

RAPORT ETAPA 1

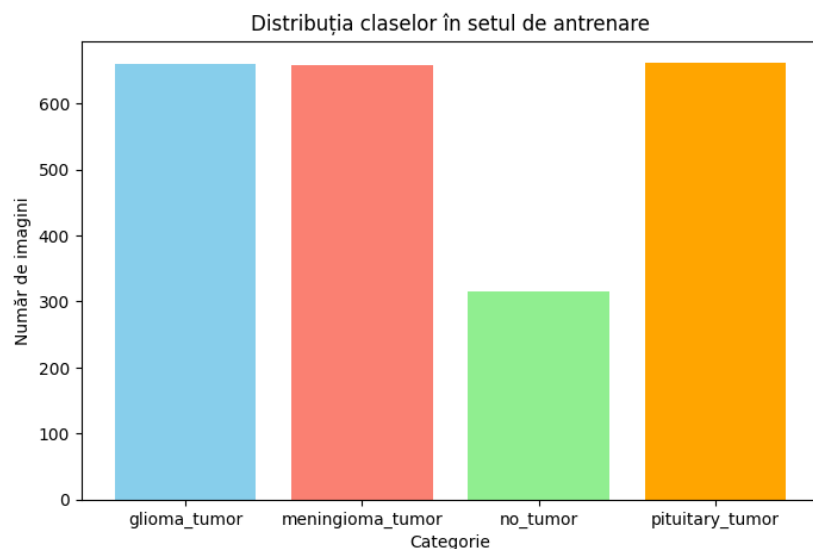
1. DISTRIBUTIA CLASELOR

Pentru crearea seturilor de validare si antrenare finale am folosit **StratifiedKfold** pe setul de antrenare initial pentru a asigura distribuita proportionala a categoriilor (labels) si evitarea sub-reprezentarii. Astfel primul fold va reprezenta setul de validare final, iar restul de 4 vor reprezenta de antrenare final, pentru a respecta proportia de 20% (validare), respectiv 80% (antrenare).

1.1 Distributia claselor in setul de antrenare:

Clasele „glioma_tumor”, „meningioma_tumor” si „pituitary_tumor” sunt bine echilibrate, cu un numar similar de imagini, fiecare avand aproximativ 600 de exemple.

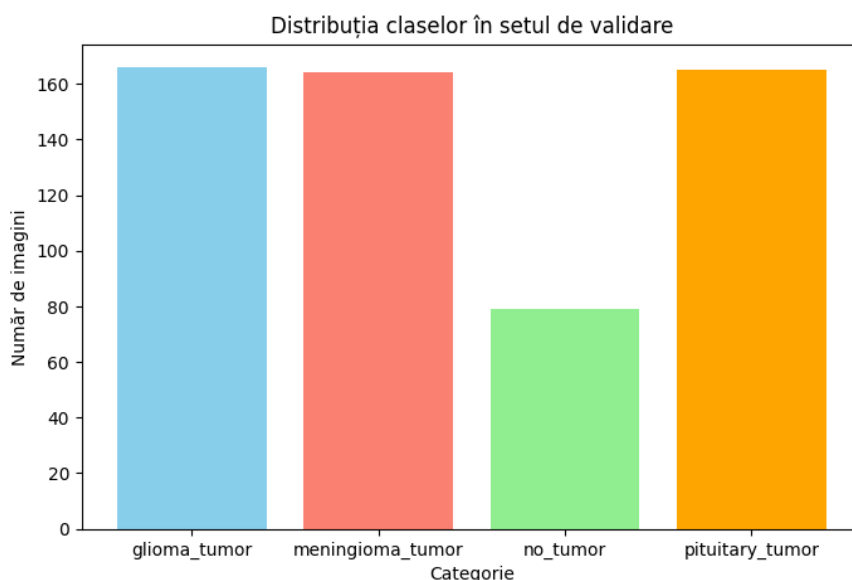
Clasa „no_tumor” este sub-reprezentata in comparatie cu celelalte, avand un numar mult mai mic de imagini (~300).



1.2 Distributia claselor in setul de validare:

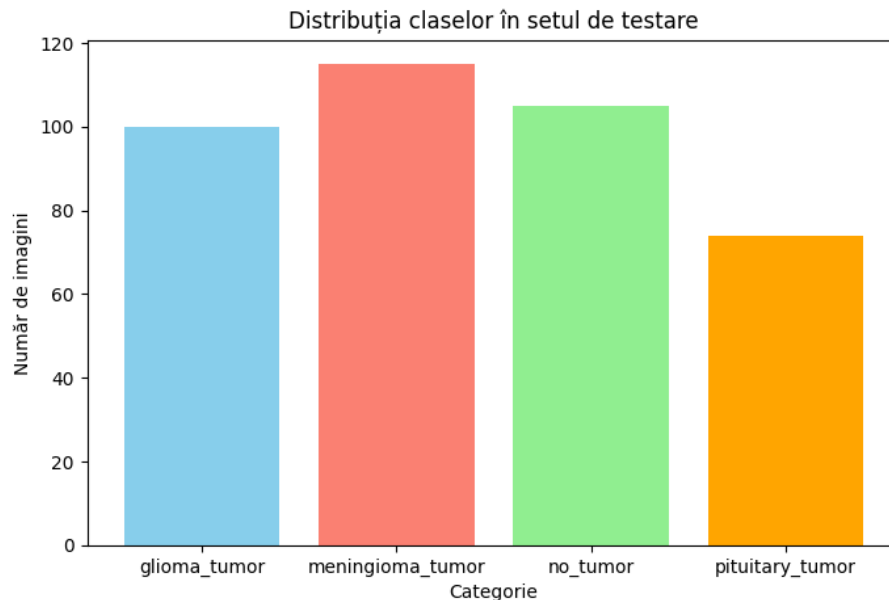
Similar setului de antrenare, clasele de tumori „glioma_tumor”, „meningioma_tumor” si „pituitary_tumor” sunt bine echilibrate.

Clasa „no_tumor” este din nou sub-reprezentata, avand aproape jumatate din numarul de imagini comparativ cu celelalte clase.



1.3 Distribuția claselor în setul de testare:

În setul de testare, distribuția este relativ echilibrată între clase, dar clasa „pituitary_tumor” are ușor mai puține imagini decât celelalte categorii.



