

Raport Tehnic

E3 - Preprocessing Team

1. Motivatie

Aplicatia noastra isi propune sa faciliteze diagnosticarea unor afectiuni ale creierului precum Alzheimer sau dementa. Sarcina asignata echipei noastre in cadrul acestui proiect a fost aceea de a primi ca input arhive cu modele 3D ale creierului uman al pacientilor si de a le trece prin 3 tipuri de algoritmi: un algoritm de slicing care "feliaza" fiecare model 3D in imagini 2D, un algoritm de editare a imaginilor 2D obtinute (resize, crop, pad) si un algoritm de selectare a imaginilor relevante pentru diagnoza.

2. Echipa

- Hrimiuc Daniel-Marin: Scrum Master, membru al echipei de slicing, distribuirea si impartirea datelor de preprocesat pentru fiecare membru.
- Tudor Cezar: Membru al echipei de slicing.
- Gradinariu Amalia: Membra a echipei de editare a imaginilor.
- Radu-Rusu Raluca: Membra a echipei de editare a imaginilor.
- Lupu Livia: Membra a echipei de editare a imaginilor.
- Sociu Lucian: Membru al echipei de selectare a imaginilor relevante.
- Micu Razvan-Florentin: Membru al echipei de selectare a imaginilor relevante.

3. Solutia propusa

Datele de intrare ale algoritmului la stadiul de preprocesare al proiectului constau in fisiere MRI cu extensia .nii.gz ce contin arhive 3D ale unor creiere umane facute la anumite intervale de timp. Au fost furnizate pentru compilare date de tip .nii.gz intr-o multitudine de directoare ce au ca denumire codul pacientului precum si ziua in care a fost efectuata scanarea.

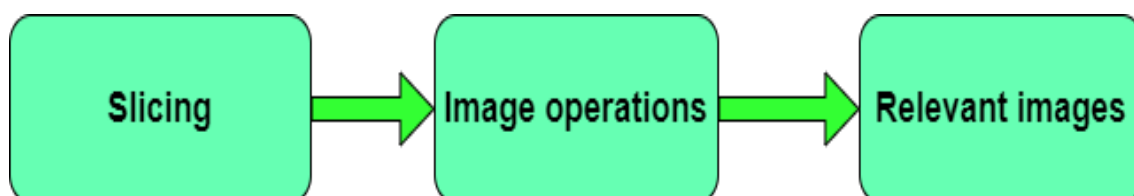
Dimensiunea totala a datelor este de 54.5 GB, avand 8732 de scanuri. In urma iterarii algoritmilor transformam arhivele 3D in imagini 2D de tip .png pentru a putea fi utilizate de urmatorul modul in scopul augmentarii.

O arhiva .nii.gz contine intre 30 si 250 de imagini 2D pe care le redimensionam, apoi eliminam o parte din imagini pentru a le pastra doar pe cele care contin informatii relevante ale pacientului. Intrucat avem foarte multe date care vor creste exponential la modulul de augmentare, am stabilit sa incercam sa pastram cat mai putine imagini dintr-un scan. Astfel eliminam aproximativ 96% din imaginile generate dintr-o arhiva 3D, ajungand in final, de la 8732 de scanuri la 51.781 de imagini png.

4. Structura sistemului de preprocesare

- slicing: -algoritmul cauta extensia ".nii.gz" sau ".nii" intr-un folder introdus ca si path, dupa care pentru fiecare fisier extrage din fiecare treimea de la mijloc (tot o imagine 2D);

- image operations: -algoritmul de redimensionare modifica imagini ca si dimensiune, maxHeight x maxWidth, pentru a nu exista cazuri in care imaginile ce urmeaza sa fie comparate in algoritmul urmator au dimensiuni diferite; de asemenea, algoritmul verifica cantitatea de alb sau negru a imaginii si o modifica in asa fel incat aceasta sa nu fie disproportionata;
- relevant images: -in cele din urma, fiecare imagine obtinuta dupa executarea primilor doi algoritmi este comparata cu un folder, in care se afla slice-uri clare si redimensionate, astfel evitand sa obtinem imagini ce nu pot fi utilizate mai departe;



5. Descrierea modului

Modulul nostru are un rol esential in dezvoltarea proiectului, si anume de a preprocesa datele de intrare primite, pentru a le transforma in date usor de editat si manevrat.

In primul rand, ca si input primim o serie de fisiere de tip nii.gz, ce sunt de fapt scanuri in format 3D, din care se doreste obtinerea de imagini 2D pe baza carora se va putea determina daca pacientul are anumite patologii neurodegenerative progresive, precum dementa.

Pentru a putea face acest lucru, incepem prin a face slicing pe imaginile 3D. Astfel, taiem imaginile 3D pe orizontala pentru a obtine imaginile 2D. Intrucat obtinem si multe imagini irelevante, spre exemplu cele din capete, cele din mijloc fiind cele mai pertinente, am ales sa impartim imaginile obtinute in 3 si sa salvam doar treimea din mijloc. Astfel, din "feliile" obtinute dintr-un scan 3D, care pot fi intre 26 si 256, salvam doar intre o treime din ele, deci intre 8 si 85.

