Învățare Supervizată - Etapa 2
Clasificarea Imaginilor RMN ale Creierului

Alexandru Licuriceanu
alicuriceanu@stud.acs.upb.ro

1 Cerința 1

Primul pas în implementarea pipeline-ului este alegerea unei metode pentru a face k-fold cross-validation. Metoda aleasa a fost utilizarea funcției StratifiedKFold din biblioteca Scikit-learn, deoarece aceasta păstrează proporțiile claselor din setul de date după care se face împărțirea. Pentru antrenare am folosit următorul model și hiperparametrii aferenți:

Table 1: Caracteristicile procesului de antrenare

Caracteristică	Explicații
Model	ResNet-18, preantrenat pe ImageNet
Optimizator	SGD, Learning Rate = 0.001, Momentum = 0.9
Funcția de eroare	Cross-Entropy Loss
Număr de epoci	10
Fold-uri	5
Dimensiunea batch-ului	32

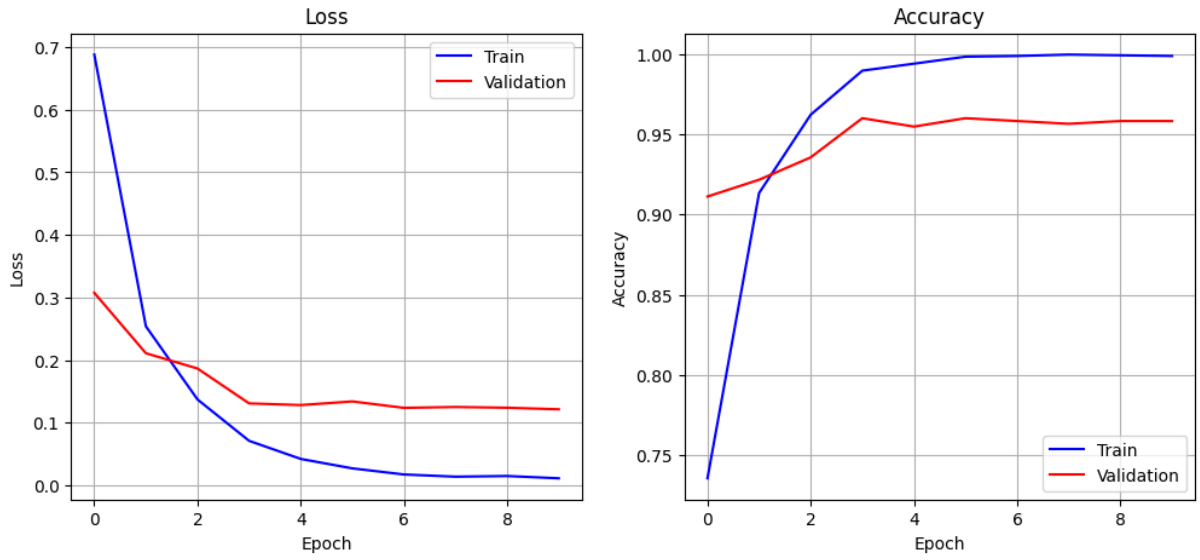

Figure 1: Curbele de antrenare pentru fold-ul 1

```
Precision: 0.8179, Recall: 0.7230, F1-Score: 0.6928, Accuracy: 0.7360
```

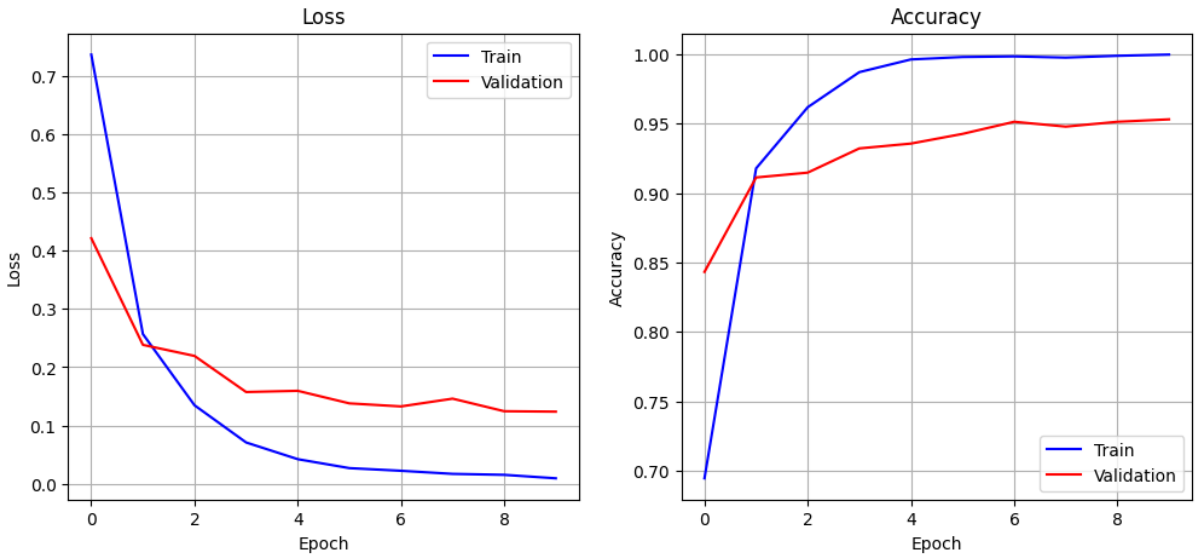

Figure 2: Curbele de antrenare pentru fold-ul 2

```
Precision: 0.8421, Recall: 0.7439, F1-Score: 0.7190, Accuracy: 0.7538
```

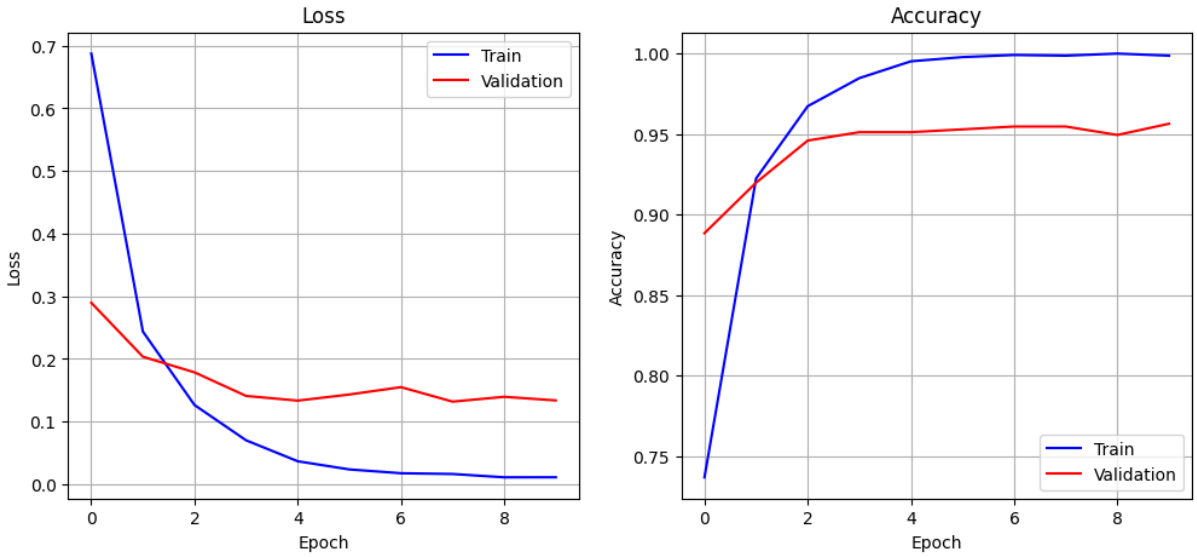

Figure 3: Curbele de antrenare pentru fold-ul 3

```
Precision: 0.8398, Recall: 0.7552, F1-Score: 0.7355, Accuracy: 0.7640
```

