

Raport Tema 2: Învățare Automată

Clasificare de Imagini și Text

Luca Plian
341C4

18 ianuarie 2026

Introducere

Acest raport prezintă analiza și rezultatele obținute în cadrul Temei 2, având ca scop clasificarea imaginilor (Setul Imagebits și Land Patches) și a textului (Ro-Sent). Analiza se concentrează pe compararea arhitecturilor de tip MLP și CNN, respectiv RNN și LSTM pentru text.

1 Partea 1: Clasificare de Imagini (Imagebits)

1.1 Explorarea Datelor

Setul de date *Imagebits* conține imagini color cu rezoluția 96×96 pixeli, împărțite în 10 clase (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). Setul este echilibrat, având 800 de imagini de antrenare și 500 de test per clasă.

1.1.1 Analiza Detaliată a Variabilității Intra-clasă

O inspecție vizuală aprofundată a eșantioanelor din cadrul aceleiași clase a scos la iveală două provocări fundamentale care cresc dificultatea sarcinii de clasificare: inconsistență scării obiectelor și diversitatea cromatică extremă.

1. Problema Imagineilor Incomplete (Padding) Un număr mare de imagini nu sunt "fullscreen". Pentru a fi aduse la rezoluția fixă de 96×96 pixeli fără a fi deformate, multor imagini le-au fost adăugate margini negre (padding) sus-jos sau stânga-dreapta. Această "umplutură" neagră reduce dimensiunea efectivă a obiectului și introduce zgomot, afectând în special clasele.

Clasa 10 (Ship), Clasa 2 (Truck), Clasa 4 (Car) și Clasa 1 (Airplane): Am observat că multe vapoare, dar și câteva mașini, camioane și avioane, sunt obiecte foarte alungite. Când au fost redimensionate pentru a intra în pătratul de 96×96 , au rămas cu benzi negre mari pe margini. Astfel, obiectul real ajunge să ocupe foarte puțini pixeli din imagine, făcând detaliile (cum ar fi roțile sau aripile) extrem de greu de observat pentru rețea, care vede mai mult "negru" decât obiect. Încă o problema e că, dacă mașina sau camionul are o culoare închisă, riscă să se confundă cu marginile negre adăugate, iar modelul nu mai înțelege clar conturul vehiculului.

Practic, rețeaua primește o imagine unde o bună parte din informație este zero (negru), ceea ce o poate deruia la extragerea trăsăturilor.

Confuzii cauzate de Culori Pe lângă problema dimensiunilor, am observat că obiectele din clasele de vehicule au culori extrem de variate, ceea ce poate induce modelul în eroare:

- La **Clasa 4 (Car)** și **Clasa 2 (Truck)**, vehiculele pot fi orice culoare (roșu, alb, albastru etc.).
- Acest lucru e problematic deoarece modelul nu se poate baza pe culoare pentru a le distinge (cum face la "cer albastru" pentru avioane). Trebuie să învețe forme (roți, geamuri), dar astăzi cum am zis mai sus, formele sunt greu de văzut când poza e mică și are margini negre.

Diferențe mari de formă în interiorul aceleiași clase Pe lângă culoare, am observat că animalele sunt mult mai greu de recunoscut decât vehiculele din două motive simple:

- **Rase diferite (Câini și Păsări):** Eticheta "Dog" este foarte generală. Vizual, un câine mic (gen Chihuahua) nu seamănă deloc cu un câine mare (Ciobănesc), deși amândoi sunt în aceeași clasă. Rețeaua trebuie să facă un efort mare să înțeleagă că ambele forme înseamnă "câine", spre deosebire de mașini care au forme standard.
- **Pozitia corpului:** Spre deosebire de o mașină care este un obiect rigid și arată cam la fel indiferent cum o rotești, animalele își schimbă formă corpului. O pisică care doarme ghemotoită arată complet diferit de o pisică ce sare sau aleargă. Asta complică mult treaba modelului de a găsi un contur unic.

Problema Fundalului ("Modelul trișează") O chestiune foarte interesantă pe care am remarcat-o este că anumite obiecte apar aproape mereu pe același tip de fundal. Există riscul ca rețeaua să nu se uite la obiect, ci la ce e în spatele lui:

- **Avioane și Vapoare (Prea mult albastru):** Majoritatea vapoarelor sunt pe apă, iar avioanele pe cer. Ambele fundaluri sunt albastre și uniforme. Modelul ar putea învăța greșit regula simplă: "mult albastru înseamnă avion sau vapor". Dacă îi dăm o poză cu un avion pe pistă (fundal gri) sau o barcă trasă pe uscat, s-ar putea să greșească pentru că îi lipsește indiciul de culoare.
- **Maimuțe (Clasa 9):** Aproape toate pozele cu maimuțe sunt făcute în copaci sau pădure. E foarte posibil ca modelul să învețe să detecteze "frunze verzi și crengi" și să zică automat "Maimuță", fără să se uite atent la trăsăturile feței animalului.

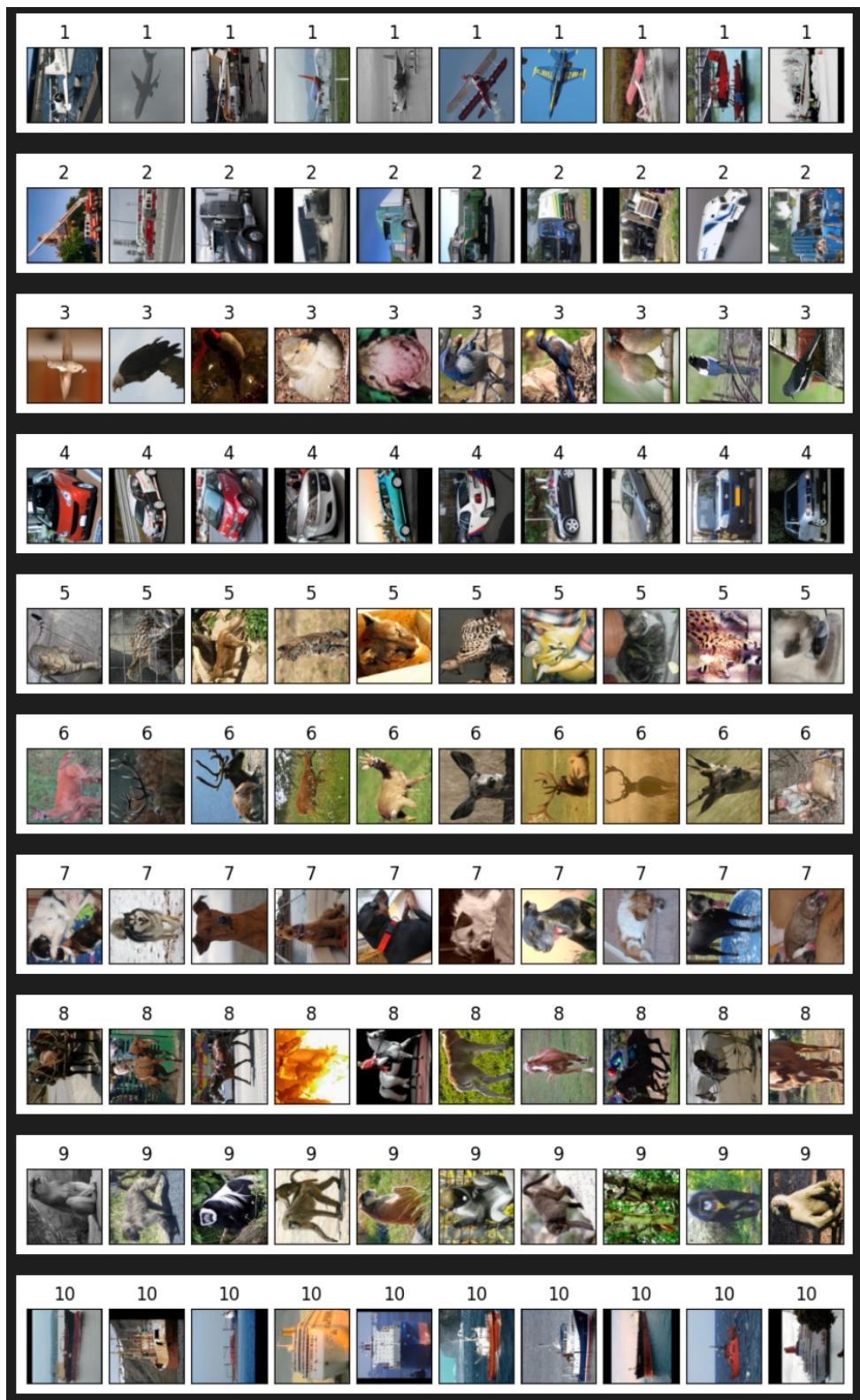


Figura 1: Exemplu intra-clasa pentru train

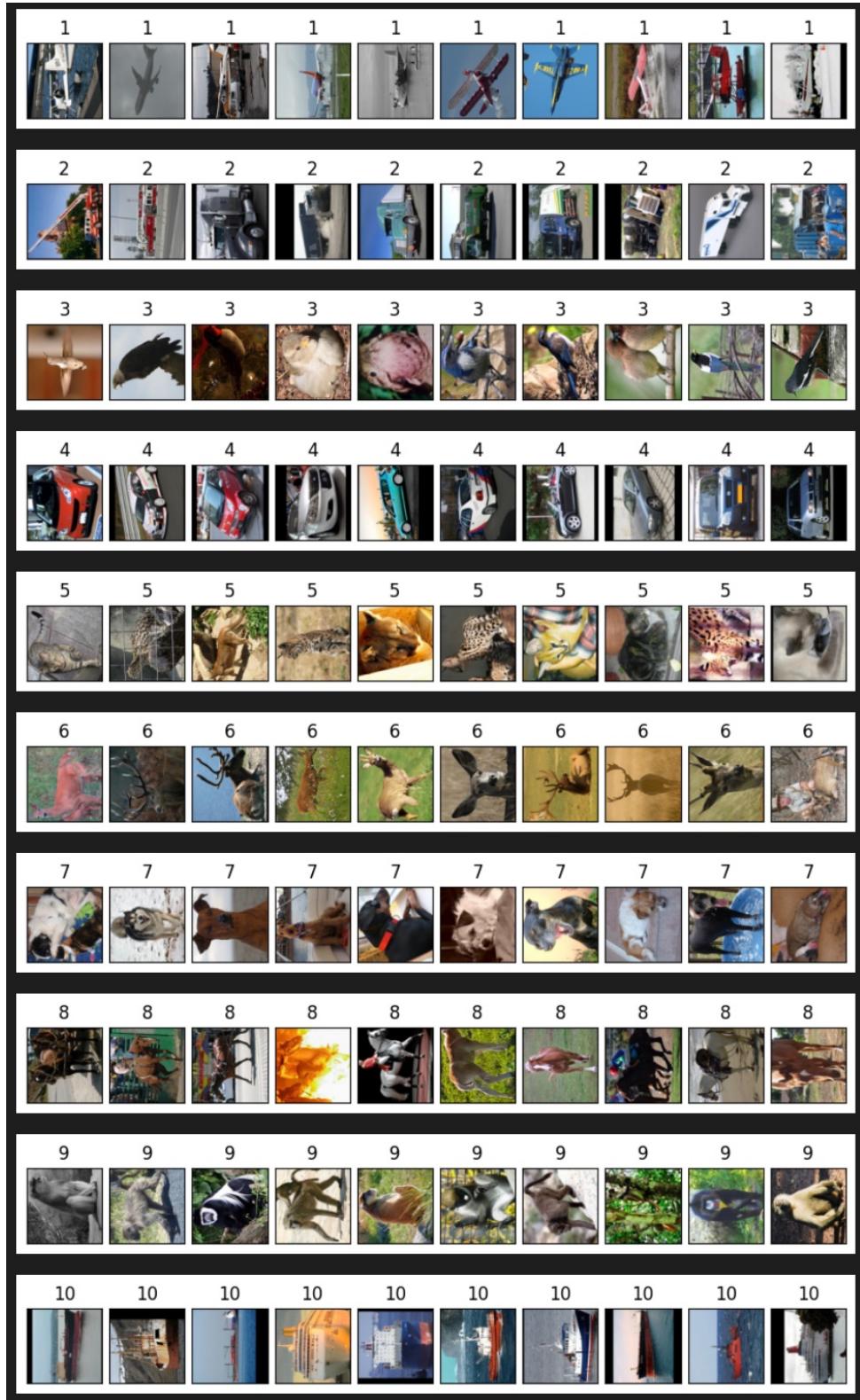


Figura 2: Exemplu intra-clasa pentru test

Obiecte parțial ascunse Mai ales la animalele sălbatice (Căprioare, Pisici), am văzut multe poze în care animalul este ascuns parțial după iarbă înaltă sau tufișuri. Pentru că nu se vede tot conturul animalului, ci doar bucăți din el (capul sau spatele), rețelei îi este greu să își dea seama despre ce formă este vorba, mai ales la rezoluția mică a imaginilor.