1.4 Concluzie

Setul de date prezintă un dezechilibru între clase, în special pentru clasa „no_tumor”, care este sub-reprezentată în setul de antrenare și validare. Acest dezechilibru poate afecta performanța modelului, în special în capacitatea sa de a recunoaște corect această clasă, deoarece va fi antrenat pe mai puține exemple din această categorie.

1.5 Variante de remediere

Pentru a echilibra setul de date și a îmbunătăți performanța modelului, putem considera următoarele tehnici:

a) Augmentarea datelor pentru clasele sub-reprezentate:

Aplicarea unor tehnici de augmentare (ex. rotație, scalare, translație, flip orizontal, ajustarea contrastului) pe imaginile din clasa „no_tumor” poate crește volumul de date pentru această clasă, permițând modelului să se antreneze pe mai multe exemple diverse din această categorie.

b) Supra-esantionarea (Oversampling):

Replicarea imaginilor din clasa sub-reprezentată în setul de antrenare poate ajuta la echilibrarea claselor. Deși acest lucru poate introduce o redundanță în setul de date, poate fi eficient pentru modele sensibile la distribuția claselor.

c) Tehnici bazate pe generarea de imagini sintetice (GAN-uri):

Utilizarea rețelelor generative adversariale (GAN) pentru a crea noi imagini sintetice ale clasei „no_tumor” ar putea îmbunătăți echilibrul fără a adăuga exemple repetitive.

d) *Ponderarea claselor:*

În loc să modificăm datele, putem ajusta pierderea (loss function) a modelului pentru a penaliza mai mult erorile pentru clasa „no_tumor”, astfel încât modelul să acorde mai multă atenție acestor exemple sub-reprezentate.

Implementarea uneia sau mai multor dintre aceste tehnici ar putea ajuta la crearea unui set de date mai echilibrat și la îmbunătățirea performanței generale a modelului în recunoașterea fiecărei clase.

2. Analiza Vizuala a Imaginilor din Datasetul Brain Tumor Classification (MRI)

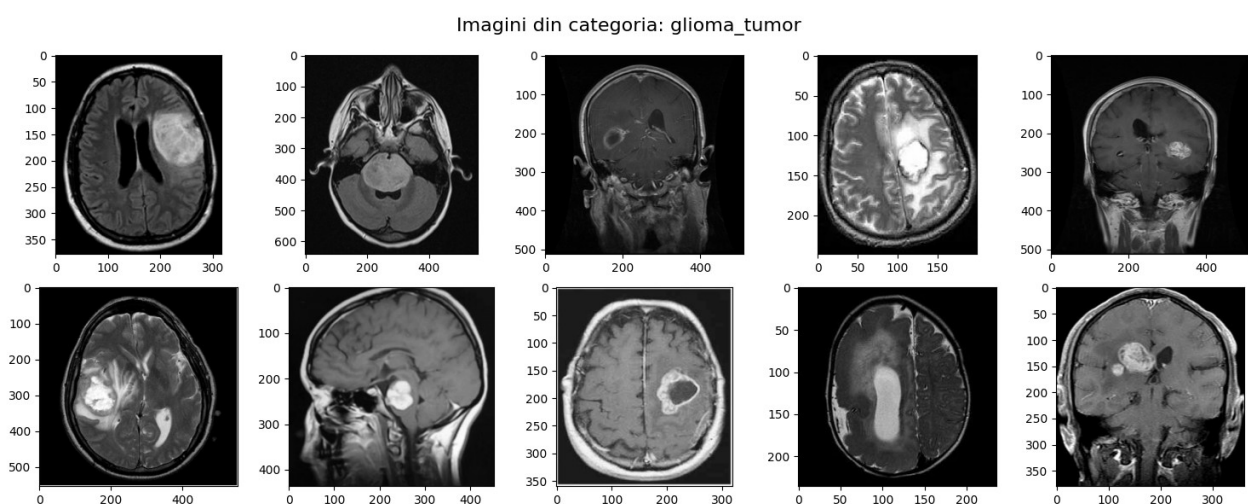
2.1 Observatii pentru fiecare categorie de tumori

a. *Glioma Tumor*

Variabilitate internă: Imaginile din categoria „glioma_tumor” arată o variabilitate semnificativă în ceea ce privește forma și dimensiunea tumorii. În unele cazuri, tumorile sunt bine delimitate și de formă rotundă, în timp ce în altele au contururi neregulate.

Textura și intensitate: Tumorile gliomice prezintă o textură inegală și pot apărea cu intensități diferite, unele fiind mai luminoase, în funcție de structura internă și de țesutul înconjurător.

Dificultăți posibile pentru clasificare: Gliomele pot avea caracteristici similare cu alte tipuri de tumori, mai ales în cazurile în care nu sunt clar delimitate.



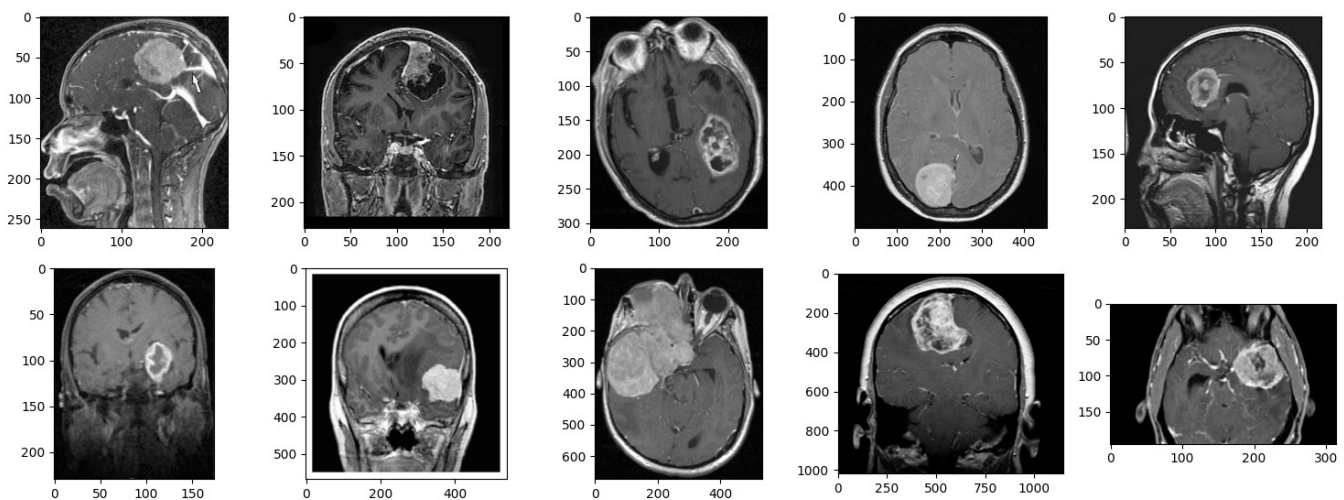
b. Meningioma Tumor

Variabilitate interna: Tumorile din aceasta clasa par sa aiba o forma mai uniforma si sunt de obicei bine delimitate, fiind vizibil separate de tesuturile inconjuratoare.