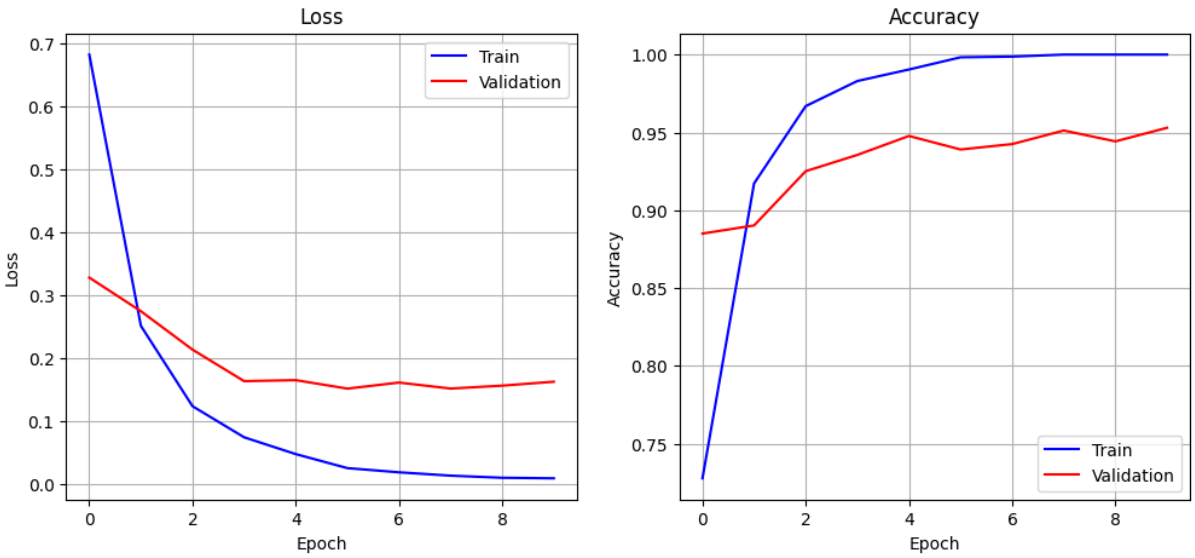

Figure 4: Curbele de antrenare pentru fold-ul 4

```
Precision: 0.8384, Recall: 0.7341, F1-Score: 0.7118, Accuracy: 0.7437
```

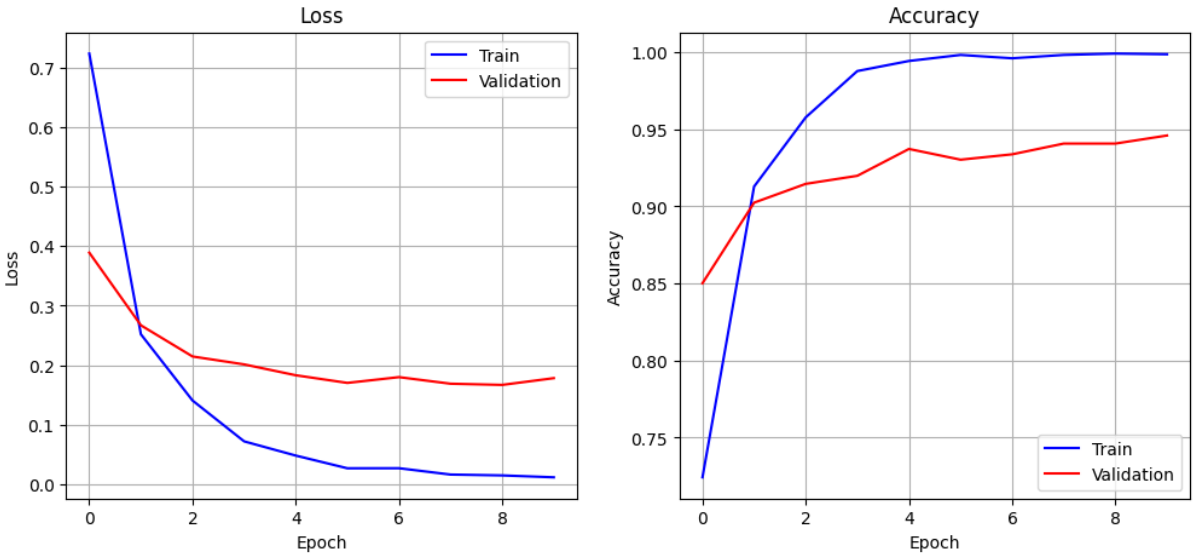

Figure 5: Curbele de antrenare pentru fold-ul 5

```
Precision: 0.8336, Recall: 0.7550, F1-Score: 0.7325, Accuracy: 0.7614
```

Pentru fiecare fold, pe setul de testare, am calculat metricile precision, recall, scor F1 și acuratețe, apoi media și deviația standard a acestora, valori pe care le-am documentat în tabelul:

Table 2: Metrici şi statistici pentru fold-uri

Item	Precision	Recall	F1-Score	Accuracy
Fold 1	0.8179	0.7230	0.6928	0.7360
Fold 2	0.8421	0.7439	0.7190	0.7538
Fold 3	0.8398	0.7552	0.7355	0.7640
Fold 4	0.8384	0.7341	0.7118	0.7437
Fold 5	0.8336	0.7550	0.7325	0.7614
Mean	0.8344	0.7422	0.7183	0.7518
Standard Deviation	0.0087	0.0124	0.0155	0.0106

Observând atât metricile obținute, cât și curbele de eroare și acuratețe, putem conculziona că această abordare generalizează datele satisfăcător, însă în același timp, mai este loc de îmbunătățiri. De asemenea, se poate observa că modelul face puțin overfitting pe toate fold-urile, deoarece pe setul de antrenare eroarea scade și acuratețea crește, dar pe setul de validate acestea stagnează sau au variațiuni foarte mici. Cu toate acestea, procesul de antrenare este stabil, nu există fluctuații, creșteri sau scăderi abrupte.

2 Cerința 2

Hiperparametrii folosiți în continuare sunt aceiași ca la cerința anterioară (doar numărul epocilor este schimbat la 20).

2.1 Funcție de pierdere cu ponderi

Pentru această cerință, am calculat ponderile asociate fiecărei clase, pe care le-am dat funcției de pierdere. Comparația poate fi facută cu unul din modelele de la cerința anterioară, care nu avea ponderi asociate claselor.