Concluzie Intra-clasă Din toate aceste observații — imaginile cu margini negre, culorile care variază mult la mașini, formele diferite ale animalelor și fundalurile care pot păcăli rețeaua — reiese că setul *Imagebits* prezintă destule dificultăți. Este clar că un model care doar memorează pixelii va face rapid overfitting. De aceea, este necesar să folosim augmentarea datelor, pentru a obliga rețeaua să învețe formele reale ale obiectelor, ignorând detaliile irelevante din fundal.

1.1.2 Verificarea Echilibrului Claselor

O etapă preliminară a constat în analiza distribuției cantitative a eșantioanelor per clasă, pentru a preveni riscul ca modelul să devină părtinit (biased) către clasele majoritare.

Conform distribuțiilor ilustrate în Figurile 5 și 6, setul de date este perfect echilibrat:

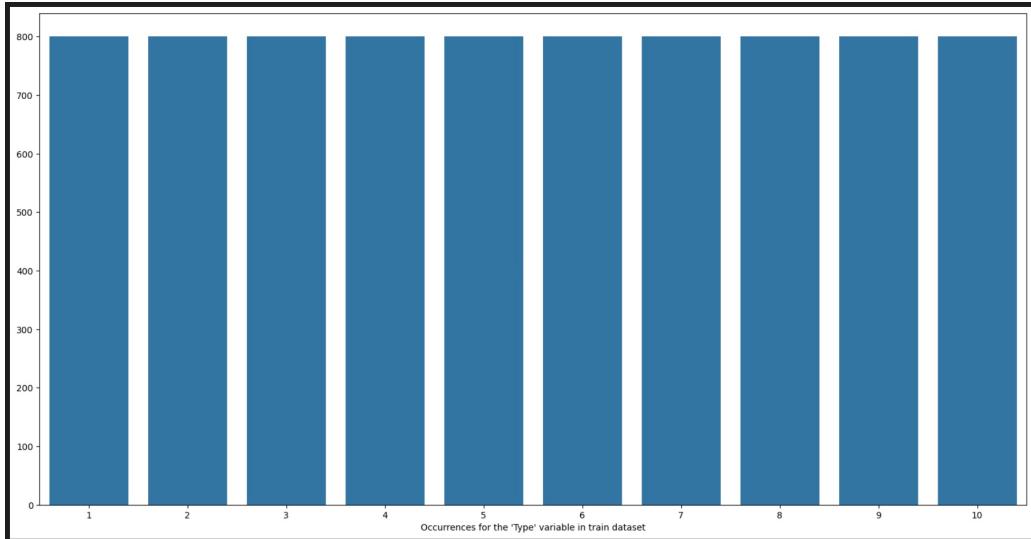


Figura 3: Distribuția uniformă a imaginilor în setul de antrenare (800 per clasă).

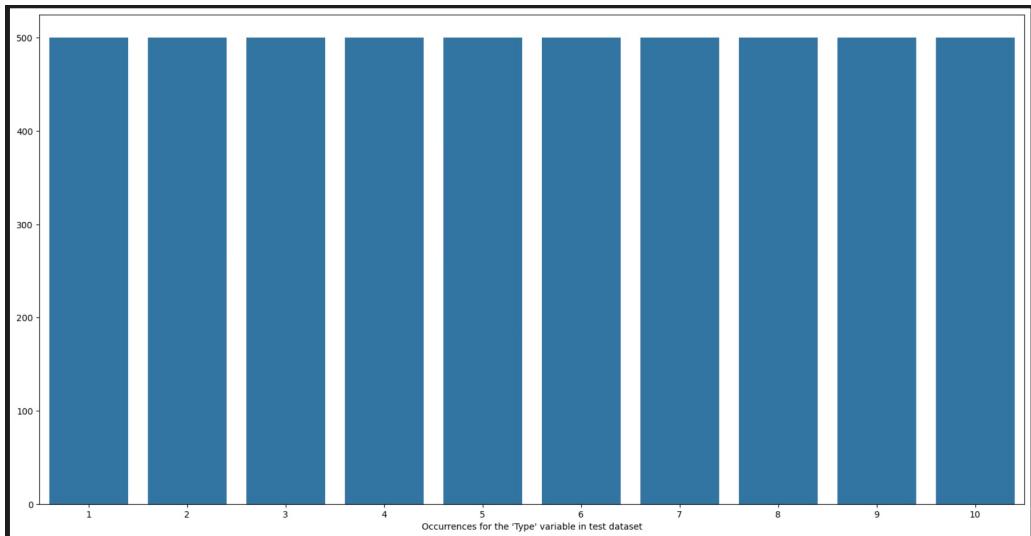


Figura 4: Distribuția uniformă a imaginilor în setul de testare (500 per clasă).

Se observă faptul că fiecare dintre cele 10 clase conține exact 800 de imagini pentru antrenare și 500 pentru testare. Această uniformitate elimină necesitatea utilizării unor

tehnici speciale de balansare (precum *Weighted Loss*) și validează Acuratețea ca metrică relevantă pentru evaluarea performanței modelului.

1.1.3 Analiza Inter-clasă (Similarități și Confuzii)

Pe lângă dificultățile interne ale claselor, analiza vizuală a scos în evidență două grupuri majore de obiecte între care apar confuzii frecvente, cauzate de trăsături comune pe care modelul le interpretează eronat.

Confuzii în Grupul Animalelor (Clasele 5, 6, 7, 8, 9) S-a observat o rată ridicată de suprapunere între clasele biologice: Pisici (5), Câini (6), Căprioare (7), Cai (8) și Maimuțe (9). Factorul principal care generează aceste confuzii este similaritatea cromatică. În majoritatea imaginilor din aceste categorii, predomină nuanțele de maro, negru și bej (culoarea blănii). Întrucât modelul primește imagini la rezoluție mică, acesta tinde să se bazeze pe distribuția culorilor, nereușind să distingă eficient între specii diferite care împart aceeași paletă de culori.

Confuzii în Grupul Vehiculelor (Clasele 2 și 4) O a doua sursă semnificativă de erori a fost identificată între Camioane (Clasa 2) și Mașini (Clasa 4). Aceste confuzii se datorează faptului că ambele clase posedă trăsături structurale identice: prezența roțiilor, forma caroseriei și aspectul metalic. Deoarece geometria de bază este foarte asemănătoare, modelul întâmpină dificultăți în a diferenția un camion de o mașină doar pe baza formei ("shape"), mai ales în imaginile în care scara obiectului nu este evidentă.

Prezența Elementelor Perturbatoare (Co-ocurența cu Oameni) O problemă specifică claselor domestice (*Horse*, *Dog*) este prezența frecventă a oamenilor în cadru.

- La clasa **Horse (8)**, multe imagini surprind calul împreună cu un călăret.
- Există riscul ca rețeaua să asocieze prezența umană cu eticheta "Cal", ceea ce ar duce la erori de clasificare dacă în imagine ar apărea un om fără cal.

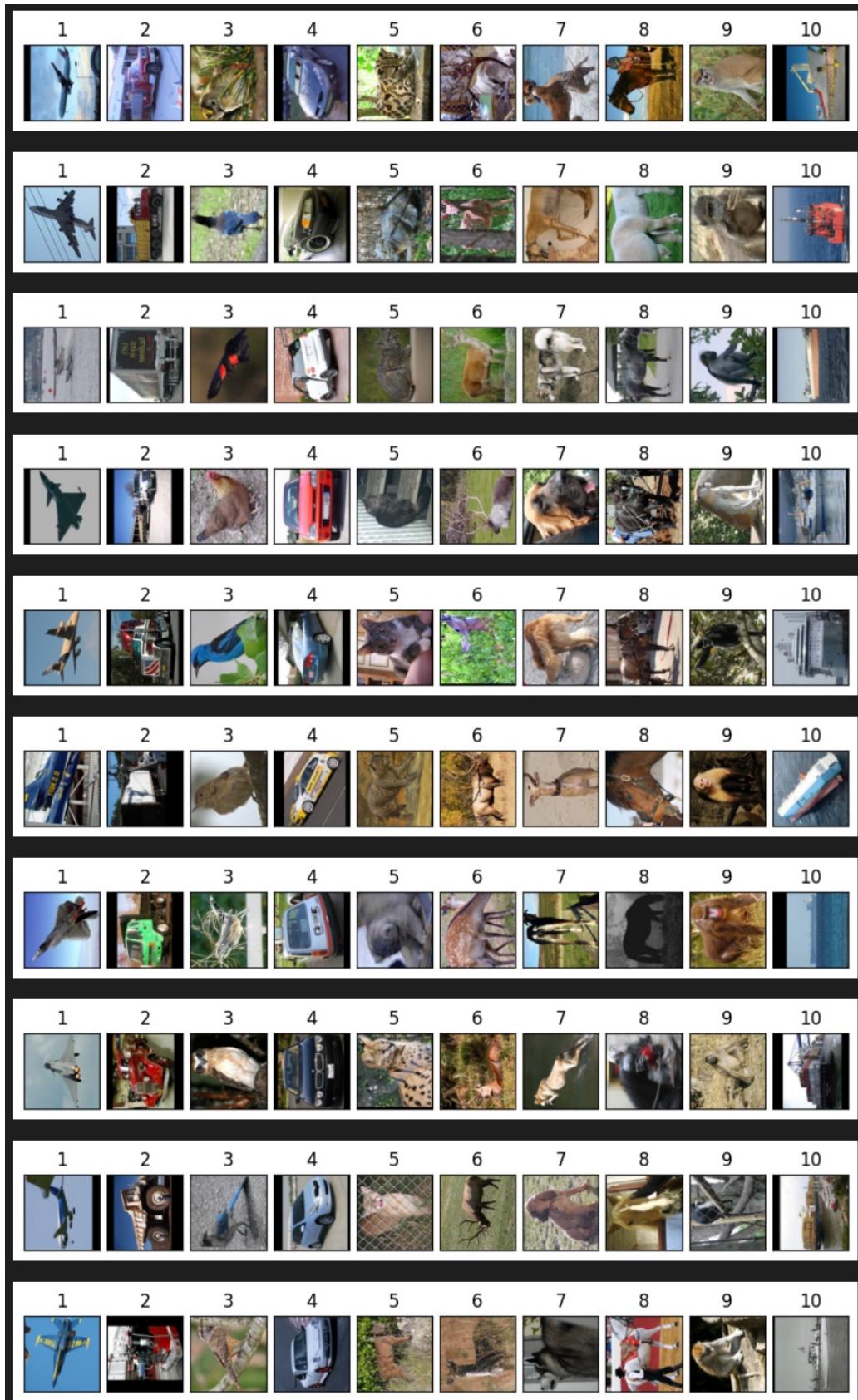


Figura 5: Exemplu inter-clasa pentru train.