Textura si intensitate: Imaginile arata ca meningiomele sunt adesea bine vizibile, avand o structura omogena si o intensitate uniforma, ceea ce le face usor de distins.

Dificultati posibile pentru clasificare: In unele cazuri, meningiomele pot parea similare cu pituitary tumor, ceea ce ar putea duce la confuzii.

Imagini din categoria: meningioma_tumor



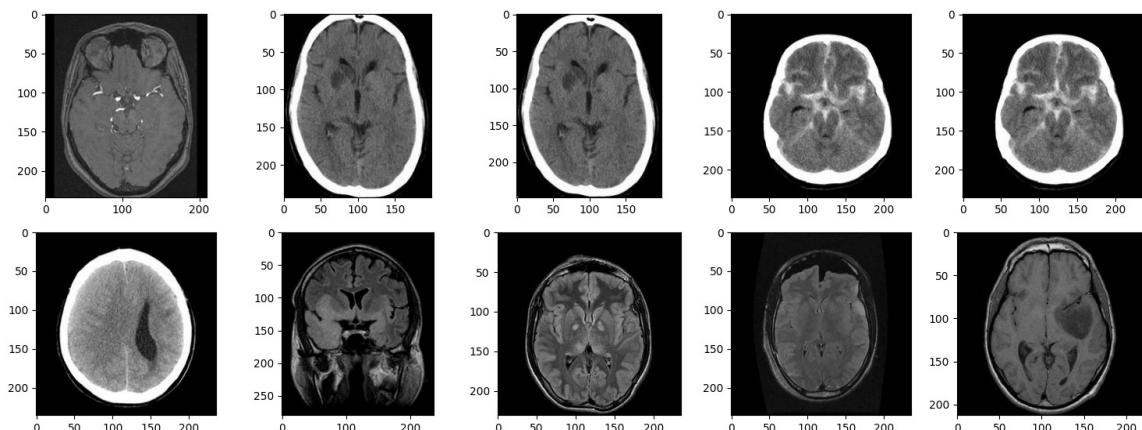
c. No Tumor

Variabilitate interna: Imaginile din categoria „no_tumor” nu prezinta anomalii evidente, iar structura creierului este normala.

Textura si intensitate: Fara tumori prezente, aceste imagini au o intensitate si o textura uniforme si sunt mai clare din punct de vedere al detaliilor anatomice.

Dificultati posibile pentru clasificare: Faptul ca aceste imagini sunt destul de clare le face relativ usor de clasificat. Totusi, imaginile de calitate scazuta sau cele care au artefacte pot complica acest proces.

Imagini din categoria: no_tumor

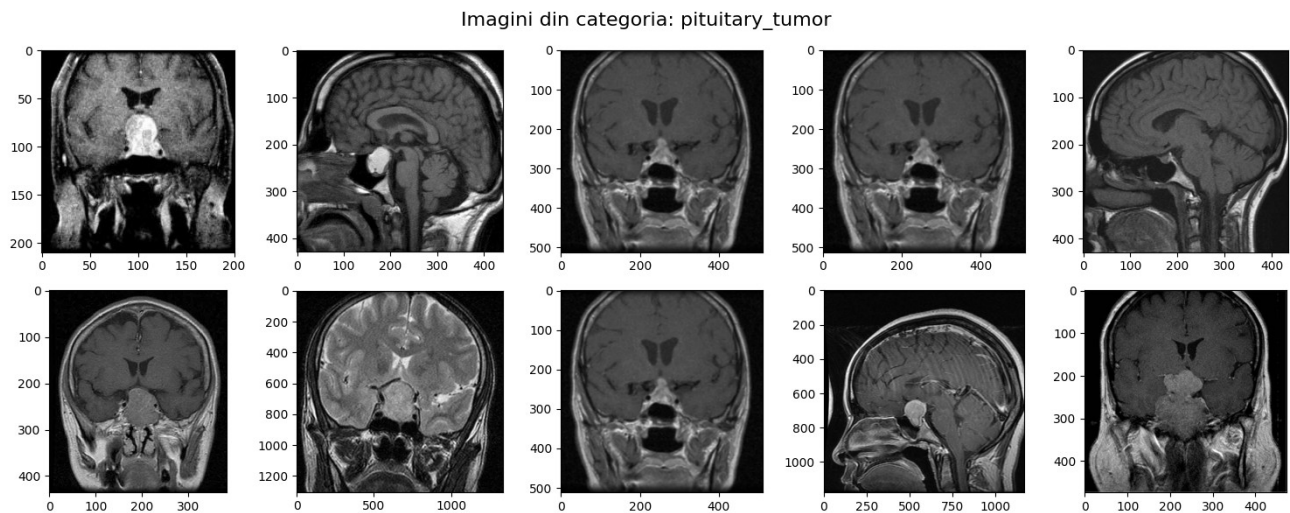


d. Pituitary Tumor

Variabilitate interna: Tumorile din categoria pituitary_tumor sunt relativ mici si localizate in zona glandei pituitare, ceea ce poate ingreuna identificarea lor in comparatie cu tumorile mai mari.

Textura si intensitate: Au o textura omogena si sunt de obicei mai intunecate decat alte tipuri de tumori, fiind usor de confundat cu structuri normale ale creierului in aceasta regiune.

Dificultati posibile pentru clasificare: Datorita dimensiunii mici si a localizarii, aceste tumori ar putea fi confundate cu imagini fara tumori (no tumor), mai ales daca sunt slab vizibile.



2.2 Similaritati intre clase si potensiale dificultati

- **Similaritati intre glioma_tumor si meningioma_tumor:** In functie de intensitate si de forma tumorii, exista cazuri in care o glioma cu margini clare ar putea arata similar cu un meningiom. Aceasta ar putea reprezenta o problema pentru modelul de clasificare.

- **Similaritati intre pituitary_tumor si no_tumor:** Tumorile pituitare, datorita dimensiunii lor mici, pot fi mai greu de identificat si uneori ar putea fi confundate cu imagini normale, mai ales in cazurile cu rezolutie mai scazuta.