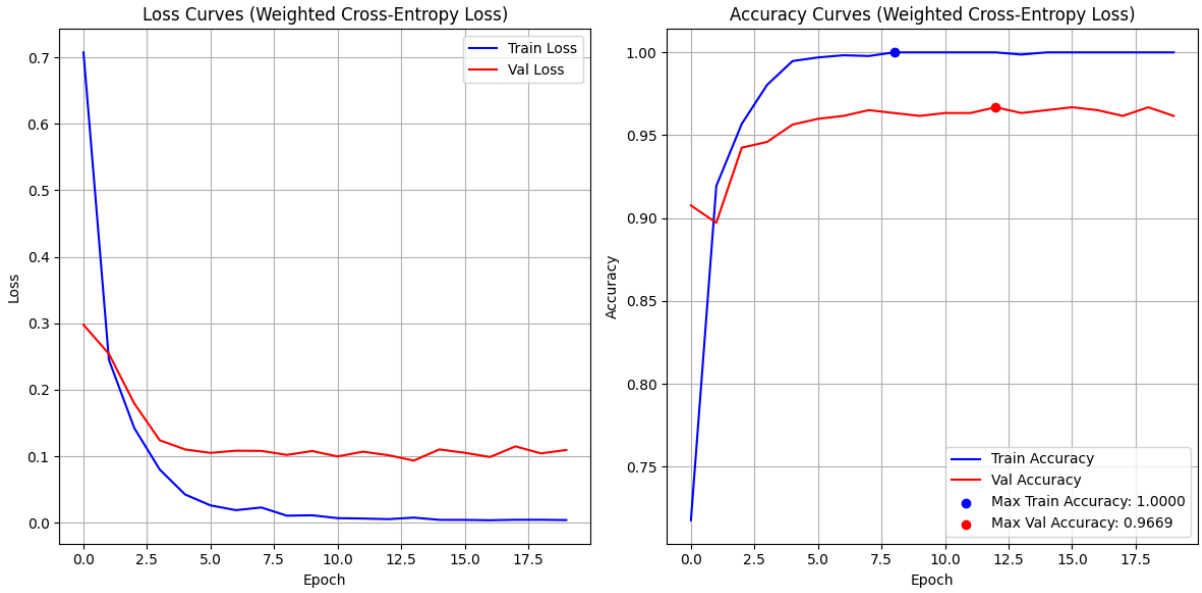

Figure 6: Curbele de antrenare pentru modelul cu funcție de pierdere ponderată

```
Precision: 0.8468, Recall: 0.7682, F1-Score: 0.7515, Accuracy: 0.7766
```

Comparativ cu cel mai bun model din cerința anterioară, acesta obține metrici puțin mai bune la toate categoriile, iar loss-ul este mai mic si modelul este mai stabil în ansamblu.

2.2 Oversampling

Augmentările aplicate imaginilor includ: RandomHorizontalFlip, RandomResizedCrop și RandomRotation cu maxim 15 grade.

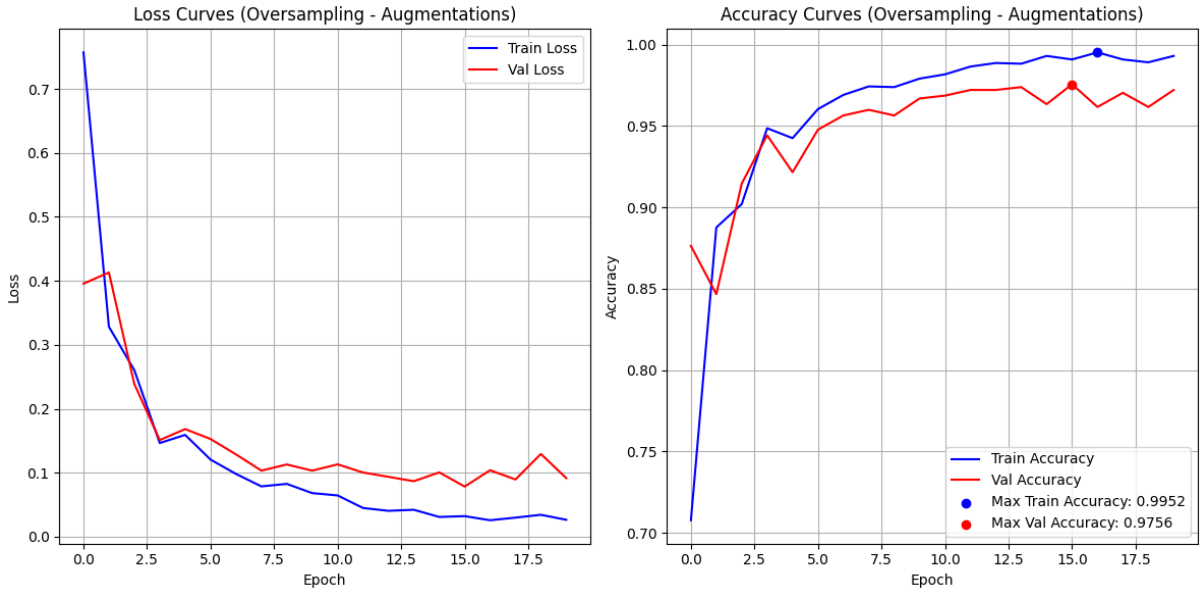

Figure 7: Curbele de antrenare pentru modelul cu imagini augmentate

```
Precision: 0.8548, Recall: 0.7811, F1-Score: 0.7620, Accuracy: 0.7892
```

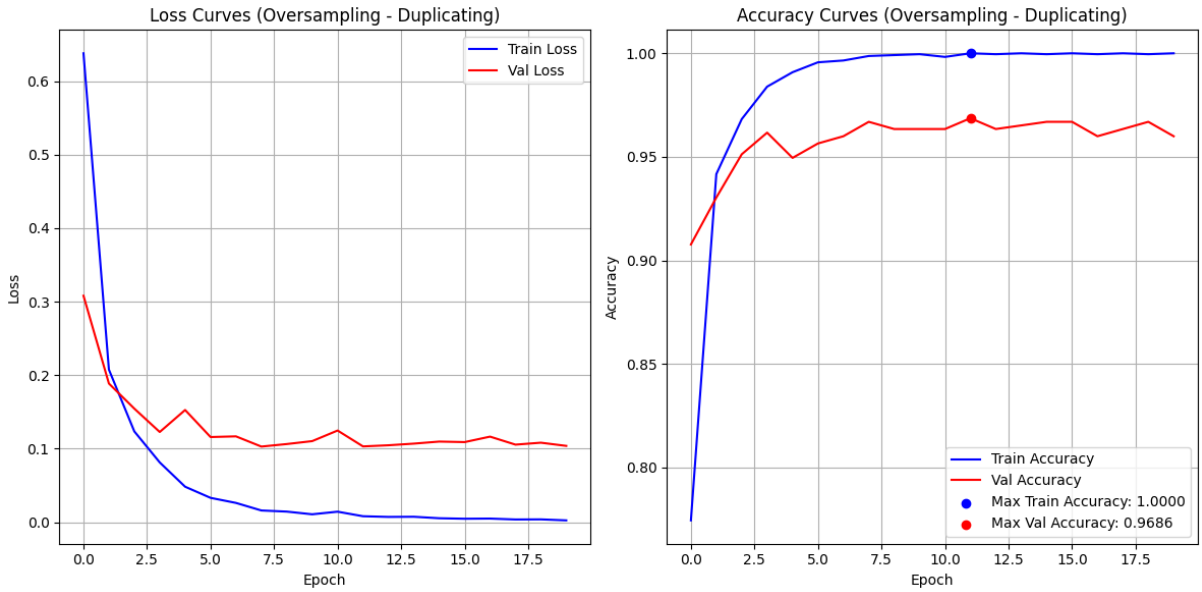

Figure 8: Curbele de antrenare pentru modelul cu imagini duplicate

```
Precision: 0.8384, Recall: 0.7547, F1-Score: 0.7242, Accuracy: 0.7614
```

Varianta cu imaginile augmentate reușește să ajungă la o acuratețe maximă de 0.78 pe setul de testare, mai mult decât celelalte modele. Pe de altă parte, metoda cu duplicarea exemplelor din clasele minoritare pare că duce la o scădere în performanță, alături de overfitting.

