

Department of Computer Science
Technical University of Cluj-Napoca



Sisteme Inteligente

Proiect Examen

Project title:
Blink to Word, a Real-Time Convolutional Neural Network Approach
Tool: Jupyter Notebook, Keras

Name: Adorian Mititean
Group: 30233

Profesor curs: Adrian Groza
Profesor indrumator: Roxana Szomiu



Contents

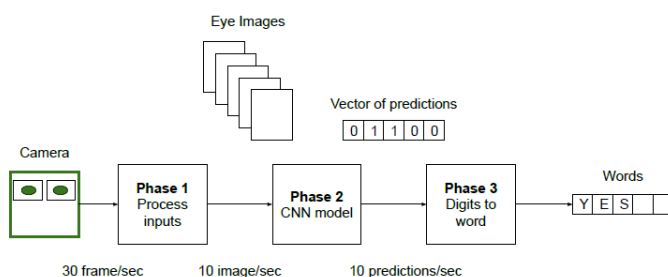
1	Prezentare Generala	3
2	Faze de design si implementare	4
2.1	Faza 1: Capturare si salvare a cadrelor	4
2.2	Faza 2: Prezicerea continutului imaginilor	4
2.3	Faza 3: Maparea	4
3	Rezultate	5

1 Prezentare Generala

Acest articol a fost dezvoltat cu scopul de a rezolva problema comunicarii cu pacientii care sufera de scleroza laterala amiotrofica care le afecteaza abilitatile motorii, lasandu-le doar posibilitatea de a clipi. Astfel se incearca obtinerea unui algoritm de machine learning bazat pe retele neuronale care sa ofere o traducere imediata a secventelor de clipire a pacientilor in cuvinte sau propozitii.

Astfel, se incearca folosirea mai multor arhitecturi ca ResNet, SqueezeNet, DenseNet si InceptionV3, cel mai bun rezultat obtinandu-se cu InceptionV3.

Procesul obtinerii pattern-urilor de clipire se realizeaza prin mai multe etape: Obtinerea si salvarea imaginilor si datelor, prezicerea continutului imaginilor, maparea rezultatelor la un dictionar de fraze comune pe care le-ar putea spune cel ce foloseste aplicatia.



2 Faze de design si implementare

2.1 Faza 1: Capturare si salvare a cadrelor

Pentru aceasta etapa am ales sa descarc acelasi set de date care este precizat si in cadrul documentului. Setul de date se regaseste la <http://mrl.cs.vsb.cz/eyedataset>. Datele sunt reprezentate prin mai multe poze de diferite dimensiuni, variind de la 65x65 pana la 120x120. Denumirile pozelor specifica informatii despre aceste poze, care sunt prezentate in detaliu in cadrul notebook-ului.

Datele au fost preluate prin intermediul unor comenzi PowerShell intr-un fisier cu extensia .csv pentru a fi usor de manipulat. In acest fisier .csv am pus pret, in special, pe partea de informatie care preciza daca ochii sunt inchisi sau deschisi.

Dupa preluarea datelor despre poze, am incarcat si pozele prin intermediul unui ImageDataGenerator.

2.2 Faza 2: Prezicerea continutului imaginilor

Pentru CNN am ales sa folosesc un model secvential cu mai multe layere. Imaginile si label-urile corespunzatoare pentru ele sunt impartite in date de antrenare si date de testare. Din cele 84864 imagini, 70000 vor fi pentru antrenare, iar restul vor fi pentru verificare.

2.3 Faza 3: Maparea

Odata cu obtinerea rezultatelor prin predictii, acestea trebuie transformate in secvente de 1 si 0 care semnifica o parte de propozitie, stocata intr-un dictionar. Daca o secventa de 0 sau de 1 este mai lunga (datorita timpului petrecut de persoana cu ochii inchisi/deschisi), aceasta va fi prescurtata la un singur numar.

S-au realizat doua functii ajutatoare pentru acest lucru convert sequence, care converteste un array intr-un string de valori de zero sau unu si CleanString, care elminia valorile extra de 0 sau 1 dintr-un astfel de sir.

3 Rezultate

Datorita complexitatii imaginilor, rezultatul obtinut de mine nu este la fel de bun ca si cel original care se apropie de 100%. Modelul pe care-l folosesc se potriveste foarte bine pentru clasificarea unor imagini similare cu cele din setul de date mnist, mai simple, (cu exceptia faptului ca ultimul layer ar trebui sa aiba 10 noduri de iesire in loc de 2). Pentru a satisface un model atat de complex ca si acesta ar fi nevoie de cel putin 14 layere.

References

[https://arxiv.org/abs/2006.01232?fbclid=](https://arxiv.org/abs/2006.01232?fbclid=IwAR3qD1mppm1edEbn3Terf_jGA_qdBAhaPFPq19hrMARul65nMpSvP2pJ3qc)

[IwAR3qD1mppm1edEbn3Terf_jGA_qdBAhaPFPq19hrMARul65nMpSvP2pJ3qc](https://arxiv.org/abs/2006.01232?fbclid=IwAR3qD1mppm1edEbn3Terf_jGA_qdBAhaPFPq19hrMARul65nMpSvP2pJ3qc)

<https://keras.io/>