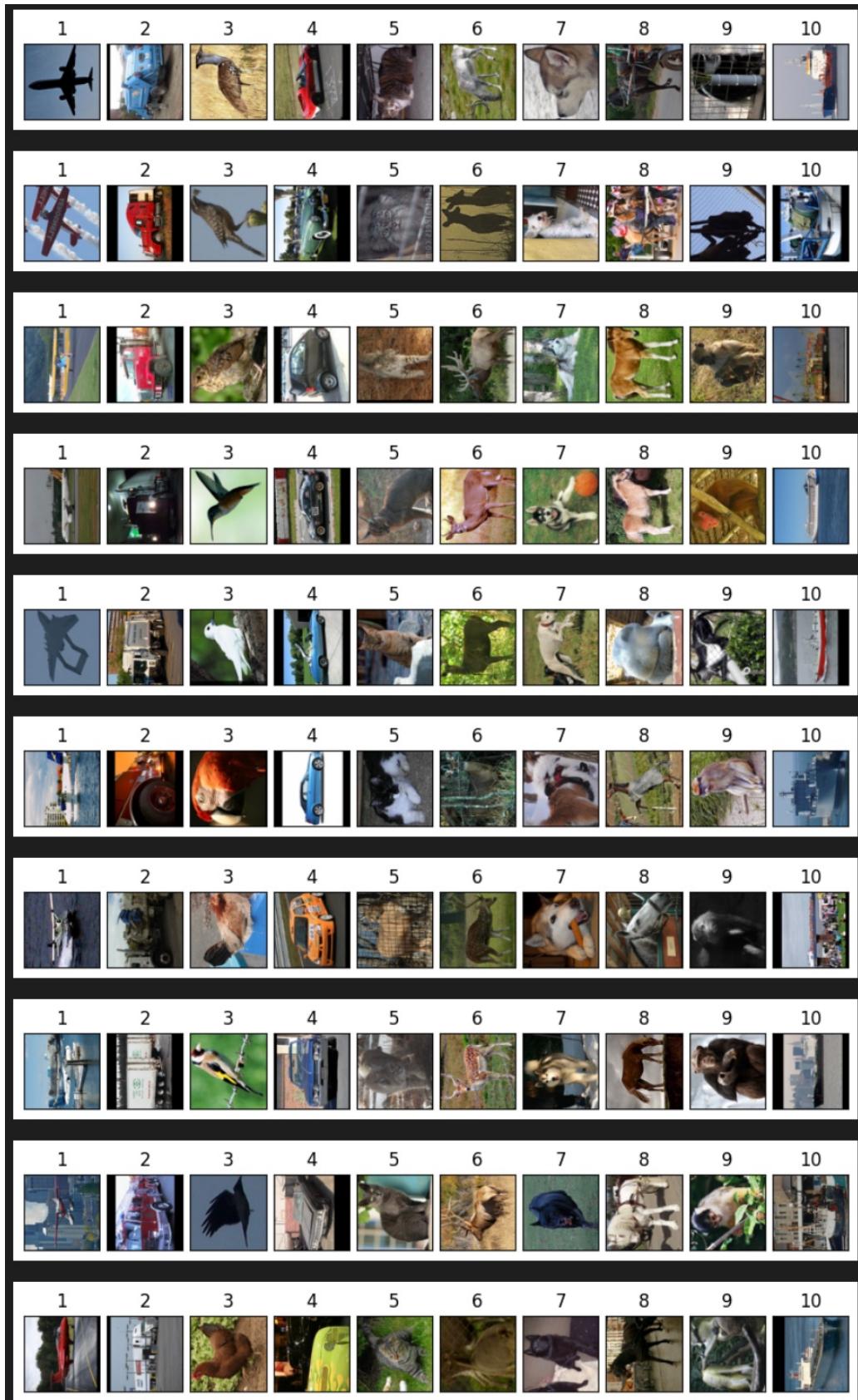


Figura 6: Exemplu inter-clasa pentru test.

Concluzie Generală (Inter-clasă) Trăgând linie, analiza acestor confuzii ne arată capcanele setului de date:

- Dacă modelul se bazează prea mult pe culoare, va confunda animalele între ele (toate sunt maro/negre).
- Dacă se bazează doar pe forme generale, va confunda mașinile cu camioanele.
- Iar dacă învață contextul (oameni în poză), riscă să greșească atunci când calul apare singur.

Aceste observații justifică de ce avem nevoie de o rețea convoluțională capabilă să observe detalii fine (cum ar fi forma urechilor sau scara vehiculului), nu doar statistici globale de culoare sau contur.

2 Antrenarea Rețelelor Neurale

În această etapă, am implementat și evaluat două arhitecturi distincte: un Perceptron Multistrat (MLP) și o Rețea Convoluțională (CNN). Obiectivul a fost maximizarea performanței, investigând impactul arhitecturii asupra capacitatei de generalizare, precum și transferul de cunoștințe (Fine-Tuning) pe un set de date secundar.

2.1 Arhitectura MLP (Baseline) - Analiza Overfitting-ului

Ca punct de referință, am antrenat o rețea de tip MLP (Perceptron Multistrat) cu arhitectura: $Input(Flatten) \rightarrow 256 \rightarrow 256 \rightarrow Output$.

Scopul acestui experiment a fost de a demonstra limitările rețelelor dense în procesarea imaginilor brute, comparativ cu rețelele convoluționale.

Analiza Curbelor de Învățare (Figura 7) Graficul evidențiază un caz manual de Overfitting sever, caracterizat prin:

- **Memorare mecanică (Train Loss ≈ 0)**: Linia pentru train scade continuu, semn că rețeaua, având un număr enorm de parametri în primul strat ($3 \times 96 \times 96 \times 256 \approx 7$ milioane de conexiuni), a reușit să memoreze pixelii imaginilor de antrenare.
- **Incapacitate de Generalizare (Val Loss \nearrow)**: Linia pentru validare diverge rapid față de cea de antrenare. În loc să scadă, loss-ul pe validare începe să crească, indicând că modelul devine din ce în ce mai "sigur" pe predicții greșite atunci când vede date noi.

Cauza Teoretică: Problema Aplatizării Eșecul este cauzat de operația de Flatten. Transformând imaginea 3D ($3 \times 96 \times 96$) într-un vector 1D, se distrug complet informația spațială (vecinătatea pixelilor). Astfel, MLP-ul nu învață că "o roată este rotundă", ci învață că "dacă pixelul 500 și pixelul 501 sunt negri, atunci e mașina". Dacă în setul de testare mașina este deplasată cu câțiva pixeli, regula învățată nu mai este valabilă.

Analiza Confuziilor (Figurile 8 și 9): Analizând matricele de confuzie, am observat erori sistematice care se repetă atât pe setul de validare, cât și pe cel de testare, evidențiind lipsa capacitatei de discriminare a formelor:

- **Confuzii în Grupul Biologic (Clasa 7):** Clasa 7 este identificată slab, fiind frecvent confundată cu **Clasa 5 (Cat)** și **Clasa 9 (Monkey)**. Faptul că aceste animale sunt amestecate între ele demonstrează că MLP-ul nu ”vede” forma obiectului (coarne vs urechi vs coadă), ci ia decizii bazate exclusiv pe statistici de culoare și textură (toate având blană și nuanțe de maro/gri).
- **Lipsa Invariantiei Spațiale:** Prin aplicarea operației de *flatten*, se distrug strucțura spațială a imaginii și vecinătatea pixelilor. Astfel, modelul nu poate învăța forma și structura unui câine comparativ cu cea a unei pisici, ci doar memorează prezența unor intensități de pixeli în zone fixe ale imaginii. Dacă subiectul nu este perfect centrat sau are o postură diferită, rețeaua nu îl mai poate recunoaște corect.

Practic, MLP-ul realizează o clasificare bazată pe ”ghicit” statistic, ignorând strucțura locală a obiectelor, ceea ce duce la performanță scăzută de aprox. 45%.

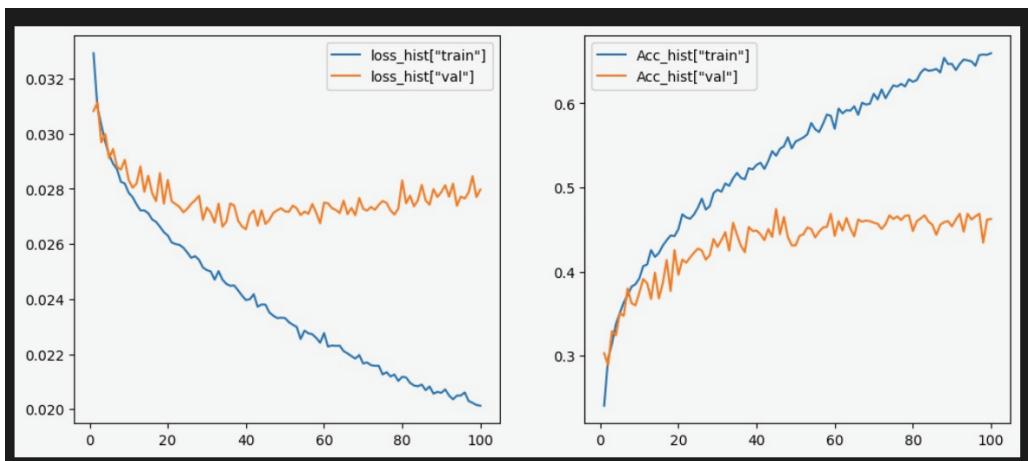


Figura 7: Curbele Loss/Acc pentru MLP. Se observă incapacitatea modelului de a atinge o performanță ridicată, plafonându-se rapid.