3. Consistenta si integritatea setului de date

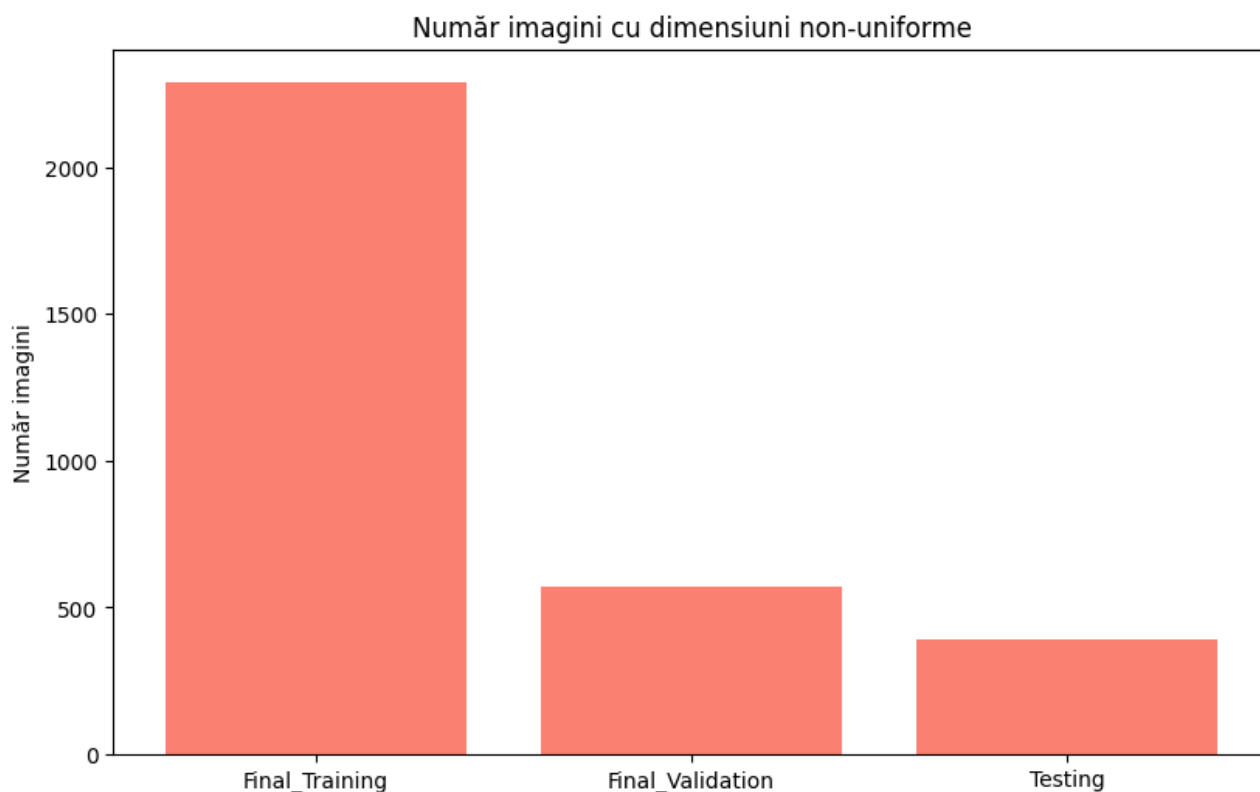
3.1 Verificarea numarului de canale:

Dupa verificarea numarului de canale a imaginilor in cele 3 dataset-uri, observam ca toate imaginile sunt RGB. Am adugat totusi o metoda de a le converti RGB daca exista alte set-uri de date care contin imagini greyscale.

3.2 Verificarea dimensiunii imaginii:

Graficul de mai jos arata distributia imaginilor cu dimensiuni diferite de 256x256 din cele 3 seturi de date.

Imaginile de dimensiuni diferite sunt redimensionate la dimensiunile dorite folosind „LANCZOS”.



3.3 Normalizarea pixelilor:

Pentru a asigura ca toate imaginile sunt pe aceeași scală de valori (adică între 0 și 1), valorile pixelilor sunt normalizate prin împărțirea lor la 255.0.

3.4 Probleme identificate:

Unele imagini aveau dimensiuni diferite de 256x256.

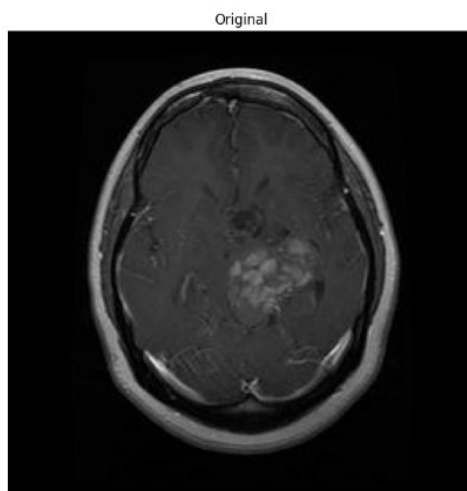
Unele imagini aveau valori ale pixelilor exprimate în diferite scale.

3.5 Procese de uniformizare aplicate:

Imaginile au fost redimensionate pentru a respecta dimensiunile standard.

Valorile pixelilor au fost normalizate pentru a fi în intervalul [0, 1].

4. Operatii de preprocesare si normalizare pentru setul de date



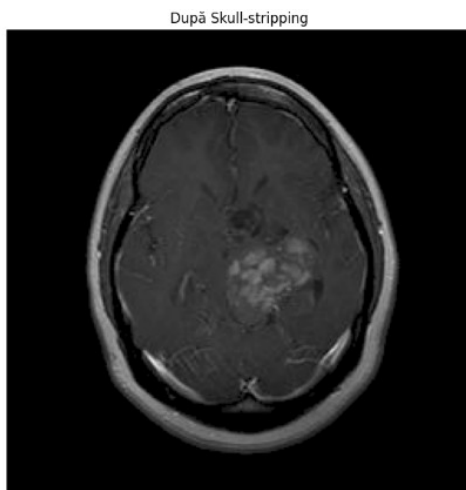
4.1 Operatiile de Preprocesare

4.1.1 Skull-Stripping (Eliminarea Craniului)

Necesitate: In imaginile cerebrale, craniul si alte structuri exterioare pot introduce zgomot si pot afecta acuratetea modelului de clasificare. Skull-stripping este o tehnica utilizata pentru a elimina aceste structuri si a pastra doar zonele relevante ale creierului.

