**Retea MTCNN pentru detectarea fetelor din poza**

**(fara nume, numai fete)**

# Introducere

Stiu ca telul este sa detectam si numele. Aceasta poate fi facuta cu mai multe retele neuronale, una din ele, pe care am si incercat-o, este dezvoltata de Google si se numeste resnet. Cele mai multe din aceste retele neuronale trebuie sa primeasca la antrenare imagini ce contin numai o singura fata (si absolut nimic altceva - nici un alt obiect sau ceva). De asemenea, toate imaginile trebuie sa fie de dimensiune 160 de pixeli pe 160 de pixeli.

# Coduri

Numele codurilor MTCNN:

* *mtcnn\_extract\_faces.py, mtcnn\_extract\_faces.ipynb* fac pregatirea imaginilor pentru a fi introduse in resnet pentru antrenare (detefctarea fetelor din poza).
* *mtcnn\_extract\_and\_change\_faces\_from\_directory\_finalV.ipynb*, *mtcnn\_extract\_and\_change\_faces\_from\_directory\_finalV.py* fac ce fac codurile de mai sus si ceva in plus. Fiecare imagine sufera la intamplare niste transformari, dupa care fetele sunt detectate din aceste imagini transformate.

## Instructiuni mtcnn\_extract\_faces

Asa cum am explicat si in introducere, codurile *mtcnn\_extract\_faces.py* si *mtcnn\_extract\_faces.ipynb* iau ca input path-ul unui folder in variabila *data\_dir*. Acest folder poate contine subfoldere in care sunt imagini. Imaiginile ce contin numai fetele, vor fi puse in folderul din path-ul representat de variabila *output\_dir* . Structura folderului cu imaginile fetelor este exact aceeasi ca a folderului de input, iar imaginile au ca nume extensia *\_face*.

Imaginile din folder-ul de input (*data\_dir=r"C:\Users\mihnea.andrei\Python scripts\resnet\_andrei\images"* in acest caz) sunt citite cu libraria openCV, transformate in RGB (asa cum vede ochiul uman), trecute prin MTCNN, care ne spune cu precizie foarte buna chenarul de pixeli in care este fata (MTCNN da practice 4 numere daca gaseste o fata: primele doua sunt coordonatele pixelului din stanga sus a chenarului ce contine fata iar ultimele doua sunt coordonatele pixelului din dreapta jos a chenarului ce contine fata). Folosind aceste coordonate ale chenarului ce contine fata, se taie imaginea originala si se salveaza intr-un folder cu numele persoanei careia ii apartine fata, in interiorul caruia se afla fisierul .jpg cu fata.

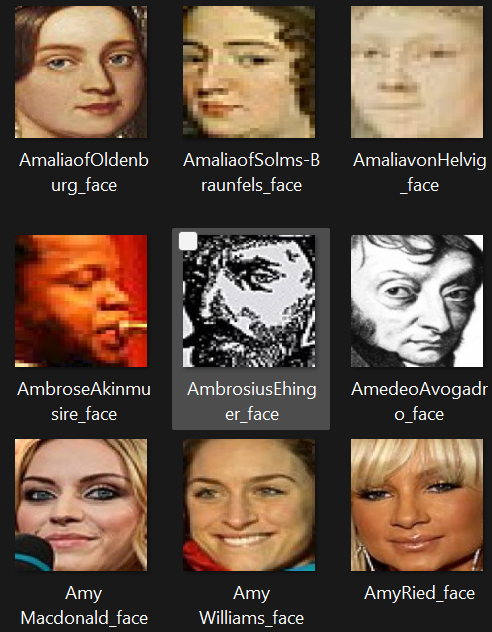
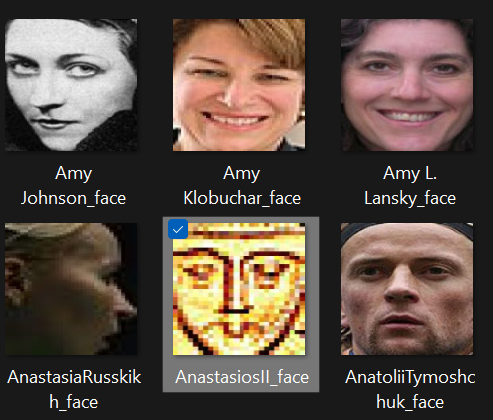
Aceasta configuratie a folderelor si fisierelor este necesara deaorece functia *ImageLoader* care o sa dea aceste imagini retelei de recunoastere facial numita resnet necesita aceasta structura. Acest lucru este datorita faptului ca este posibil ca, in general, sa avem mai mult de o imagine a unei personae astfel incat reteaua de recunoastere facial ar invata mai bine fata acelei personae. Dar, in general, atunci cand platforma va incepe, nu aveti decaot o poza a unui cetatean roman. Pe masura ce platforma merge, oamenii se inregistreaza, iar fetele recunoscute facial, pot fi salvate si reintroduse in retea pentru o invatara si mai buna. Adica, daca eu imi deschid cont pe [www.romania.gov.ro](http://www.romania.gov.ro), si recunoasterea facial isi da seama ca sunt eu, atunci o sa salveze o poza de la crearea contului in fisierul dedicat numelui meu (sau CNP-ului meu). De altfel, poate salva o poza de fiecare data cand ma recunoaste la logare in fisierul dedicate numelui meu. Aceste noi poze (impreuna cu toate ale tuturor cetatenilor care si-au creat cu success cont), reintra in antrenarea retelei neuronale, care se poate imbunatati astfel in mod continuu, fara a scrie nici macar o linie de cod noua.