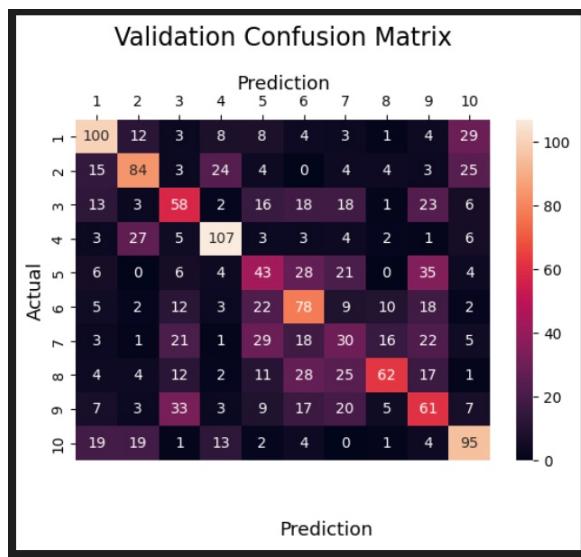


Figura 8: Matrice Confuzie MLP (Validare)

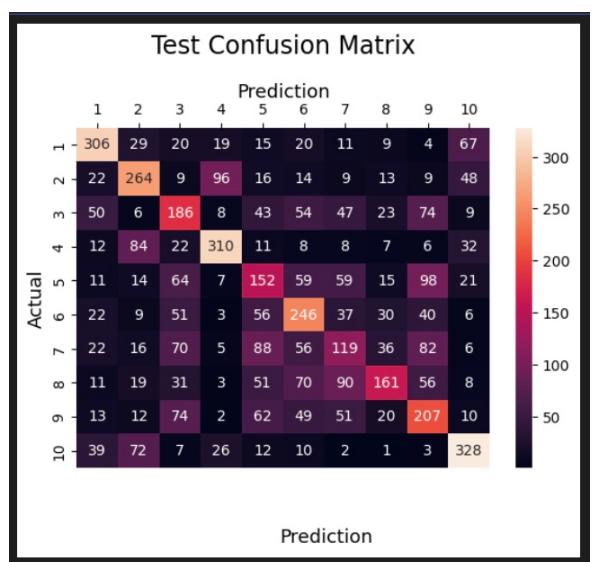


Figura 9: Matrice Confuzie MLP (Test)

2.2 Arhitectura CNN (Modelul Propus)

Trecerea la arhitectura conoluțională (CNN) a rezolvat problema generalizării. Modelul utilizează filtre 3×3 și straturi de Pooling pentru a extrage trăsături ierarhice, fiind antrenat timp de 100 de epoci.

Performanță:

- Modelul atinge o acuratețe pe setul de test de **69.78%**, o creștere majoră față de MLP (+24%).
- Așa cum se vede în Figura 10, curbele de antrenare și validare rămân apropiate, semn că regularizarea (Dropout, Weight Decay, BatchNorm) funcționează corect.
- Matricele de confuzie (Figurile 11 și 12) arată o diagonală principală mult mai pronunțată, deși persistă confuzii între clasele similare (Truck vs Car).

Interpretarea Curbelor de Învățare (Figura 10): Spre deosebire de MLP, graficele pentru CNN indică o învățare sănătoasă:

- **Convergență:** Loss-ul scade constant, iar acuratețea pe validare urcă până la un platou de aprox. 70%.
- **Generalization Gap:** Există o diferență între Train (82%) și Validare (70%), ceea ce indică un ușor overfitting, ținut sub control de Dropout și Augmentare. Salturile vizibile corespund intervenției scheduler-ului care a redus rata de învățare.

Analiza Detaliată a Erorilor (Figurile 11 și 12): Matricea de test relevă o distincție clară între performanța pe vehicule și cea pe animale:

- **Colapsul Clasei 7 (Deer):** Punctul critic al modelului este Clasa 7, care are doar 202 predicții corecte (acuratețe de 40%). Aceasta este masiv dispersată în tot spectrul animalelor, fiind confundată cu Maimuțele (77 cazuri) și Caii (67 cazuri).
- **Confuziile Clasei 5 (Cat):** Similar cu Clasa 7, Clasa 5 este slab definită (doar 256 corecte). Analiza erorilor arată o distribuție uniformă a confuziilor către celelalte mamifere:
 - 59 confuzii cu Clasa 6 (Dog);
 - 59 confuzii cu Clasa 7 (Deer);
 - 56 confuzii cu Clasa 9 (Monkey).

Acest tipar demonstrează că rețeaua detectează corect textura ("blană") și clasa generală ("mamifer patruped"), dar rezoluția și capacitatea modelului nu sunt suficiente pentru a distinge detaliile fine morfologice (bot de câine vs bot de pisică) în lipsa culorilor distinctive.

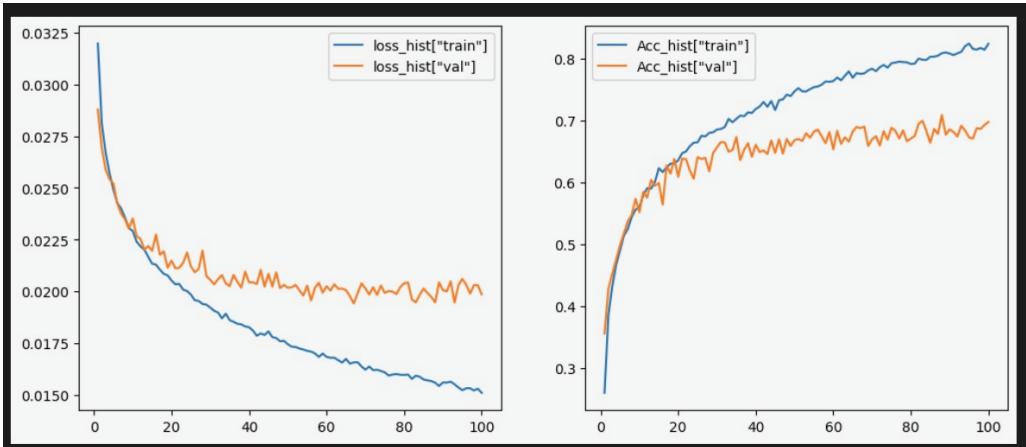


Figura 10: Curbele Loss/Acc pentru CNN Standard (Convergență stabilă).

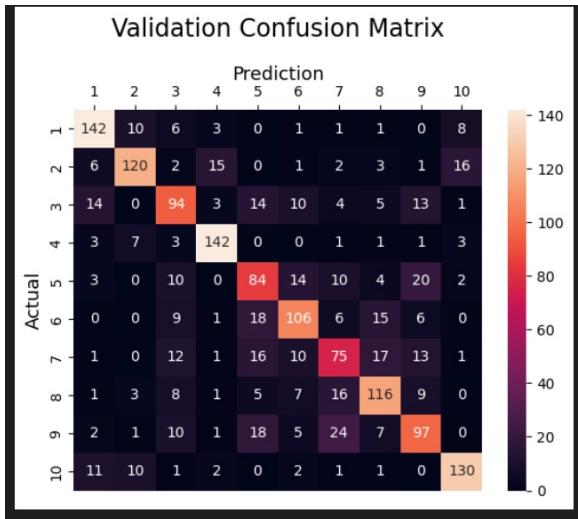


Figura 11: Matrice Confuzie CNN (Validare).

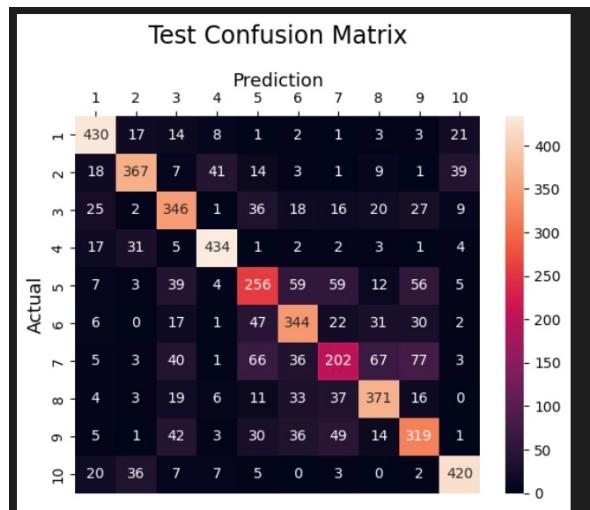


Figura 12: Matrice Confuzie CNN (Test).

2.3 Strategii de Fine-Tuning

Pentru adaptarea modelului la setul de date cu suport redus ("Land Patches"), am testat două strategii distincte, pornind de la ponderile modelului CNN antrenat anterior.

2.3.1 Strategia A: Frozen Layers

Am înghețat toate straturile convoluționale și am antrenat doar clasificatorul final timp de 50 de epoci.

- **Rezultat:** Acuratețe Test: 68.74%.
- **Observație:** Curbele (Fig. 13) sunt stabile, dar performanța este ușor sub cea a modelului de bază, sugerând că trăsăturile extrase inițial nu sunt perfect aliniate cu noul domeniu fără o mică ajustare.

- **Îmbunătățire pe Clasa 5 (Cat):** Performanța a crescut ușor pentru pisici. Acest lucru sugerează că trăsăturile de textură (blană) învățate de CNN-ul original sunt robuste și transferabile direct.
- **Degradare pe Clasa 7 (Deer):** În schimb, performanța pentru căprioare a scăzut. Deoarece straturile convoluționale nu s-au putut ajusta, modelul nu a putut rafina trăsăturile specifice (cum ar fi detaliile coarnelor sau silueta în pădure), rămânând blocat cu filtrele generale care nu erau optimizate pentru aceste detalii fine.

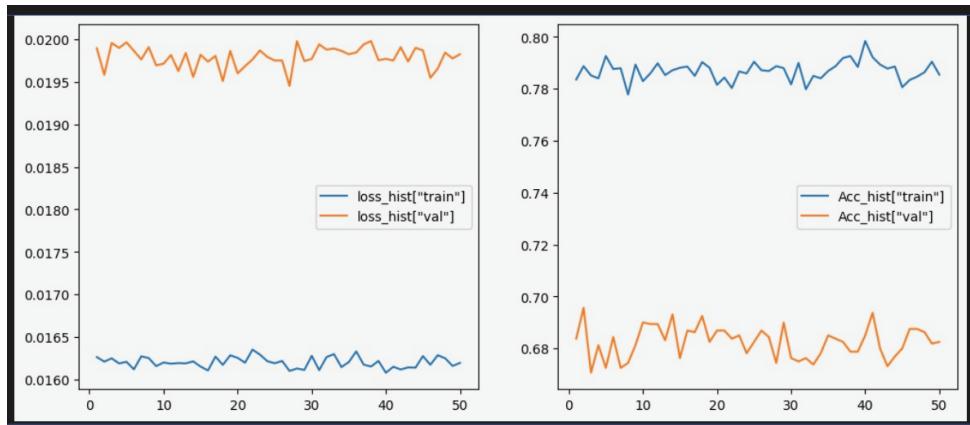


Figura 13: Curge Fine-Tuning (Frozen Layers).

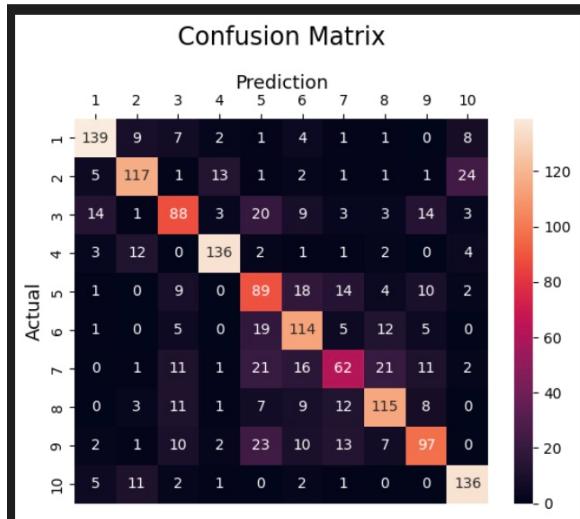


Figura 14: CM Frozen (Validare).