Metoda: Am simulat aceasta tehnica prin aplicarea unui prag de intensitate asupra imaginii. Folosind un prag de intensitate (20), am creat o masca binara pentru a pastra doar zonele creierului si a elimina structurile externe.

Impact: Eliminarea craniului reduce complexitatea imaginii, imbunatatind claritatea zonei cerebrale si ajutand modelul sa se concentreze pe structurile relevante.

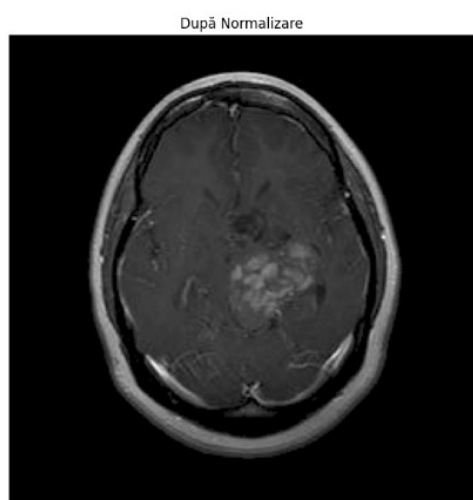


4.1.2 Normalizarea Intensitatii

Necesitate: Imaginile MRI pot varia in ceea ce priveste scala valorilor pixelilor, ceea ce poate cauza inconsistente intre imagini. Normalizarea intensitatii aduce valorile pixelilor la o scala uniforma, intre [0, 1].

Metoda: Am calculat valorile minime si maxime ale intensitatii pentru fiecare imagine, apoi am scalat valorile pixelilor intre [0, 1] si le-am rescalat la [0, 255] pentru consistenta vizuala.

Impact: Normalizarea intensitatii reduce variabilitatea in luminozitate intre imagini si asigura o scalare uniforma, astfel incat modelul sa poata procesa imaginile intr-un mod mai consistent.

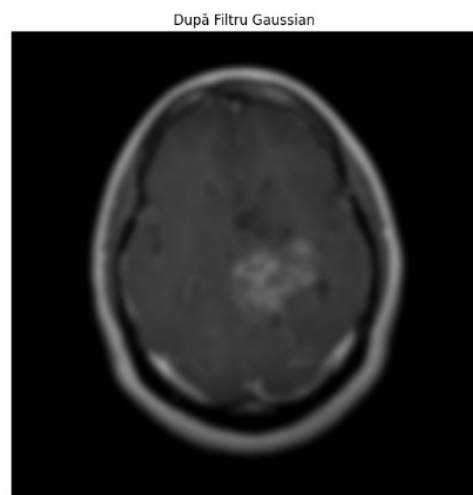


4.1.3 Filtrul Gaussian pentru Reducerea Zgomotului

Necesitate: Imaginile MRI pot contine zgomot care poate afecta detectarea si clasificarea structurilor importante. Filtrul Gaussian este o tehnica de reducere a zgomotului care estompeaza usor imaginea, pastrand in acelasi timp conturile importante.

Metoda: Am aplicat un filtru Gaussian cu o deviatie standard (σ) de 2 pentru a netezi imaginea.

Impact: Aplicarea filtrului reduce zgomotul de inalta frecventa, clarificand structurile cerebrale si facilitand detectarea marginilor tumorilor fara a compromite conturul lor.

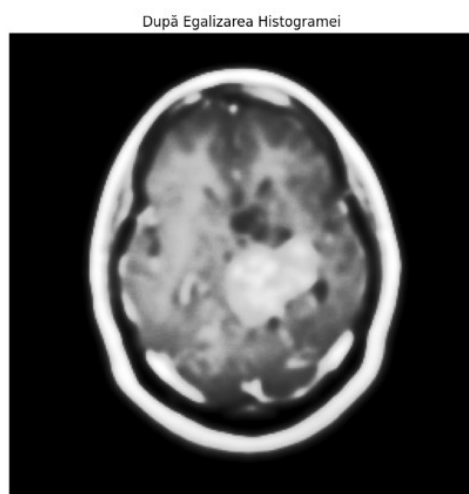


4.1.4 Egalizarea Histogramei

Necesitate: Egalizarea histogramei îmbunătățește distribuția valorilor pixelilor și crește contrastul global al imaginii, făcând zonele mai vizibile. Este o tehnică utilă în special pentru imagini medicale cu iluminare neuniformă.

Metoda: Am aplicat funcția `equalizeHist` din OpenCV pentru a obține o distribuție mai uniformă a intensităților.

Impact: Creșterea contrastului ajută la evidențierea structurii interne a creierului și face ca tumorile să fie mai vizibile, ceea ce poate ajuta modelul să se concentreze mai bine pe zonele relevante.

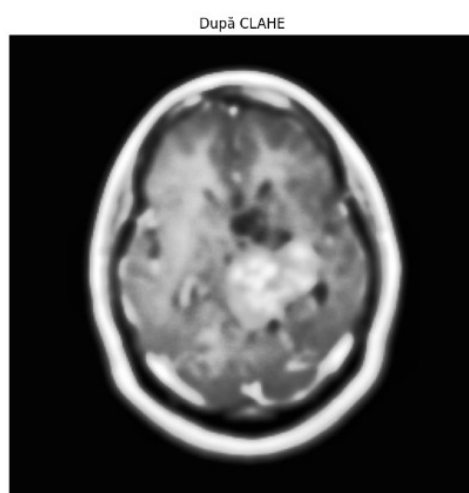


4.1.5 CLAHE (Contrast Limited Adaptive Histogram Equalization)

Necesitate: CLAHE este o tehnică avansată de îmbunătățire a contrastului care evită amplificarea excesivă a zgomotului prin limitarea clipului. Este deosebit de utilă pentru imagini MRI, unde contrastul fin poate fi esențial pentru detectarea detaliilor tumorale.

Metoda: Am utilizat CLAHE cu un `clipLimit` de 2.0 și o dimensiune a grilei de 8x8 pentru a realiza o îmbunătățire locală a contrastului în regiuni mici ale imaginii.

Impact: CLAHE accentuează detaliile locale ale imaginii și evidențiază contururile fine ale tumorilor, contribuind la o mai bună separare a structurilor relevante față de fundal.