3 Cerința 3

Pentru această cerință, am ales 4 seturi de transformări pe care să le aplic celor 5 folduri. Am ales să separ aceste 4 seturi în funcție de cum sunt manipulate imaginile în urma aplicării lor:

Table 3: Setul 1 - Transformări de bază

Transformare	Paremetrii
RandomVerticalFlip	Probabilitate 0.5
RandomHorizontalFlip	Probabilitate 0.5
RandomRotation	Maxim 15 grade

Table 4: Setul 2 - Transformări de intensitate

Transformare	Paremetrii
ColorJitter	Brightness 0.5, Contrast 0.5, Stauration 0.5, Hue 0.5
Normalizare	Media și deviația standard din ImageNet

Table 5: Setul 3 - Transformări de distorsionare

Transformare	Paremetrii
RandomPerspective	Distortion Scale 0.5, Probabilitate 0.5
RandomAffine	Probabilitate 0.5
RandomResizedCrop	Scale 0.8, 1.0

Table 6: Setul 4 - Blur

Transformare	Paremetrii
Blur Gaussian	Kernel 5x5

3.1 Curbele de antrenare și metrici

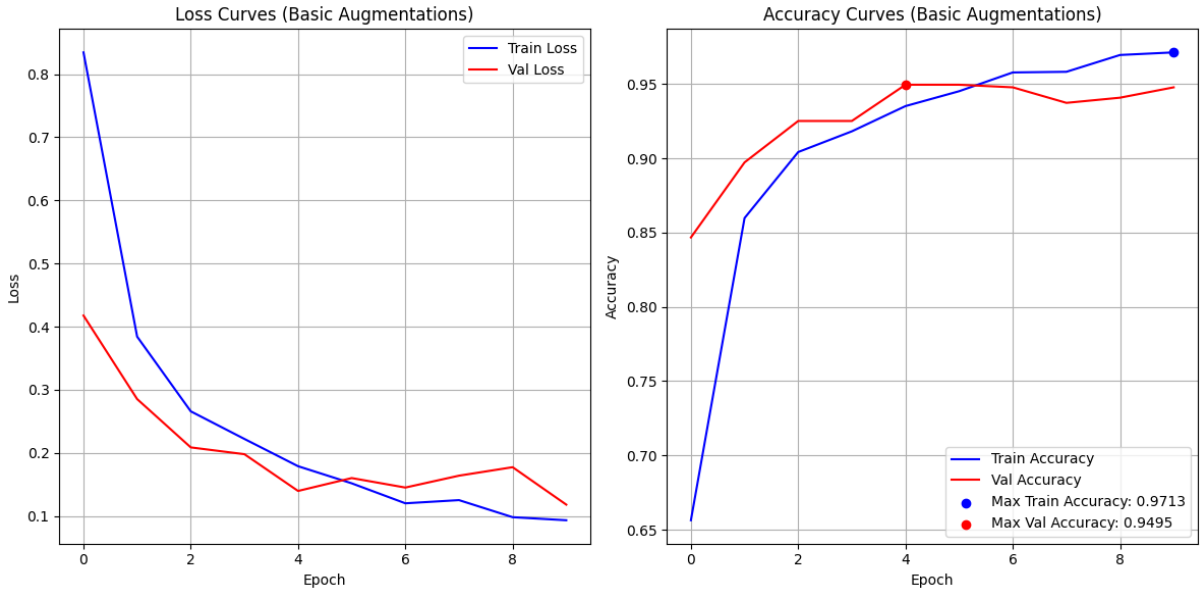

Figure 9: Curbele de antrenare pentru fold-ul 1 cu setul 1 de transformări

```
Precision: 0.8479, Recall: 0.7681, F1-Score: 0.7433, Accuracy: 0.7686
AUC Scores per class: 0.7888 (G), 0.9501 (M), 0.9623 (N), 0.9375 (P)
Macro-Average AUC: 0.9097
```

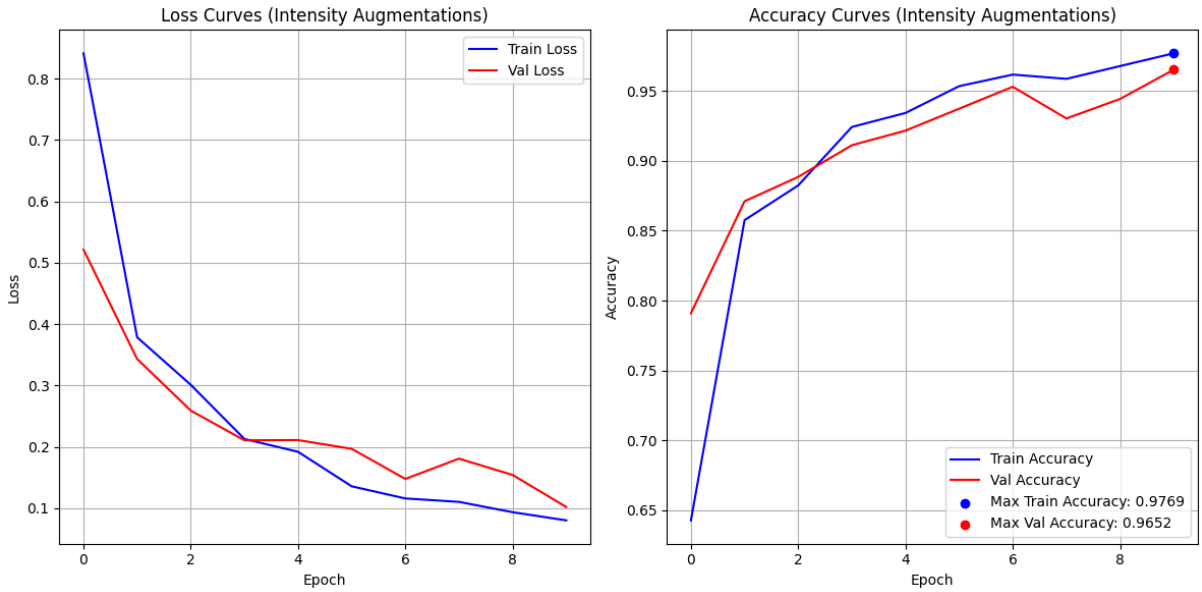

Figure 10: Curbele de antrenare pentru fold-ul 2 cu setul 2 de transformări

```
Precision: 0.8020, Recall: 0.6636, F1-Score: 0.6455, Accuracy: 0.6853
AUC Scores per class: 0.7928 (G), 0.9499 (M), 0.9696 (N), 0.9790 (P)
Macro-Average AUC: 0.9228
```

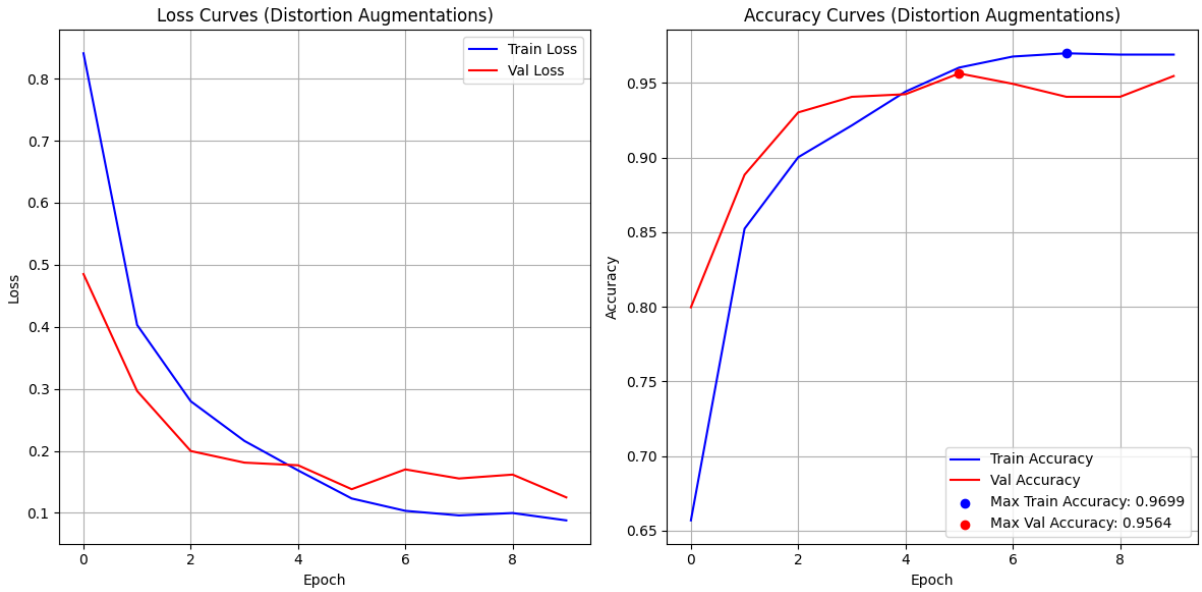

Figure 11: Curbele de antrenare pentru fold-ul 3 cu setul 3 de transformări

```
Precision: 0.8178, Recall: 0.6946, F1-Score: 0.6793, Accuracy: 0.7157
AUC Scores per class: 0.7472 (G), 0.9469 (M), 0.9676 (N), 0.9673 (P)
Macro-Average AUC: 0.9073
```