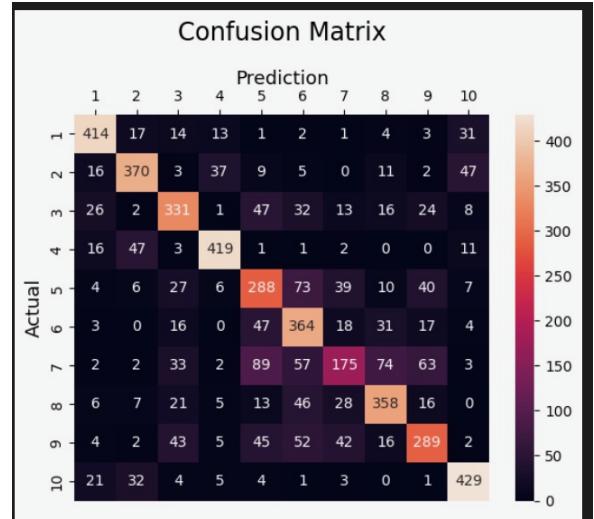


Figura 15: CM Frozen (Test).

2.3.2 Strategia B: Differential Learning Rates (Best Performance)

Am dezghețat întreaga rețea, dar am folosit rate de învățare diferite: un LR foarte mic ($3e - 6$) pentru straturile convoluționale (pentru a nu distrugă filtrele) și un LR standard ($3e - 4$) pentru clasificator.

- **Rezultat:** Acuratețe Test: 69.66%.

- **Observație:** Această strategie (Fig. 16) a oferit cea mai bună balanță, permitând modelului să se adapteze fin la noile date fără a intra în overfitting catastrofal.

Analiza Comparativă a Claselor (Fig. 18 vs Baseline): Analizând matricea de confuzie pe Test (Figura 18), observăm că strategia *Diff LR* a replicat și accentuat comportamentul observat la strategia *Frozen*, făcând un schimb de performanță între clase:

- **Consolidarea Cunoștințelor (Clasa 5 - Cat):** Performanța a crescut spectaculos de la 256 (Baseline) la 288 de detecții corecte. Deoarece rata de învățare pe filtre a fost mică, modelul a conservat și rafinat filtrele de textură (blană), devenind ”expert” în recunoașterea pisicilor.
- **Degradarea Claselor Ambigue (Clasa 7 - Deer):** În schimb, performanța pe căprioare a scăzut de la 202 (Baseline) la 175. Acest lucru confirmă că, pe un set de date mic, rețeaua tinde să sacrifice clasele cu trăsături vizuale complexe (coarne, pădure) pentru a minimiza eroarea globală, favorizând clasele cu pattern-uri mai ușor de recunoscut (cum sunt pisicile).

Concluzie: Deși media globală a crescut la 69.66%, acest câștig nu este uniform. Strategia *Differential LR* a funcționat prin ”ascuțirea” trăsăturilor deja puternice, dar nu a avut suficiente date noi pentru a rezolva confuziile structurale ale claselor dificile.

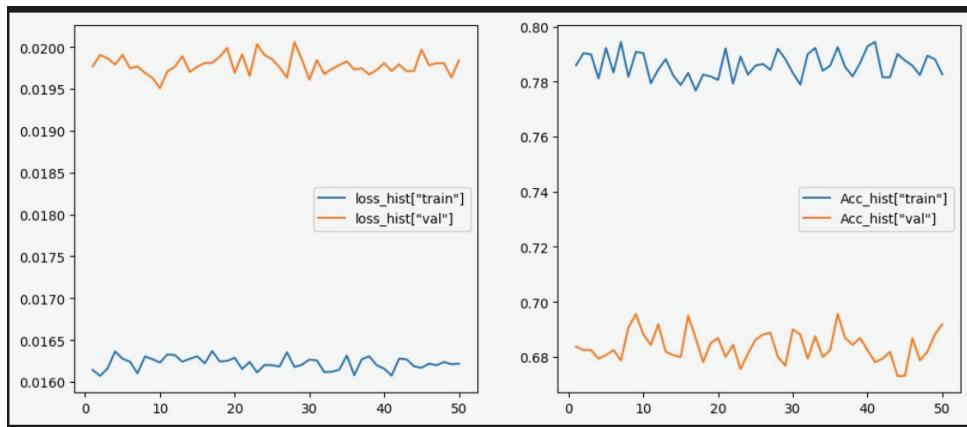


Figura 16: Curge Fine-Tuning (Differential LR) - Cea mai stabilă învățare.

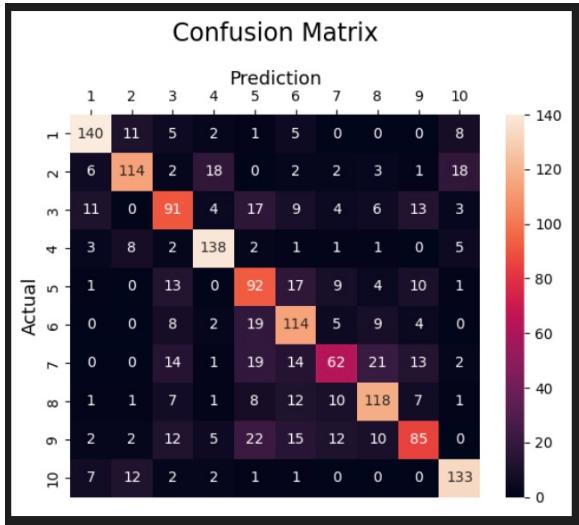


Figura 17: CM Diff LR (Validare).

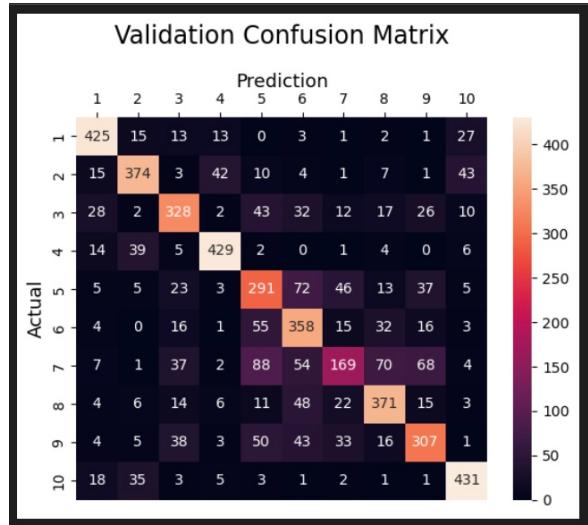


Figura 18: CM Diff LR (Test).

2.4 Rezultate Centralizate

Tabelul de mai jos sintetizează performanța tuturor experimentelor rulate pe setul de Test. Se observă superioritatea clară a arhitecturilor convoluționale față de MLP.

Tabela 1: Compararea metricilor de performanță pe setul de Test

Model	Epoci	Strategie / Modificări	Acc	F1-score	Precision	Recall
CNN (Base)	100	Baseline (Adam, LR schedule)	0.6978	0.6946	0.6938	0.6978
CNN (FT Frozen)	50	Frozen layers (except FC)	0.6874	0.6830	0.6855	0.6874
CNN (FT Diff LR)	50	Different LR per layers	0.6966	0.6915	0.6941	0.6966
MLP	100	Baseline (Aceiași hiperparametri)	0.4558	0.4538	0.4577	0.4558

Concluzie Finală: Experimentele confirmă că arhitectura CNN_ cu optimizatorul Adam și scheduler ReduceLROnPlateau este cea mai robustă soluție. MLP-ul eșuează din cauza lipsei invariantei spațiale, iar Fine-Tuning-ul cu rate diferențiate s-a dovedit a fi strategia optimă pentru adaptarea la domenii noi, păstrând aproape integral performanța modelului original.

3 Concluzii Partea 1

Experimentele au demonstrat superioritatea arhitecturilor convoluționale pentru date vizuale. Deși MLP-ul poate învăța corelații simple, CNN-ul capturează ierarhii vizuale esențiale pentru clasificarea obiectelor complexe.

4 Partea 2: Clasificare pe Setul de Date "Land Patches"

4.1 Explorarea Datelor (EDA)

Setul de date *Land Patches* conține imagini satelitare multispectrale (RGB) de rezoluție 64×64 pixeli, clasificate în 10 categorii de utilizare a terenului (*AnnualCrop*, *Forest*, *Herbaceous Vegetation*, *Highway*, *Industrial*, *Pasture*, *PermanentCrop*, *Residential*, *River*, *SeaLake*).

4.1.1 Analiza Detaliată a Variabilității Intra-clasă

Analiza vizuală a esantioanelor (Figura 21) relevă provocări specifice imaginilor de teledetectie, unde variația nu provine din forma obiectului (ca la animale), ci din textura și compoziția spectrală a suprafeței.

Variabilitatea Cromatică a Apei (SeaLake) O inspecție a clasei *SeaLake* arată o inconsistență cromatică semnificativă. Deși ne-am așteptă ca apa să fie albastră:

- Multe lacuri apar în nuanțe de verde închis, turcoaz sau chiar maroniu, în funcție de turbiditate, adâncime și prezența algelor.
- Această varianță face imposibilă definirea unui simplu "filtru de culoare albastră" pentru detecția apei. Modelul trebuie să învețe textura specifică a suprafeței lichide (reflexii, uniformitate) pentru a nu confunda un lac verde cu o pădure.

Complexitatea Texturală a Zonelor Urbane (Residential) Clasa *Residential* prezintă cea mai mare complexitate internă. Spre deosebire de o pădure (textură uniformă), o zonă rezidențială este un mozaic eterogen:

- Imaginele conțin combinații de acoperișuri (roșii, gri, albe), străzi asfaltate, vegetație intersticială și umbre lăsate de clădiri.
- Densitatea clădirilor variază enorm: de la cartiere dense cu case mici, la zone cu blocuri rare și mult spațiu verde. Această lipsă de omogenitate forțează rețeaua să învețe pattern-uri structurale complexe (linii drepte, unghiuri de 90 de grade) și nu doar culori.