4.1.6 Sobel (Detectarea Marginilor)

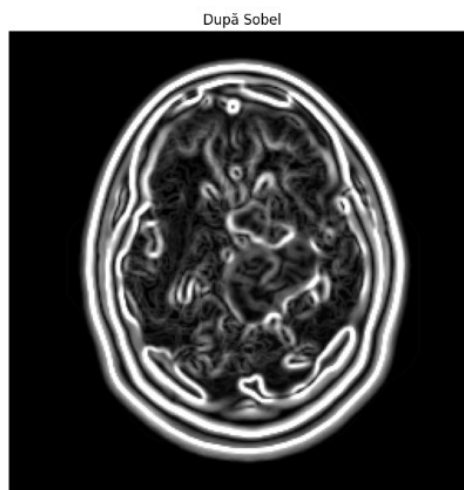
Necesitate: In procesarea imaginilor medicale, detectarea marginilor este cruciala pentru evidentierea structurilor importante, cum ar fi contururile tumorilor sau alte zone de interes. Filtrul Sobel identifica rapid schimbarile de intensitate ale pixelilor din imagine, care corespund marginilor obiectelor.

Metoda: Am aplicat filtrul Sobel pentru a calcula gradientul de intensitate al pixelilor in doua directii:

- **Directia X:** Detecteaza schimbarile orizontale (de exemplu, marginile verticale).
- **Directia Y:** Detecteaza schimbarile verticale (de exemplu, marginile orizontale).

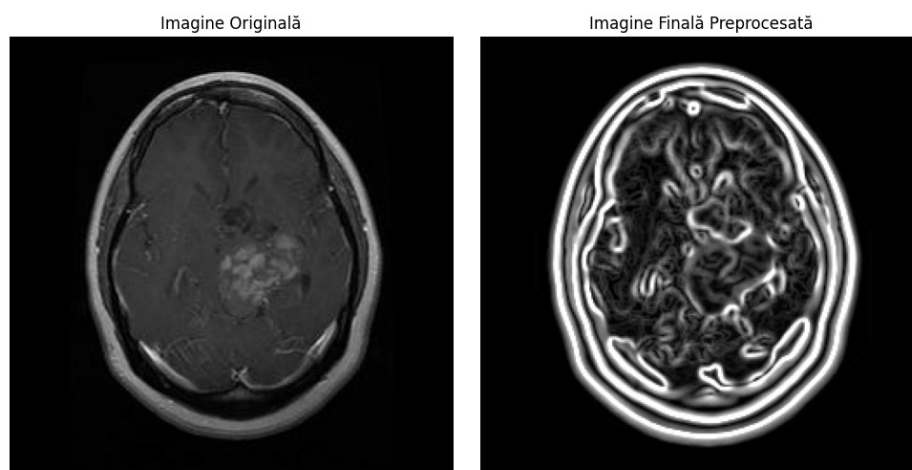
Prin combinarea gradientelor in ambele directii, am obtinut magnitudinea gradientului, care reprezinta toate marginile din imagine. Aceasta tehnica este implementata prin utilizarea convolutiei cu kernel-uri specifice Sobel pentru fiecare directie.

Impact: Filtrul Sobel evidentiaza clar contururile si structurile din imagine, ceea ce permite modelului sa se concentreze pe marginile tumorilor si sa ignore detaliile irelevante sau zgomotul din imagine.



4.2 Implementarea Finala

Am reunit toate aceste operatii intr-o clasa numita `PreprocessImageOptimized`, care aplica secvential fiecare tehnica pe imaginea initiala. Imaginea rezultata este convertita inapoi intr-un format RGB pentru vizualizare.



4.3 Concluzii si Observatii

Aplicarea acestor tehnici de preprocesare asigura ca imaginile sunt de calitate inalta si consistenta, contribuind la o performanta mai buna a modelului de clasificare a tumorilor cerebrale. Impactul vizual al fiecărei transformari a fost analizat prin compararea imaginilor rezultate cu cele originale, iar concluziile principale sunt:

Skull-stripping: Elimina structurile nerelevante si permite o concentrare mai buna pe zona cerebrala.

Normalizarea intensitatii: Standardizeaza nivelurile de intensitate, prevenind variatiile de luminozitate.

Filtrul Gaussian: Reduce zgomotul, pastrand in acelasi timp structurile esentiale.

Egalizarea histogramei: Creste contrastul global, facand structurile mai clare.

CLAHE: Accentuarea contrastului local imbunatateste detaliile importante pentru detectarea tumorilor.

Filtrul Sobel: Evidentiaza marginile esentiale, ajutand modelul sa detecteze schimbarile importante de intensitate, cum ar fi contururile tumorilor si reduce complexitatea imaginii, pastrand doar informatiile relevante legate de structura si forma obiectelor.

5. Pocesul de antrenare

5.1 Pipeline de Antrenare a Modelului

5.1.1 Incarcarea Datelor

Am folosit o clasa derivata din **Dataset**, denumita **BrainTumorDataset**, pentru a incarca si prelucra datele. Setul de date este structurat in patru categorii de tumori: glioma, meningioma, fara tumora si tumora pituitara. Datele sunt impartite in trei subseturi: antrenare, validare si testare. Dimensiunea batch-ului este **32**, fiind un echilibru intre viteza de procesare si stabilitatea actualizarilor de greutate.

5.1.2 Modelul (Arhitectura CNN)

Arhitectura rețelei neuronale convolutive utilizate este compusa din patru blocuri de straturi convolutive, fiecare urmat de activari **Leaky ReLU**, **Batch Normalization** si **pooling adaptiv** pentru reducerea dimensiunii imaginilor intermediare, astfel fiind o arhitectura modulara si scalabila, pastrandu-si complexitatea controlata.

- **Straturi Convolutionale:**
 - Incep cu filtre de dimensiuni mici si numar redus (8 pentru primul strat) si cresc exponential (64 pentru ultimul strat), cresterea numarului de filtre permitand modelului sa capteze trasaturi mai complexe la niveluri mai profunde.