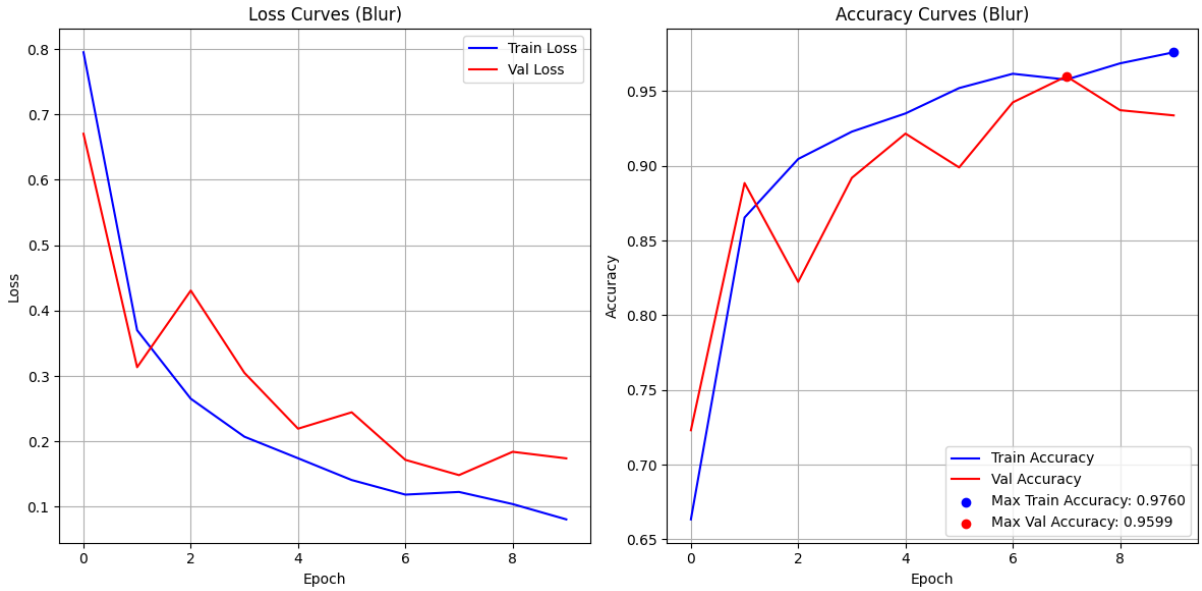

Figure 12: Curbele de antrenare pentru fold-ul 4 cu setul 4 de transformări

```
Precision: 0.7889, Recall: 0.6732, F1-Score: 0.6472, Accuracy: 0.6929
AUC Scores per class: 0.8039 (G), 0.9521 (M), 0.9533 (N), 0.8881 (P)
Macro-Average AUC: 0.8994
```

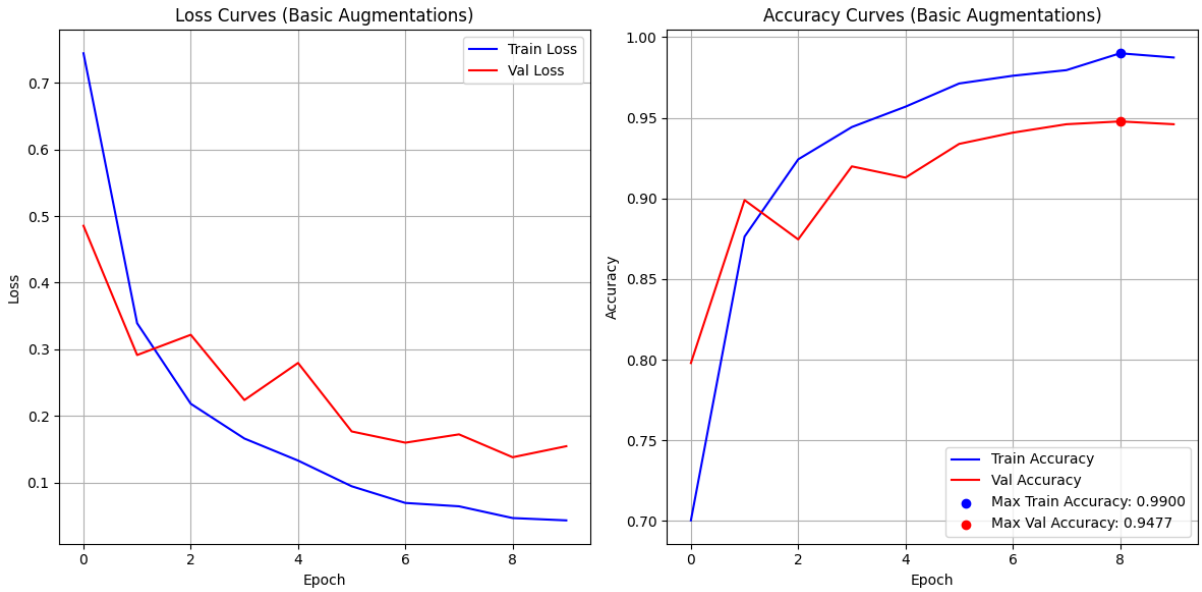

Figure 13: Curbele de antrenare pentru fold-ul 5 cu setul 1 de transformări

```
Precision: 0.8257, Recall: 0.7622, F1-Score: 0.7326, Accuracy: 0.7640
AUC Scores per class: 0.8576 (G), 0.9518 (M), 0.9788 (N), 0.9899 (P)
Macro-Average AUC: 0.9445
```

Table 7: Metrici şi statistici pentru fold-uri cu și fără seturi de augmentări

Item	Precision	Recall	F1-Score	Accuracy	Average AUC
Fold 1 (fără augmentări)	0.8179	0.7230	0.6928	0.7360	-
Fold 2 (fără augmentări)	0.8421	0.7439	0.7190	0.7538	-
Fold 3 (fără augmentări)	0.8398	0.7552	0.7355	0.7640	-
Fold 4 (fără augmentări)	0.8384	0.7341	0.7118	0.7437	-
Fold 5 (fără augmentări)	0.8336	0.7550	0.7325	0.7614	-
Fold 1 (augmentări setul 1)	0.8479	0.7681	0.7433	0.7686	0.9097
Fold 2 (augmentări setul 2)	0.8020	0.6636	0.6455	0.6853	0.9228
Fold 3 (augmentări setul 3)	0.8178	0.6946	0.6793	0.7157	0.9073
Fold 4 (augmentări setul 4)	0.7889	0.6732	0.6472	0.6929	0.8994
Fold 5 (augmentări setul 1)	0.8257	0.7622	0.7326	0.7640	0.9445

Observând datele obținute, putem spune că transformările care modifică intensitatea pixelilor nu ajută capabilitatea modelului să generalizeze, ci scade din performanță, cum este în cazul fold-ului 2 unde se vede o scădere mare în acuratețe. Acest lucru probabil se întâmplă pentru că transformările distorsionează mult prea mult imaginea, iar astfel modelul nu reușește să învețe detaliile imaginilor. De asemenea, distorsionările din setul 3 cum sunt cele de perspectivă înrăutățesc performanța modelului, probabil pentru că în setul de testare nu se regăsesc imagini care au variații mari de perspectivă.