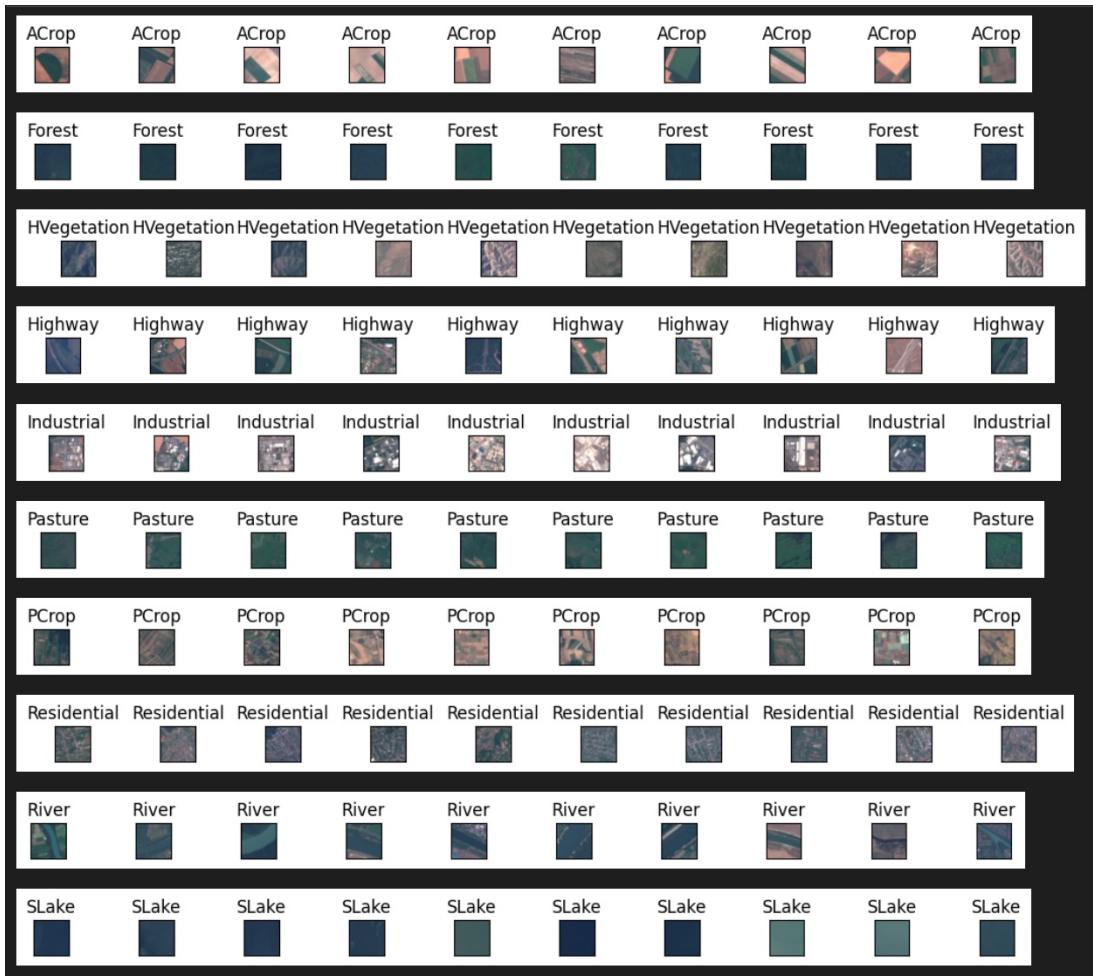


Figura 19: Variabilitate intra-clasă aplicat pentru antrenare: Observați diferențele de culoare la *SeaLake* și diversitatea structurilor la *Residential*.

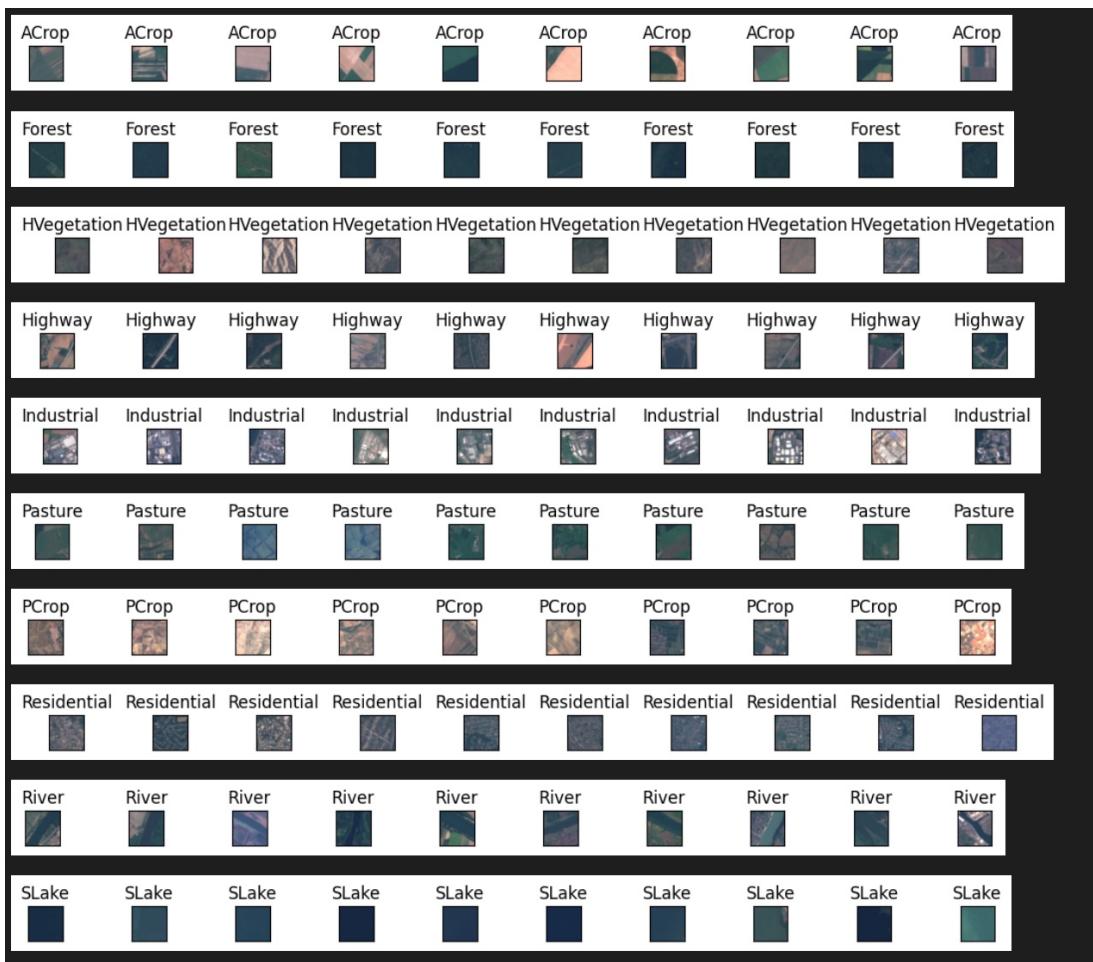


Figura 20: Variabilitate intra-clasă aplicat pentru validare



Figura 21: Variabilitate intra-clasă aplicat pentru testare

4.1.2 Analiza Distribuției Claselor

Verificarea echilibrului setului de date (Figurile 23 și 24) arată o distribuție perfect uniformă.

- Setul de antrenare conține **200 de imagini** per clasă.
- Setul de testare conține **700 de imagini** per clasă.

Acest echilibru este ideal pentru antrenare, eliminând riscul ca modelul să ignore clasele minoritare (ex: râurile) în favoarea celor dominante.

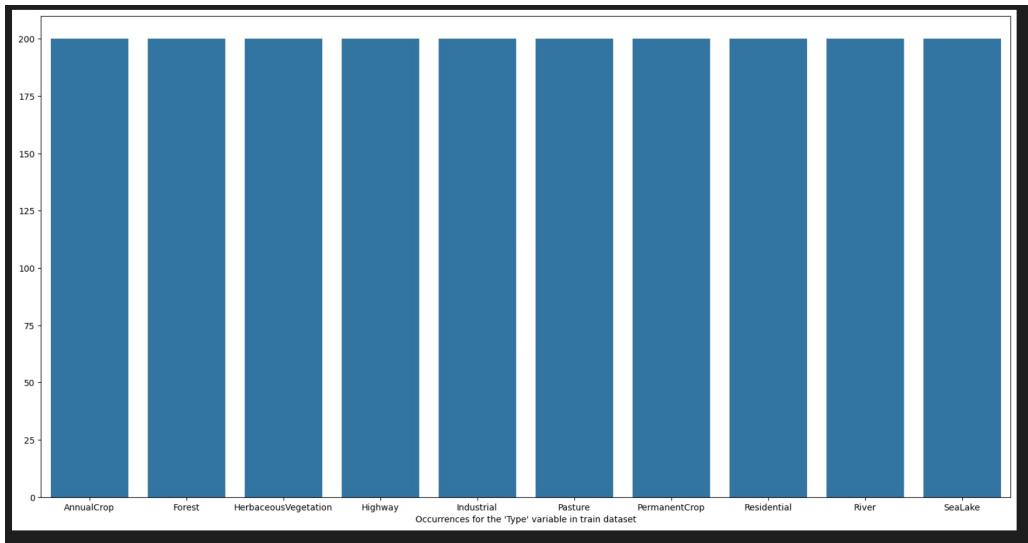


Figura 22: Distribuția uniformă a imaginilor în setul de antrenare (200 per clasă).

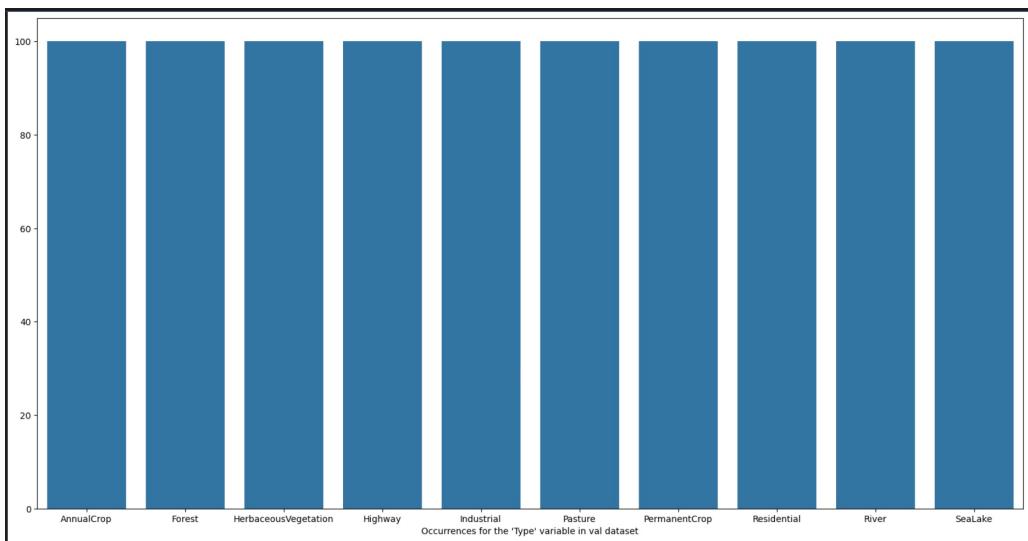


Figura 23: Distribuția uniformă a imaginilor în setul de validare (100 per clasă).

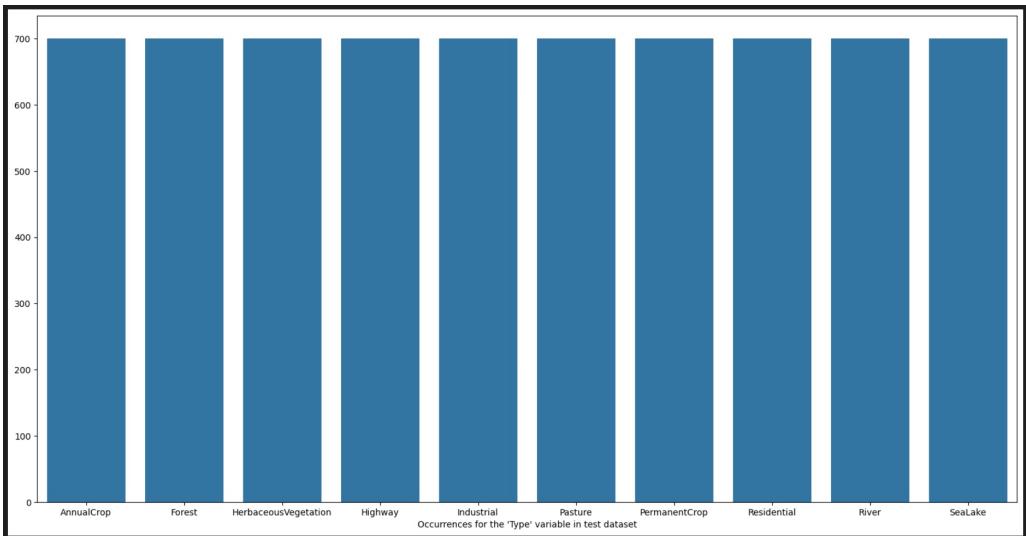


Figura 24: Distribuția uniformă a imaginilor în setul de testare (700 per clasă).

