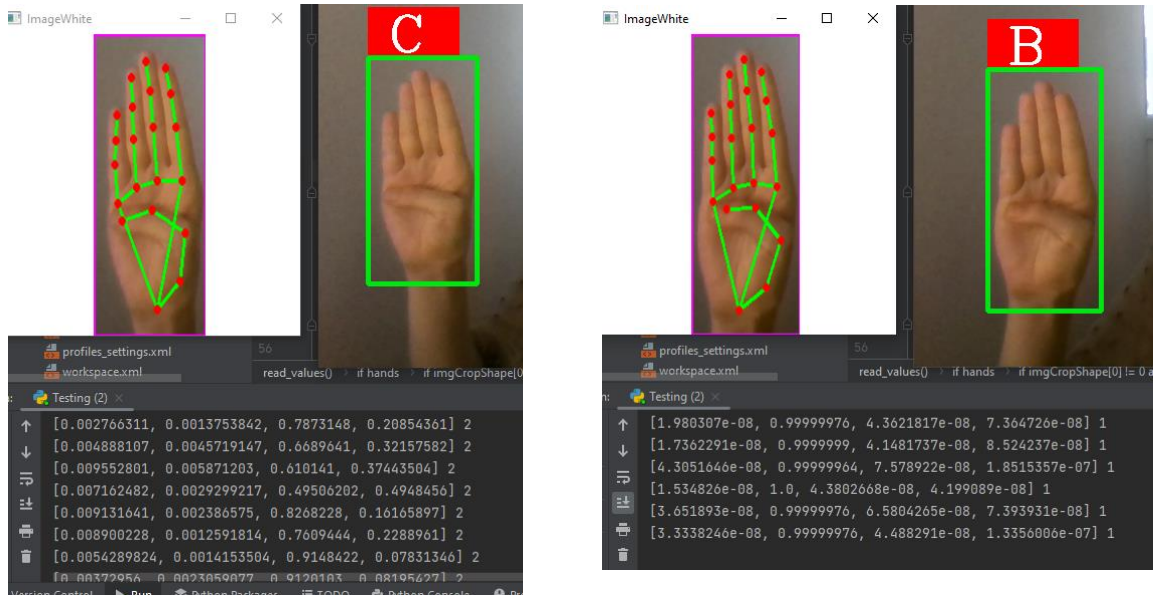


Rezultatele se pot observa în timp real în GUI, astfel încât orice modificare detectată a mesajului poate fi vizualizată.



3. Dificultăți întâmpinate în realizarea temei

Am întâmpinat afișarea eronată a predicției din cauza faptului că în metoda *getPrediction* dădeam ca parametru imaginea inițială. Corectarea a constat în înlocuirea acestui parametru cu imaginea redimensionată.



În imaginea din stânga de mai sus se observă faptul că semnul indicat de clasificator este C, dar cel definit de noi este de fapt B. Imaginea din dreapta arată rezolvarea acestei probleme.

De asemenea, în procesul de colectare al datelor am observat că este necesar că atunci când prelucram litere B și D, înălțimea imaginilor rezultate era mai mare decât înălțimea imaginii standard ($\text{imgSize} = 300$), iar acest lucru genera o eroare. O eroare similară am întâlnit și în cazul în care aveam lățimea mai mare decât cea a imaginii standard.

```
imgSize = cv2.resize(imgCrop, (wlat, imgSize))
cv2.error: OpenCV(4.5.5) D:\a\opencv-python\opencv-python\opencv\modules\imgproc\src\resize.cpp:4052: error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'
[ WARN:009.084] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) 'anonymous-namespace':::SourceReaderCB::~SourceReaderCB terminating async callback
```

Pentru a rezolva această eroare a fost necesar să obținem o imagine uniformă și standardizată a mâinii indiferent de formă sau orientare (de tip portret sau aspect orizontal). Astfel am redimensionat imaginea având în vedere păstrarea unui raport de aspect.

4. Concluzii

Concluzia proiectului constă în realizarea unui sistem de vedere artificială capabil să identifice mesajul unui alfabet definit pe baza semnelor unei mâini. Acest sistem folosește tehnici de detecție și urmărire a mâinilor, algoritmi de clasificare a semnelor și interacțiunea cu un GUI pentru afișarea rezultatelor.

Pe parcursul dezvoltării proiectului s-a folosit ca limbaj de programare Python, utilizându-se biblioteci și framework-uri precum OpenCV, TensorFlow și Keras pentru implementarea funcționalităților dorite.

Prin acest proiect s-a demonstrat modul în care tehnologiile de vedere artificială pot fi utilizate pentru a ca omul să poată interacționa cu mașini prin intermediul gesturilor și semnelor mâinilor.