## Cod mtcnn\_extract\_faces



## Rezultat mtcnn\_extract\_faces

Reteaua MTCNN nu numai ca este incredibil de rapida (proceseaza aproximativ 1100 de imagini in 7 minute), dar este si incredibil de exacta. Nu numai ca gaseste fetele dintr-o poza, dar poate gasi si fetele dintr-o pictura:



## Instructiuni *mtcnn\_extract\_and\_change\_faces\_from\_directory\_finalV*

Acest cod foloseste tot MTCNN pentru a extrage fetele. Singura diferenta este ca imaginea originala trece prin niste transformari aleatoare, si dupa aceea, fata este extrasa. Asadar, codul este foarte asemanator cu am vazut mai devreme.

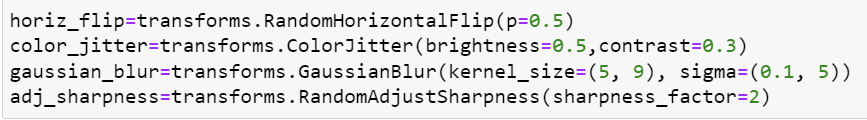
Aceasta versiune este facuta pentru a crea impresia de mai multe poze pentru aceeasi persoana, in caz ca reteaua neuronala are nevoie de mai multa antrenare. Ideal, am avea mai multe poze pentru aceeasi persoana. In cazul cetatenilor romani, nu avem decat una initial: cea din buletin. Ulterior, o data ce o persoana este identificata la inregistrarea pe platforma, o imagine este salvata in fisierul cu numele persoanei. Pozele din buletin, impreuna cu cele noi, sunt trecute prin acelasi process de detectare a fetei si de re-antrenare a retelei neuronale ce recunoaste numele pentru a obtine o acuratete mai mare.

## Cod

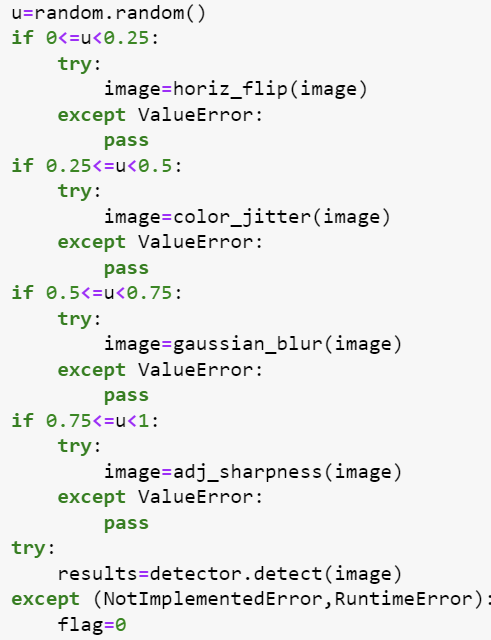
Fiecare imagine din folder-ul de input este trecuta in mod aleator printr-o transformare. Transformarile folosite sunt:

*  - care schimba unghiul din care este facuta poza (stanga sau dreapta).
* - care schimba contrastul si luminozitatea imaginii in acest caz
*  - creeaza impresia falsa de lipsa de focalizare a camerei care a facut poza
*  - creeaza impresia de camera ce au acuratete mai mare sau mai mica.

Aceste transformari pot fi memorate in niste variabile:



Dupa aceea, citesc fiecare imagine din folder-ul sursa de *nr\_repeats* ori. Pentru fiecare repetare de citire, se aplica la intamplare una dintre transformarile de mai sus:



Dupa ce se aplica o transformare la intamplare, se detecteaza imaginea folosind MTCNN si se salveaza intr-un folder destinatie care are ca subfolder numele persoanei, iar in acest subfolder sunt fetele detectate in aceste imagini transformate.