4.1.3 Analiza Inter-clasă (Similarități și Riscuri de Confuzie)

Principala provocare a acestui set de date o reprezintă similitudinile vizuale puternice între clase distincte semantic. Am identificat două zone critice de confuzie:

Confuzia ”Verde pe Verde” (Forest vs. Pasture vs. SeaLake) Clasele naturale împart o paletă cromatică dominată de verde, ceea ce poate deruata rețeaua:

- **Forest vs. Pasture:** Ambele sunt suprafețe verzi. Diferența este dată de textură (coroana copacilor e rugoasă, pășunea e fină). La rezoluția mică de 64×64 , detaliile texturale se pot pierde, transformând imaginea într-un bloc verde uniform, greu de diferențiat.
- **Forest vs. SeaLake (Verde):** Așa cum am menționat la variabilitatea intra-clasă, lacurile cu apă verde-tulbure pot fi confundate ușor cu o pădure densă privită de sus, dacă modelul se bazează exclusiv pe histogramele de culoare.

Ambiguitatea Geometrică (Highway vs. River) Cea mai interesantă problemă este cea morfologică. Atât *Highway* (Autostradă) cât și *River* (Râu) arată ca niște ”linii curbe continue care traversează imaginea”.

- **Contextul induce în eroare:** Adesea, autostrăzile trec prin zone verzi (câmpii), exact ca râurile. Dacă râul este îngust sau are o culoare gri-albăstruie (care seamănă cu asfaltul), confuzia este aproape garantată pentru un model simplu.
- Rețeaua trebuie să distingă subtilități: autostrăzile au margini mai drepte și uniforme, râurile sunt mai sinuoase și neregulate.



Figura 25: Exemple de similaritate Inter-clasă pentru antrenare: Observați similitudinea de formă dintre *Highway* și *River*, precum și dominanța verdelui la *Forest/Pasture*.

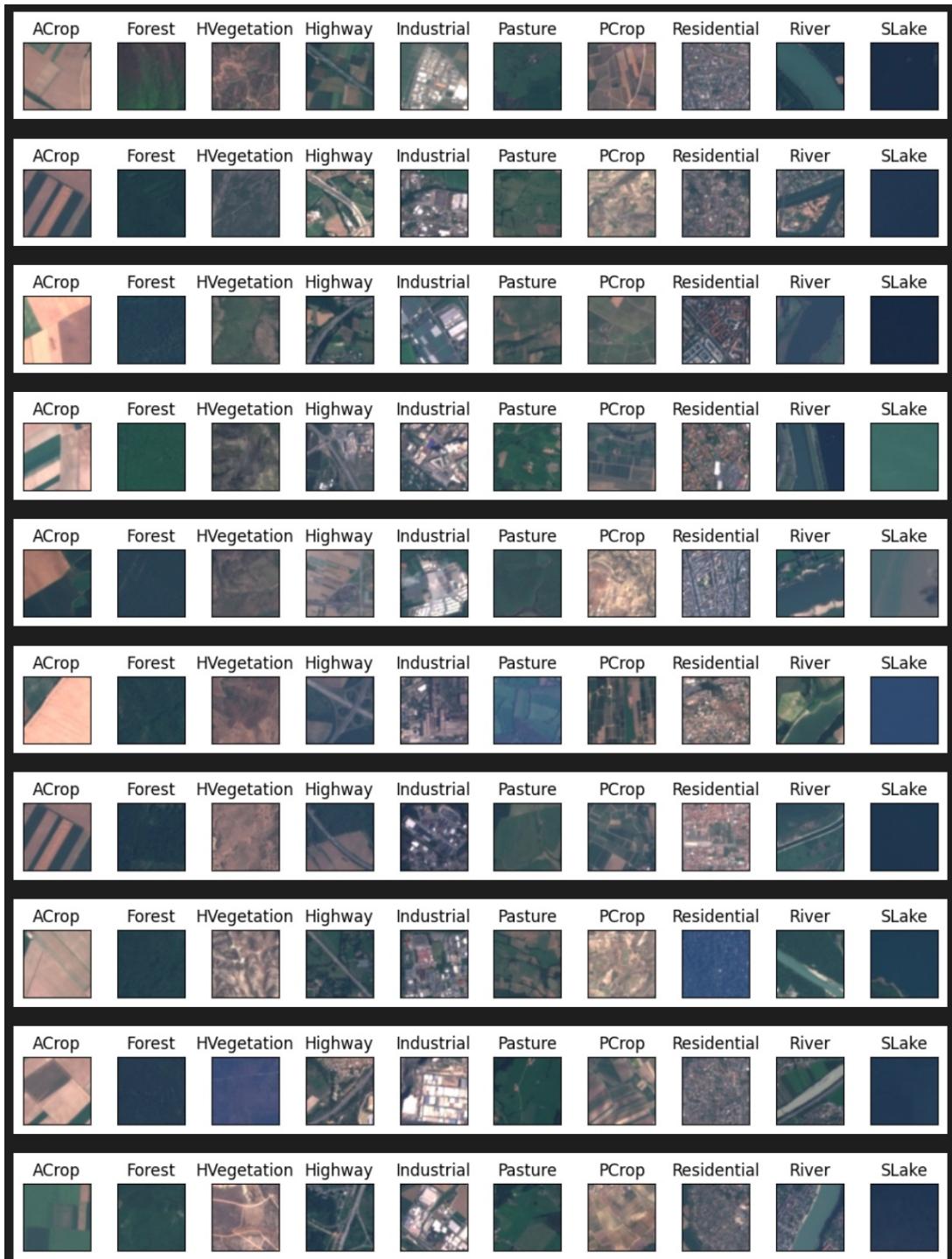


Figura 26: Exemple de similaritate Inter-clasă pentru validare

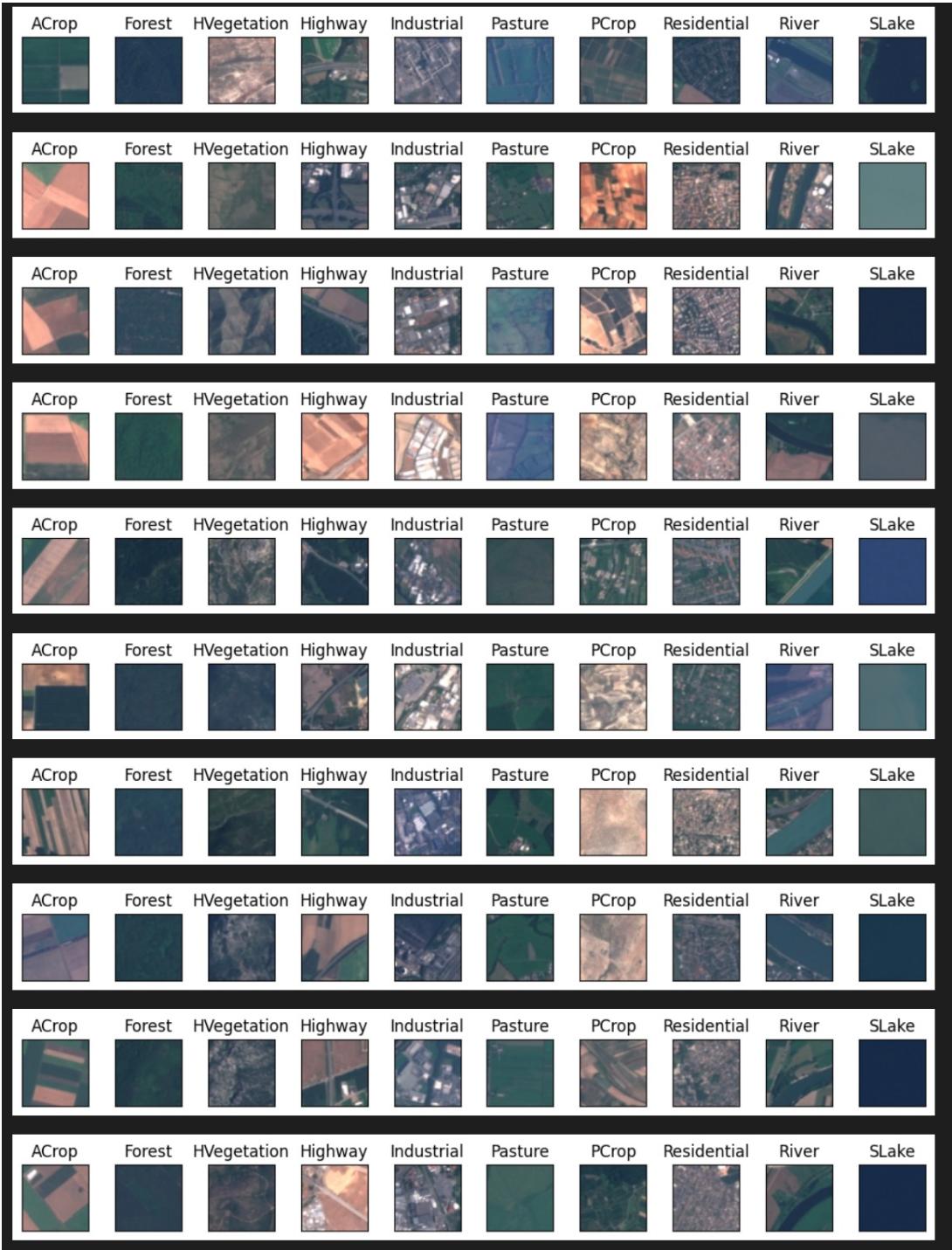


Figura 27: Exemplu de similaritate Inter-clasă pentru testare

Concluzie Generală (Land Patches) Analiza EDA sugerează că sarcina nu este trivială. Deși avem multe date, clasele sunt separate de nuanțe fine de textură și formă. Un model care ignoră textura (bazându-se doar pe culori medii) va eșua lamentabil în a distinge pădurile de pășuni sau lacurile verzi. De asemenea, modelul are nevoie de o capacitate de extragere a formelor (edge detection) superioară pentru a nu confunda curbele râurilor cu cele ale autostrăzilor.

5 Antrenarea Rețelelor Neurale pe Setul "Land Patterns"

Similar cu prima parte a proiectului, am început prin stabilirea unui punct de referință (baseline) folosind o arhitectură densă (MLP), pentru a justifica necesitatea utilizării rețelelor convoluționale și a strategiilor de Transfer Learning pe imaginile satelitare.

5.1 Arhitectura MLP (Baseline) - Analiza Eșecului pe Texturi

Am antrenat o rețea MLP cu arhitectura: $Input(Flatten64 \times 64) \rightarrow 256 \rightarrow 256 \rightarrow Output$.

Scopul a fost evaluarea capacitații unei rețele dense de a distinge tipuri de teren bazându-se doar pe informația spectrală brută, fără a procesa textura spațială.