Pe de altă parte, pentru setul 1 de augmentări, care conține rotații și răsturnări de imagini se poate observa o ușoară creștere în performanță. Este oarecum normal ca augmentarea de RandomHorizontalFlip să ajute deoarece tumorile (in special cele gliomice) pot apărea în orice parte a creierului, iar astfel modelul învață să nu țină cont de poziționarea din setul de antrenare. De asemenea, rotația este benefică deoarece pacientul nu va ține capul perfect drept în momentul RMN-ului. De menționat este faptul că tumorile gliomice sunt cele mai slab clasificate, tot din cauză că pot apărea în orice locație în creier.

4 Cerința 4

Pentru secțiunea aceasta, am ales fold-ul cel mai slab, fold-ul 1, care a obținut 0.73 acuratețe.

4.1 Early Stopping

Pentru parametrii early stopping, am ales patience = 3 (numărul maxim de epoci consecutive pentru care să continue antrenarea, chiar dacă pierderea pe setul de validare nu a scăzut) și gamma = 0.1 (diferența minima dintre pierderi, între epoci - minim 10% în cazul de față). Rezultatele sunt următoarele:

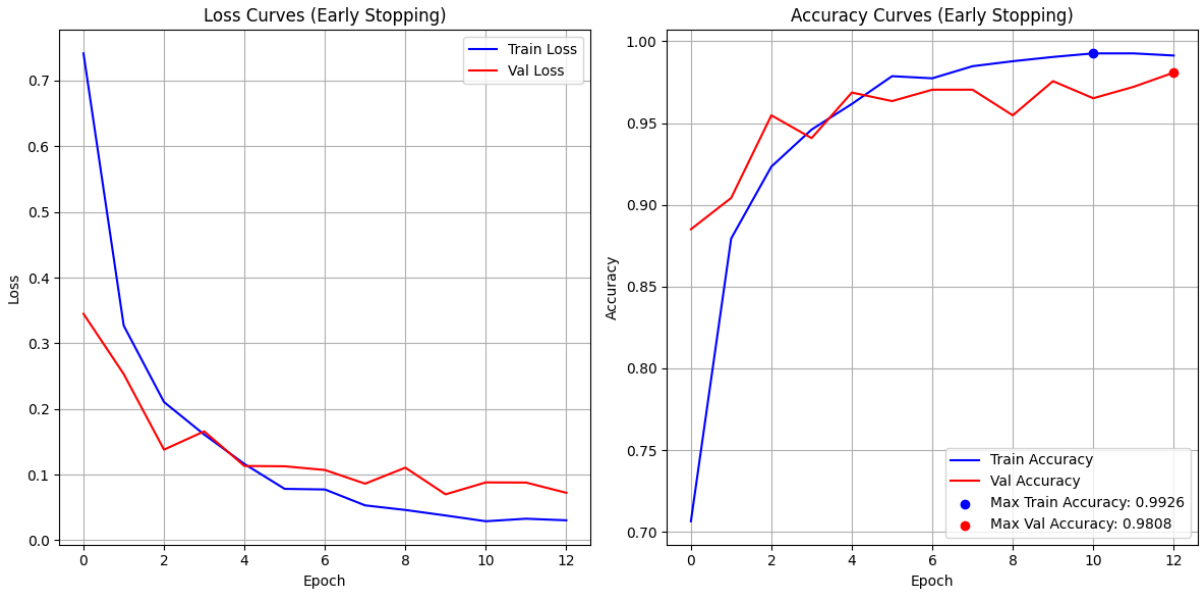

Figure 14: Curbele de antrenare pentru modelul cu early stopping

```
Precision: 0.8278, Recall: 0.7448, F1-Score: 0.7072, Accuracy: 0.7538
```

Se poate observa că antrenarea cu early stopping ajută la îmbunătățirea preformanței pe fold-ul acesta, având o creștere mică a acurateții de la 0.73 la 0.75, precum și la celelalte metrici. De asemenea, nu sunt prezente semne de overfitting.

4.2 Learning Rate Scheduling - StepLR

Pe post de scheduler am ales varianta StepLR care reduce rata de învățare cu un anumit factor la un interval fix de epoci. Parametrii aleși sunt: gamma = 0.1 (factorul cu care se reduce rata de învățare inițială) și step size = 5 (după câte epoci se reduce rata de învățare cu factorul gamma).

Această abordare a dus de asemenea la o creștere similară în performanță, obținând rezultate apropiate (aceeasi acuratețe) cu varianta early stopping. La fel ca la metoda cu early stopping, modelul nu face overfitting (deși sunt creșteri și scăderi bruște la început, antrenarea se stabilizează spre final):

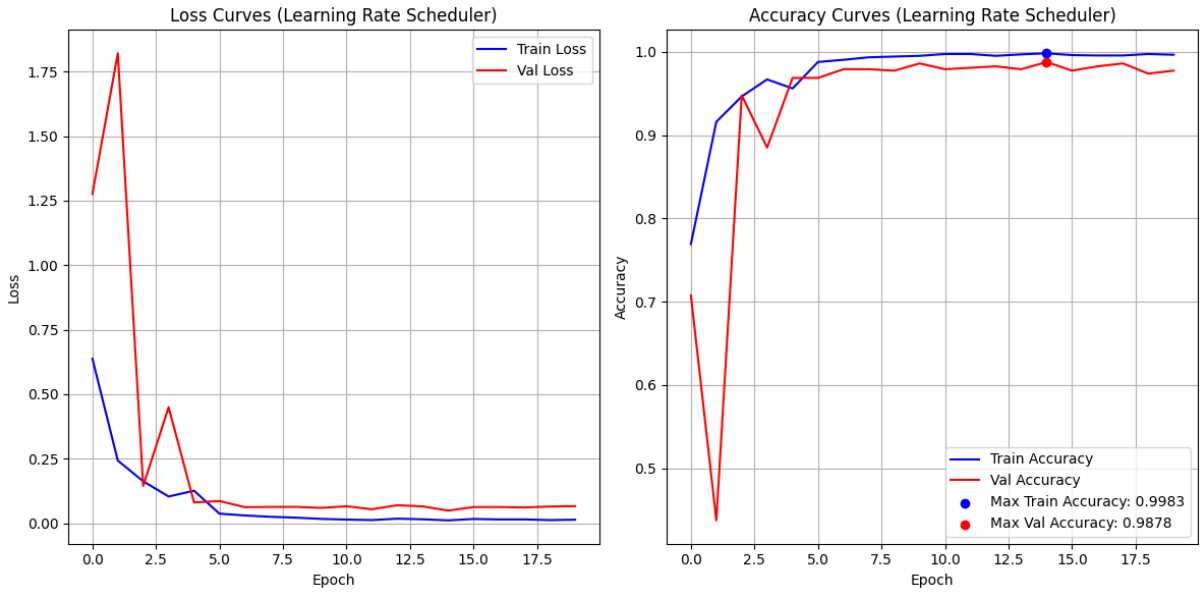

Figure 15: Curbele de antrenare pentru modelul cu scheduler

```
Precision: 0.8418, Recall: 0.7409, F1-Score: 0.7145, Accuracy: 0.7538
```

Table 8: Comparație fold 1 cu early stopping și scheduler

Item	Precision	Recall	F1-Score	Accuracy	Time
Fold 1	0.8179	0.7230	0.6928	0.7360	6m 20s
Fold 1 (early stopping)	0.8278	0.7448	0.7072	0.7538	4m 18s
Fold 1 (scheduler)	0.8418	0.7409	0.7145	0.7538	6m 29s

În concluzie, utilizarea tehnicilor precum early stopping sau folosirea unui scheduler aduc o mică îmbunătățire performanței modelului, dar nimic semnificativ. O posibilă justificare pentru folosirea early stopping-ului, de exemplu, ar putea fi economisirea resurselor computaționale și a timpului, ținând cont că se obține aceeași acuratețe, cum este cazul în care modelul se antrenează cu numărul total de epoci, dar cu scheduler.