- Fiecare strat foloseste un kernel de dimensiune 5×5 pentru detectarea caracteristicilor locale in imagini, fiind util mai ales pentru datele de imagistica medicala, unde detaliile locale sunt cruciale.
- **Batch Normalization** stabilizeaza distributia activarilor din retea, accelerand antrenarea si reducand riscul de gradient explosion/vanishing.
- **Pooling-ul adaptiv** reduce dimensiunile activarilor, fara a compromite integritatea informatiilor spatiale importante, permitand un control mai precis asupra dimensiunii finale a datelor care intra in straturile complet conectate.
- **Straturi Complet Conectate (Fully-Connected):**
 - Primul strat conectat complet contine **100 de neuroni**, care realizeaza o agregare a caracteristicilor extrase de straturile convolutionale.
 - Stratul final contine **4 neuroni**, corespunzatori celor 4 clase: glioma tumor, meningioma tumor, no tumor, si pituitary tumor. Iesirea acestui strat este transformata in probabilitati prin functia **Softmax** implicita in **CrossEntropyLoss**.
- **Dropout:**
 - Este introdus in primul strat complet conectat, cu o rata de **0.25**, pentru a preveni **suprainvatarea** (overfitting). Dropout functioneaza prin eliminarea aleatorie a conexiunilor neuronale in timpul antrenarii, fortand retea sa fie mai robusta si sa nu depinda excesiv de anumite conexiuni.

Arhitectura este proiectata pentru a echilibra complexitatea modelului cu capacitatea sa de generalizare, optimizand performanta fara supra-invatare, fiind suficient de profunda pentru a invata caracteristici complexe, fara a deveni prea costisitoare din punct de vedere computational.

5.1.3 Functia de Pierdere si Optimizatorul

Pentru antrenarea retelei convolutionale, s-au folosit:

- **Functia de pierdere: CrossEntropyLoss** din **torch.nn**, deoarece este potrivita pentru sarcinile de clasificare multi-clasa, calculand diferenta intre predictiile modelului si etichetele reale. Combina o functie de **Softmax** pentru normalizarea iesirilor si un **log loss** pentru calcularea erorii: daca modelul prezice o probabilitate mare pentru clasa corecta, pierderea este mica; altfel, pierderea este mare, ceea ce directioneaza optimizatorul sa ajusteze greutatea pentru a imbunatati predictiile.
- **Optimizator: AdamW**, cu rata initiala de invatare setata la 3×10^{-4} pentru convergenta treptata, este o varianta imbunatatita a algoritmului Adam, care adauga un termen de regularizare L2 (weight decay):
 - **Adam** ajusteaza dinamica ratei de invatare pentru fiecare parametru, facandu-l potrivit pentru modele complexe.

- **Weight Decay** reduce riscul ca modelul sa suprainvete pe setul de antrenare, imbunatatind generalizarea.
- **Scheduler de invatare: OneCycleLR**, unde rata de invatare creste la inceput pana la un maxim specificat (1×10^{-3}) si scade treptat spre final, este folosit pentru accelerarea antrenarii si imbunatatirea generalizarii:
 - Cresterea initiala ajuta modelul sa iasa rapid din minime locale slabe.
 - Reducerea ulterioara asigura stabilitatea optimizarii pe masura ce modelul se apropie de solutia optima.

5.1.4 Antrenarea Modelului

Antrenarea este realizata pe parcursul a **60 de epoci**, folosind un ciclu structurat de forward pass, backward pass si evaluare.

- **Forward pass:** Setul de date este trecut prin retea pentru a calcula iesirile, iar rezultatele sunt comparate cu etichetele reale folosind functia de pierdere.
- **Backward pass:**
 - Functia de pierdere determina diferenta dintre predictii si etichetele reale.
 - Gradientul erorii este calculat si propagat inapoi prin retea folosind backpropagation.
 - Optimizatorul **AdamW** ajusteaza greutatea si bias-urile pe baza gradientului pentru a minimiza pierderea.
 - Acest proces se repeta pentru fiecare batch, imbunatatind modelul treptat.
- **Evaluare pe setul de validare:** Dupa fiecare epoca, modelul este evaluat pe setul de validare pentru a monitoriza pierderea si acuratetea, prevenind supra-invatarea. Modelul cu cea mai mica pierdere pe setul de validare este salvat automat in fisierul **best_model.pth** pentru utilizare ulterioara.

5.1.5 Evaluarea modelului pe setul de test

Dupa antrenare, modelul este evaluat pe setul de test pentru a calcula:

- **Acuratetea:** Procentul de predictii corecte.
- **Precizia, Recall și F1-score:** Metode standard de evaluare a clasificarii.
- **Matricea de Confuzie:** Aceasta permite vizualizarea numarului de clasificari corecte si gresite pentru fiecare clasa.

5.2 Rezultate Experimentale (60 epoci; aplicarea preprocesarilor, fara Sobel)

5.2.1 Acuratetea si Pierderea pe Epoci

Graficul de mai jos ilustreaza evolutia pierderii si a acuratetei pentru seturile de antrenare si validare de-a lungul epocilor.

1. Loss: - pentru train si validation scade constant in primele ~10-15 epoci, indicand ca modelul invata sa minimizeze eroarea;

- pe validation fluctueaza usor dupa ~20 epoci, dar ramane aproape de train, sugerand ca modelul generalizeaza bine pe setul de validare;

- nu exista o crestere pronuntata pe validation dupa multe epoci, ceea ce ar indica supra-antrenare; in schimb, comportamentul este stabil.

2. Accuracy: - pe train si validation creste rapid in primele ~15 epoci si se stabilizeaza ulterior;

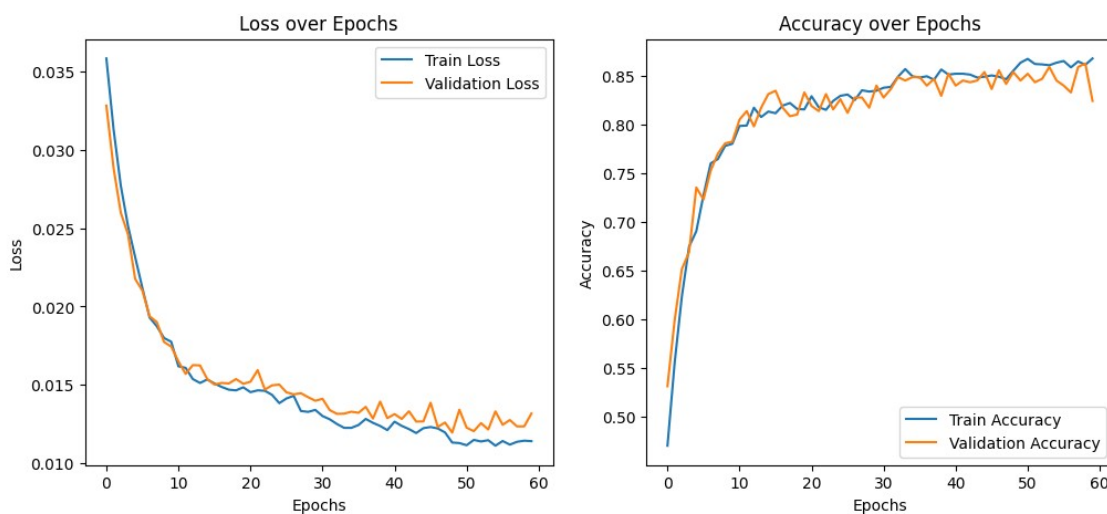
- diferenta dintre train si validation este foarte mica, ceea ce arata ca modelul nu este supra-antrenat si functioneaza bine atat pe datele de antrenare, cat si pe cele de validare.