Cu toate acestea, sunt inca foarte multe. Astfel, la etapa de redimensionare le eliminam pe cele in care albul ocupa mai mult de 28,57% din imagine si pe cele care negrul ocupa mai mult de 66,67% din imagine, aceste imagini fiind dispensabile.

De asemenea, pentru ca printre datele de iesire sa fie cat mai putine imagini neutilizabile, imaginile redimensionate sunt trecute si printr-un filtru, acestea fiind comparate cu imagini valide, iar daca depasesc un anumit procent de similitudine acestea sunt puse in folderul de output.

Pentru a vedea cat de eficient este algoritmul putem lua un set de date de intrare format din 180 scanuri in format nii.gz care au o dimensiune totala de 1.27 GB. Din acestea obtinem 8,338 de imagini 2D, in format .png, care ocupa 257 MB. Dupa ce aplicam algoritmul de redimensionare obtinem 5,553 imagini ce ocupa 182 MB. In final, in urma rularii algoritmului relevant images, inregistram un numar de 2,707 imagini, a caror dimensiune este de 231 MB.

Timpul de executie al algoritmului de slicing pentru 180 scanuri este de aproximativ 540 sec, imaginile ramase fiind trecute prin algoritmul de image operations in aproximativ 80 secunde, iar algoritmul de relevant images aplicat pe imaginile ramase ruleaza pentru aprox 50 minute, ducand timpul total de compilare la aproximativ 60 minute.

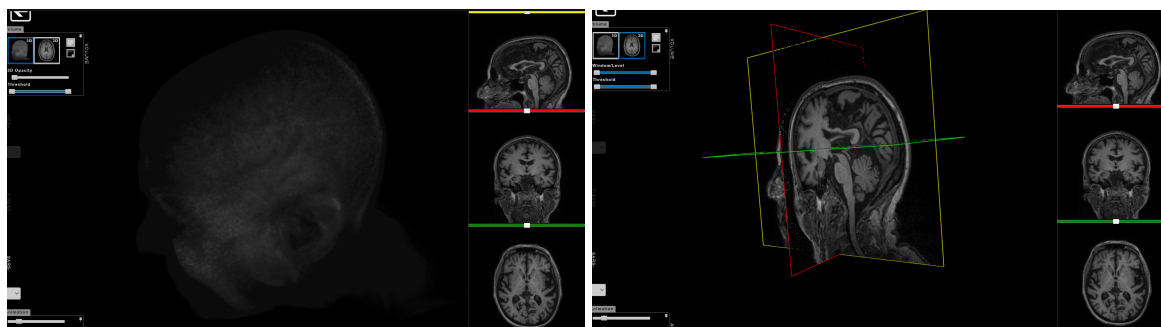


Fig. 1. Date de intrare: 3D scan si axele sale

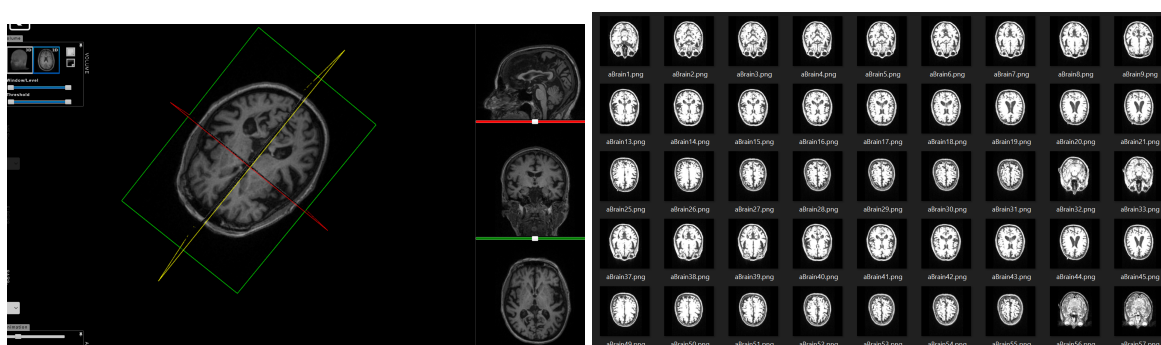


Fig. 2. Date de iesire: axa pe care luam sliceurile si imaginile rezultate dupa rulara tuturor algoritmilor

6. Operatii de baza

- slicing: Primeste ca input un folder cu mai multe fisiere de tip nii.gz sau nii. Programul parcurge axa Z a imaginii 3D si salveaza intr-un alt folder imaginile 2D ce fac parte din treimea din mijloc.
- pad: Se adauga spatiu negru in jurul imaginii
- resize: Redimensionam imaginea. Scopul: dupa slicing imaginile sunt aduse la aceeasi marime pentru a facilita operatiile urmatoare care urmeaza sa fie facute pe ele
- eliminare imagini irelevante: Imaginile care au un volum prea ridicat de alb sau negru sunt de multe ori nefolositoare si irelevante deci ne putem lipsi de ele.
- comparare imagine cu alte imagini relevante: Imaginile rezultate din slicing sunt comparate cu alte imagini despre care stim deja ca sunt folositoare. Daca anumite proprietati sunt destul de asemanatoare, programul genereaza un procent pe care il verificam pentru a stabili cat de relevanta este imaginea.