5 Cerința 5

5.1 Modificarea funcției de pierdere

În loc să folosesc Binary Cross-Entropy Loss drept de funcție de pierdere, am ales să implementez și să experimentez cu Focal Loss. Am antrenat rețeaua cu diferite valori pentru parametrii alpha și gamma:

Table 9: Comparație BCE și Focal Loss pe Fold 1

Loss Function	Parameters	Precision	Recall	F1-Score	Accuracy	AUC	Accuracy (Val)
BCE	-	0.81	0.72	0.69	0.73	0.90	0.95
Focal Loss	$\alpha = 1, \gamma = 2$	0.81	0.68	0.65	0.69	0.91	0.91
Focal Loss	$\alpha = 1, \gamma = 3$	0.83	0.75	0.72	0.75	0.92	0.96
Focal Loss	$\alpha = 1, \gamma = 4$	0.83	0.71	0.69	0.73	0.92	0.96
Focal Loss	$\alpha = 0.25, \gamma = 2$	0.81	0.69	0.66	0.71	0.89	0.96

5.2 Modificarea optimizatorului

În general, când se face fine-tuning pe o rețea preantrenată, cum este cazul de față, se folosește optimizatorul SGD cu momentum. Așadar, am testat antrenarea cu optimizatorul Adam și RMSprop:

Table 10: Comparație optimizatori (folosind Cross-Entropy Loss)

Optimizer	Parameters	Precision	Recall	F1-Score	Accuracy	AUC	Accuracy (Val)
SGD	Learning Rate = 0.001 Momentum = 0.9	0.81	0.72	0.69	0.73	0.90	0.95
Adam	Learning Rate = 0.001	0.83	0.74	0.71	0.75	0.94	0.97
RMSprop	Learning Rate = 0.001	0.80	0.77	0.73	0.76	0.89	0.93

Table 11: Comparație optimizatori (folosind Focal Loss cu cei mai buni parametrii gasiți anterior)

Optimizer	Parameters	Precision	Recall	F1-Score	Accuracy	AUC	Accuracy (Val)
SGD	Learning Rate = 0.001 Momentum = 0.9	0.83	0.75	0.72	0.75	0.92	0.96
Adam	Learning Rate = 0.001	0.83	0.77	0.73	0.77	0.92	0.95
RMSprop	Learning Rate = 0.001	0.76	0.71	0.68	0.73	0.90	0.92

De menționat este că antrenarea cu optimizatorul SGD este mult mai stabilă față de Adam sau RMSprop, unde sunt fluctuații mari ale pierderii și acurateții între epoci. De asemenea, SGD converge în mai puține epoci la aproximativ același rezultat obținut de Adam sau RMSprop.

5.3 Modificarea dimensiunii batch-ului

Am folosit criteriul Cross-Entropy Loss, optimizatorul SGD cu learning rate = 0.001, momentum = 0.9 și am testat modelul cu diferite dimensiuni de batch-uri:

Table 12: Comparație între anternarea cu dimensiuni de batch-uri diferite

Batch Size	Time	Precision	Recall	F1-Score	Accuracy	AUC	Accuracy (Val)
16	7m 16s	0.85	0.76	0.74	0.77	0.95	0.98
32	6m 55s	0.81	0.72	0.69	0.73	0.90	0.95
64	6m 38s	0.83	0.75	0.73	0.76	0.93	0.97
128	6m 20s	0.82	0.74	0.71	0.75	0.91	0.95

5.4 Tehnici de regularizare

Am testat diferite metode de regularizare pentru a preveni overfitting-ul. La fel ca la cerința anterioară, am folosit BCE Loss și optimizatorul SGD:

Table 13: Comparație metode de regularizare

Item	Parameters	Precision	Recall	F1-Score	Accuracy	AUC	Accuracy (Val)
L2 Regularization	Decay = 1e-2	0.81	0.71	0.68	0.72	0.91	0.95
L2 Regularization	Decay = 1e-4	0.82	0.70	0.67	0.71	0.92	0.97
Dropout Layer	Probability 0.3	0.78	0.67	0.64	0.69	0.91	0.95
Early Stopping	patience = 3 gamma = 0.1	0.82	0.74	0.70	0.75	0.90	0.98

5.5 Interpretarea rezultatelor

5.5.1 Comparația între versiuni ale modelului

În cazul augmentărilor, modelul care avea aplicate transformări de bază cum ar fi rotația și răsturnarea imaginilor a avut o creștere în performanță față de modelul fără imagini augmentate (de la 0.73 acuratețe la 0.77 pe fold-ul 1). O creștere asemănătoare se poate observa de altfel pe toate fold-urile. Pe de altă parte, augmentările care modifică perspecitva imaginilor sau alterează intensitatea pixelilor duc la o scădere în performanța modelului. Aceste augmentări ajută modelul să învețe diferite poziționări ale tumorilor, dar și pozițiile diferite în care un pacient își poate ține capul în timpul scanării.

Folosirea algoritmului de early stopping pentru a opri antrenarea dacă acuratețea pe setul de validare nu se îmbunătățește a adus un mic plus de performanță pe setul de testare. Varianta cu calendarul în trepte pentru rata de învățare nu a modificat deloc performanța modelului. O explicație posibilă este că oprirea mai devreme a antrenării nu mai permite modelului să facă overfitting pe setul de antrenare, iar astfel se obțin rezultate mai bune pe setul de testare.

5.5.2 Performanța pe clasele minoritare

Metoda care a adus cea mai mare creștere în performanță pe fold-ul 1 a fost aplicarea augmentărilor de bază descrise mai sus, alături de oversampling pentru a echilibra distribuția claselor, care a adus modelul de la 0.73 acuratețe la 0.78 pe setul de testare. De asemenea, făcând oversampling doar prin duplicarea imaginilor duce la o creștere, însă mai puțin decât în cazul anterior, ajungând la 0.76 acuratețe pe setul de testare.

Utilizarea funcției de loss cu ponderi ajută, însă nu în aceeași măsură precum varianta cu augmentări și oversampling pentru echilibrare. Cu funcția de loss cu ponderi, modelul ajunge la acuratețea de 0.77.

5.5.3 Impactul modificărilor hiperparametrilor

Dacă utilizăm Focal Loss pe post de funcție de pierdere, în loc de Binary Cross-Entropy Loss, crește puțin performanța pe setul de testare, ajungând la 0.75 acuratețe pentru setul potrivit de parametri ales pentru Focal Loss. Acest fapt se poate explica prin faptul că Focal Loss a fost proiectat să se descurce mai bine în clasificări cu clase dezechilibrate.

În combinație cu de funcția de pierdere, optimizatorul influențează acuratețea modelului. Pentru modelul care folosește Cross-Entropy Loss, Adam si RMSprop obțin rezultate mai bune decât SGD (0.75 și respectiv 0.76 acuratețe pe setul de testare). Pentru modelul cu Focal Loss, Adam performează la fel ca SGD, dar RMSprop este mai slab, scazând acuratețea la 0.68.