5.2.2 Interpretare generala:

1. Performanta buna: Curbele pentru train si validation sunt apropiate atat pentru loss, cat si pentru accuracy, ceea ce indica faptul ca modelul nu sufera de overfitting sau underfitting.

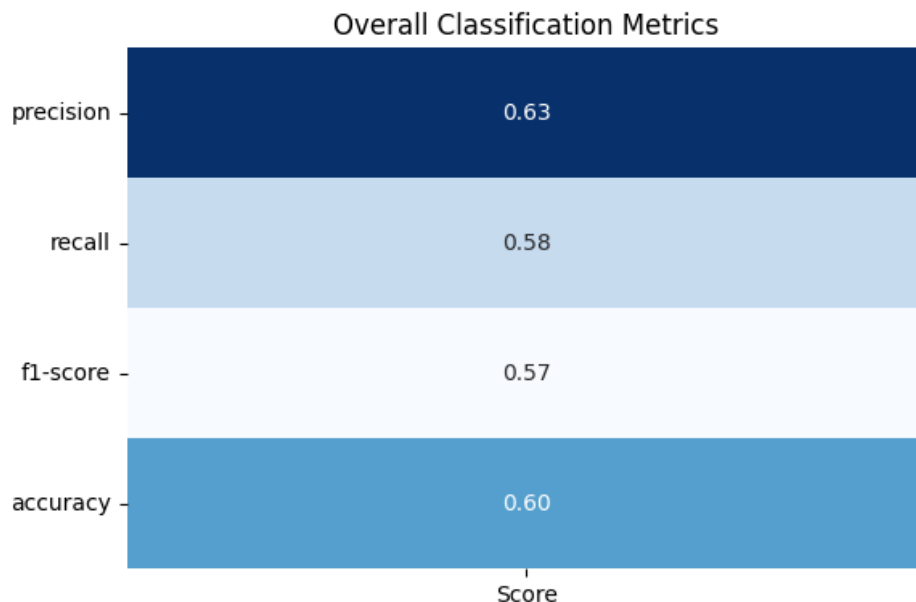
2. Convergenta: Modelul ajunge la o convergenta stabila in jurul epocii 40, cu valori consistente ale accuracy (~85%) si pierderii (~0.01-0.015).

3. Generalizare buna: Comportamentul similar pe seturile de antrenare si validare sugereaza ca modelul are o capacitate buna de generalizare.



5.2.3 Raportul de Clasificare

Mai jos este un tabel sumar al metricei de clasificare (precizie, recall, F1-score, acuratete) calculata pe setul de test:

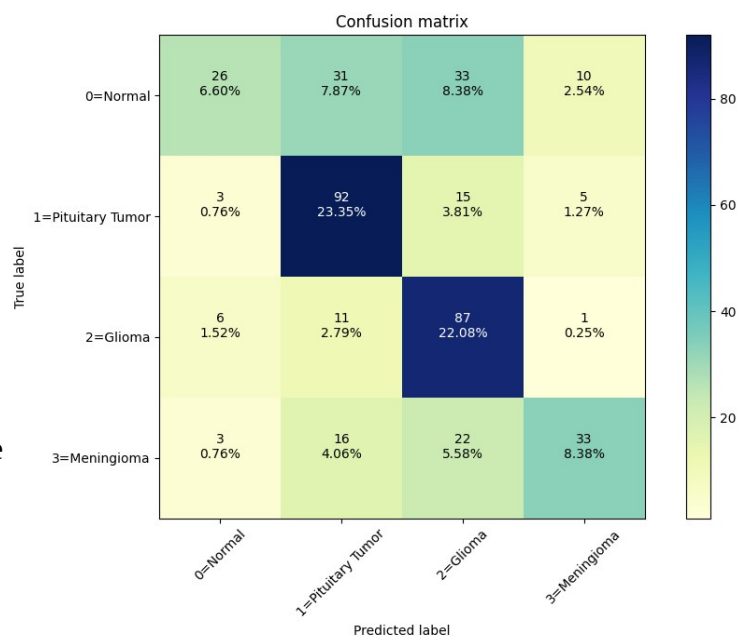


1. **Precizia** arata ca din predictiile facute, 63% au fost corecte, fapt ce indica o buna capacitate de a nu clasifica un caz negativ ca pozitiv.
2. **Recall-ul** arata ca modelul reuseste sa gaseasca 58% din toate cazurile reale de tumori. Deoarece este mai mic decat prezicia rezulta ca modelul rateaza o parte din cazurile care contin tumori.
3. **F1-score** sugereaza o performanta modesta a modelului, fiind un indicator echilibrat intre precizie si recall.
4. **Acuratetea** generala de 60% sugereaza ca modelul prezice corect putin peste jumatate din cazuri.

5.2.4 Matricea de Confuzie

Matricea de confuzie prezentata mai jos indica numarul de clasificari corecte si incorecte pentru fiecare clasa.

1. **Clasa „No tumor” este greu de identificat:** Modelul intampina mari dificultati in a identifica corect cazurile normale, ceea ce indica o caracteristica deosebita a imaginilor normale care le face dificil de separat de clasele tumorale.



2. **Performanta buna pentru „Pituitary Tumor” si „Glioma”:** Modelul are o performanta buna pentru tumorile pituitare si glioma, clasand corect majoritatea cazurilor.

3. Confuzia majora intre clase:

Clasele „**No tumor**” si „**Glioma**” au fost confundate frecvent intre ele. 33 de cazuri normale au fost clasificate gresit ca glioma, si cateva cazuri de glioma au fost clasificate ca „No tumor”.

De asemenea, exista o confuzie notabila intre „**Meningioma**” si „**Pituitary Tumor**”, precum si intre „**Meningioma**” si „**Glioma**”.