În ambele situații, SGD converge mai rapid și este mult mai stabil pentru rețeaua aleasă, față de Adam și RMSprop care au fluctuații mari de loss și acurateță în timpul antrenării.

Miscșorarea ratei de învățare nu modifică deloc performanța modelului, doar timpul de convergență. Creșterea ratei de învățare induce instabilitate în procesul de învățare și nu se obțin rezultate la fel de bune (cel mai probabil în timpul gradient descent-ului se sare peste punctul de minim, având rata de învățare prea mare).

Modificarea dimensiunii la 16 impactează în mod pozitiv acuratețea, ducând-o la 0.77, dimensiuni mai mari de batch-uri doar scad acuratețea modelului.

5.5.4 Impactul regularizării

Toate metodele de regularizare încercate (în afară de early stopping) au dus la scăderea performanței modelului. O explicație ar putea fi că modelul nu făcea overfitting oricum (cum se vede și pe curbele de antrenare), deci tehnicile de regularizare diminuau din capabilitatea modelului de a învăța mai mult decât să îl ajute în vreun fel.

6 Bonus 2

Ținând cont că la restul proiectului am folosit modelul ResNet-18 preantrenat, am decis ca la această cerință să folosesc EfficientNet-B0, pentru a compara performanțele celor două rețele. Hiperparametrii cu care am antrenat rețeaua sunt:

Table 14: Caracteristicile procesului de antrenare

Caracteristică	Explicații
Model	EfficientNet-B0, preantrenat pe ImageNet
Optimizator	SGD, Learning Rate = 0.001, Momentum = 0.9
Funcția de eroare	Cross-Entropy Loss cu echilibrare de clase
Număr de epoci	20
Dimensiunea batch-ului	16
Augmentări	RandomHorizontalFlip, RandomRotate

Table 15: Comparație antrenări

Item	Precision	Recall	F1-Score	Accuracy	Accuracy (Val)
Fine-Tuning	0.84	0.76	0.74	0.77	0.98
Fully-Connected	0.73	0.59	0.56	0.60	0.90

6.1 Fine-tuning pe toată rețeaua

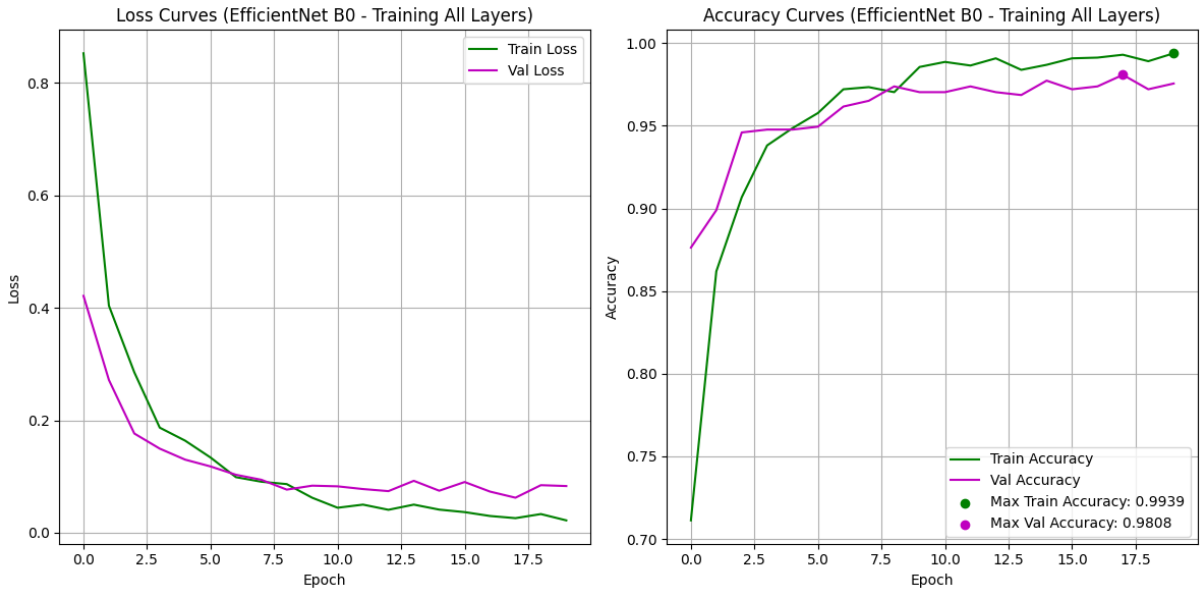

Figure 16: Curbele de antrenare pentru fine-tuning

6.2 Antrenarea stratului Fully-Connected

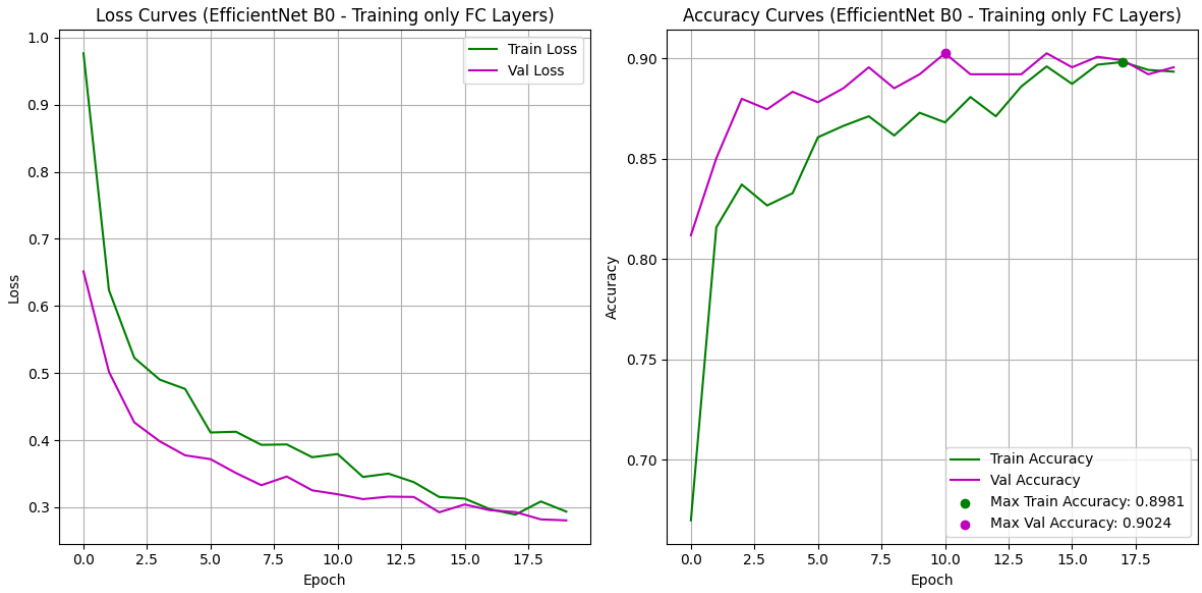

Figure 17: Curbele de antrenare doar pentru stratul fully-connected

Varianta modelului cu toate straturile dezghețate obține rezultate aproape identice cu modelul folosit pe parcursul acestei etape. De aici, o concluzie poate fi că acuratețea de 0.78-0.79 este aproape de maximul care se poate obține pe acest set de date folosind modelele preantrenate de aceste dimensiuni.

Pentru antrenarea doar cu stratul Fully-Connected dezghețat, modelul reușește să aibă rezultate semnificativ mai slabe decât pentru cel pe care s-a făcut fine-tuning, semn că nu este suficientă doar antrenarea stratului fully-connected pentru a obține rezultate satisfăcătoare.