Analiza Curbelor de Învățare (Figura 28) Graficul ilustrează un caz de instabilitate și plafonare, diferit de overfitting-ul "curat" de la Imagebits:

- **Convergență Limitată:** Acuratețea pe setul de validare (linia portocalie) se plafonează rapid în jurul valorii de 45-46%, refuzând să crească indiferent de numărul de epoci.
- **Divergență Loss-ului:** În timp ce *Train Loss* (albastru) scade constant (memorare), *Validation Loss* stagnează și are fluctuații, semn că modelul nu reușește să găsească reguli generalizabile pentru texturile complexe (ex: zone rezidențiale vs industriale).

Cauza Teoretică: Distrugerea Texturii În imaginile satelitare, clasa este definită de textură (aranjamentul spațial al pixelilor), nu de o formă centrală. Operația de **Flatten** distrugе tocmai această informație. Un MLP nu poate face diferență între o "pădure" (textură rugoasă, verde) și o "pășune" (textură fină, verde), deoarece vede doar o "supă" de pixeli verzi, pierzând relația de vecinătate care definește rugozitatea.

Analiza Confuziilor (Figurile 29 și 30): Matricea de testare scoate la iveală probleme majore de discriminare, mult mai pronunțate decât la setul anterior:

Cauza Teoretică: Distrugerea Texturii În imaginile satelitare, clasa este definită de textură (aranjamentul spațial al pixelilor), nu de o formă centrală. Operația de **Flatten** distrugе tocmai această informație. Un MLP nu poate face diferență între o "pădure" (textură rugoasă, verde) și o "pășune" (textură fină, verde), deoarece vede doar o "supă" de pixeli verzi, pierzând relația de vecinătate care definește rugozitatea.

Analiza Confuziilor (Figurile 29 și 30): Matricea de testare scoate la iveală probleme majore de discriminare, mult mai pronunțate decât la setul anterior:

- **Colapsul Clasei 3 (HerbaceousVegetation):** Cea mai gravă eroare a modelului se observă aici. Pe setul de test, doar **9 imagini** din această clasă au fost identificate corect.
 - Majoritatea covârșitoare a imaginilor *HerbaceousVegetation* (3) sunt confundate cu *PermanentCrop* (7).

- **Cauza:** Ambele clase reprezintă terenuri cu vegetație joasă/agricolă, având histograme de culoare aproape identice. Fără capacitatea de a analiza pattern-ul de creștere a plantelor (textura), MLP-ul le consideră aceeași clasă.
- **Bias-ul către Clasa 2 (Forest):** Analizând coloana 2, observăm un număr uriaș de False Positives. Modelul clasifică eronat multe alte clase ca fiind ”Forest”:
 - **SeaLake (10) → Forest (2):** 163 de erori. Lacurile întunecate sunt confundate masiv cu pădurile, MLP-ul bazându-se doar pe intensitatea pixelilor (dark pixels).
 - **Pasture (6) → Forest (2):** 133 de erori. Ambele sunt verzi, devenind indistinguibile pentru o rețea fără convoluții.
- **Confuzia Geometrică (Highway vs River):** O eroare remarcabilă apare între Autostrăzi (Clasa 4) și Râuri (Clasa 9). Clasa 4 este confundată cu 9 în **123 de cazuri**. Deoarece ambele clase reprezintă ”linii sinuoase”, MLP-ul le amestecă frecvent, bazându-se probabil doar pe culorile de fundal.

Concluzia este că MLP-ul acționează preponderent ca un clasificator de culoare medie, esuând în sarcini care necesită înțelegere structurală.

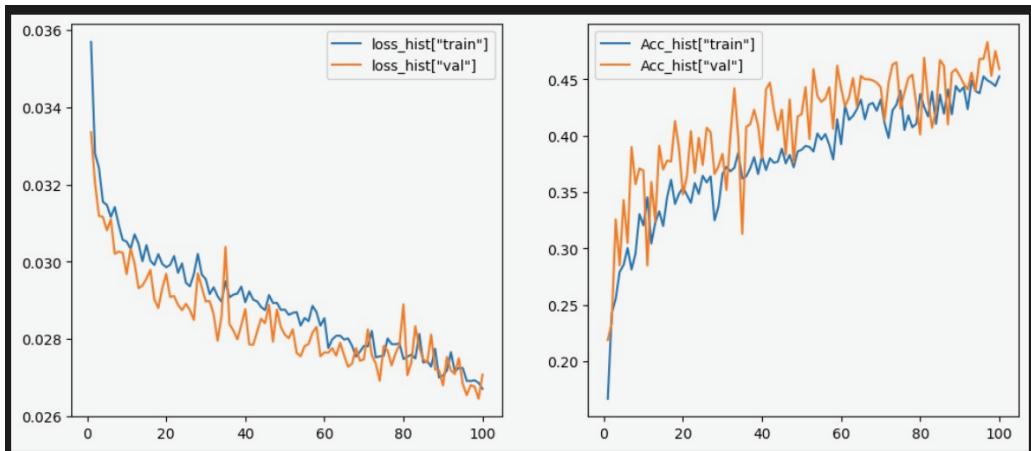


Figura 28: Curbele Loss/Acc pentru MLP pe Land Patches. Se observă plafonarea performanței sub 50%.

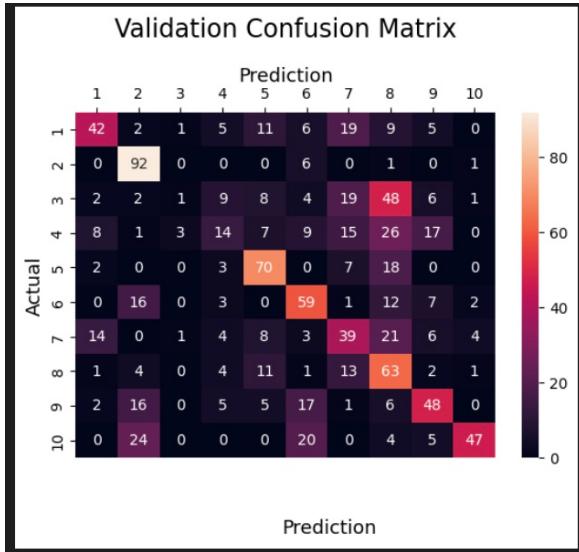


Figura 29: Matrice Confuzie MLP (Validare).

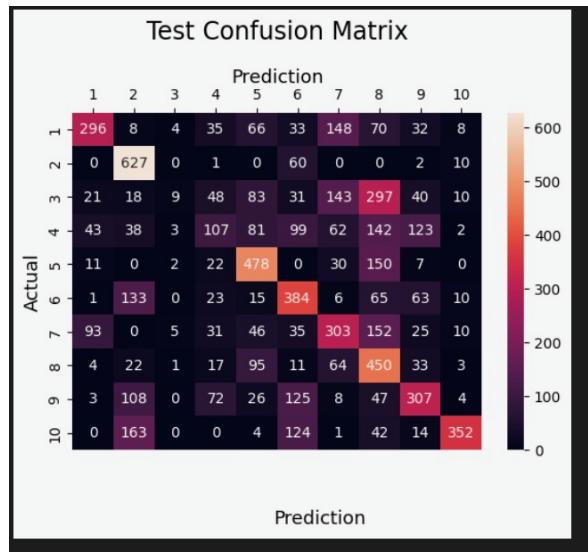


Figura 30: Matrice Confuzie MLP (Test) - Bias puternic către Clasa 2.

5.2 Arhitectura CNN (Modelul Propus)

Trecerea la arhitectura convoluțională (CNN_{_}) a schimbat dramatic rezultatele. Modelul a fost antrenat timp de 100 de epoci.

Performanță:

- Modelul atinge o acuratețe pe setul de test de **77.64%**, o creștere de +30% față de MLP.
- Curbele de învățare (Figura 31) arată o convergență sănătoasă, deși există un ”gap” între train și validare, sugerând că datele de teren sunt mult mai diverse decât cele din Imagebits.

Analiza Detaliată a Erorilor (Figura 33): Matricea de test arată că CNN-ul a rezolvat bias-ul de culoare, dar se confruntă cu două obstacole majore:

- **Problema Principală - Confuzia Vegetală (3 vs 7):** Deși s-a înregistrat un progres imens față de MLP (unde clasa 3 era aproape invizibilă), suprapunerea dintre *Herbaceous Vegetation* (3) și *PermanentCrop* (7) rămâne cea mai mare sursă de erori din matrice. Similitudinea vizuală extremă și definițiile semantice apropiate fac ca aceste două clase să fie greu de separat perfect chiar și pentru o rețea convoluțională.
- **Ambiguitatea Structurală (Highway vs River):** A doua problemă majoră este de natură geometrică, persistând chiar și cu utilizarea convoluțiilor:
 - 104 imagini reale *Highway* (4) sunt clasificate ca *River* (9).
 - 99 imagini reale *River* (9) sunt clasificate ca *Highway* (4).

La rezoluția de 64×64 , o autostradă gri și un râu tulbure arată aproape identic geometric (linii curbe continue), iar modelul nu are suficient context spațial pentru a le diferenția.

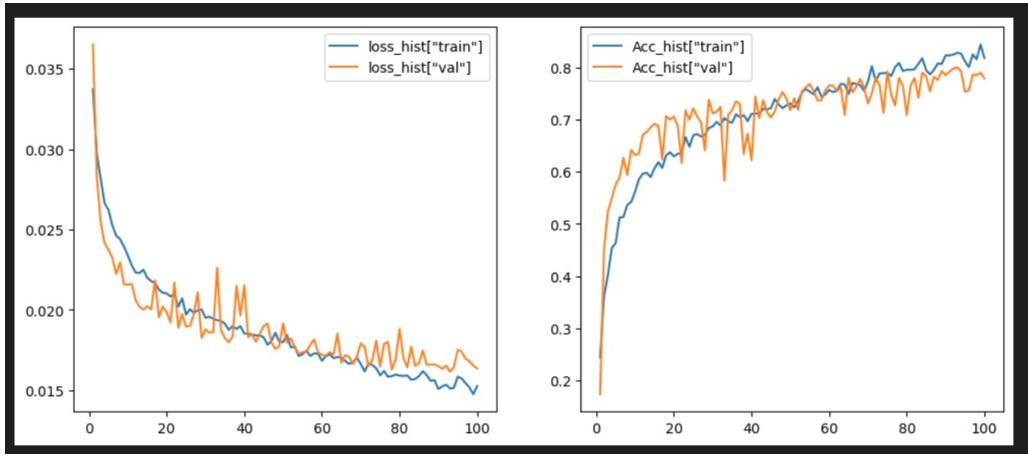


Figura 31: Curbele Loss/Acc pentru CNN Standard pe Land Patches.

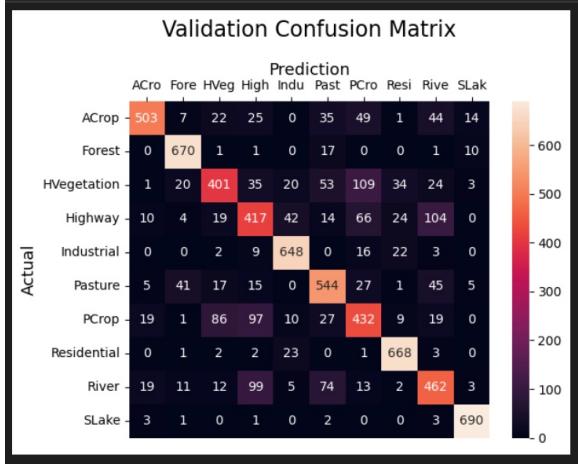


Figura 32: Matrice Confuzie CNN (Validare).

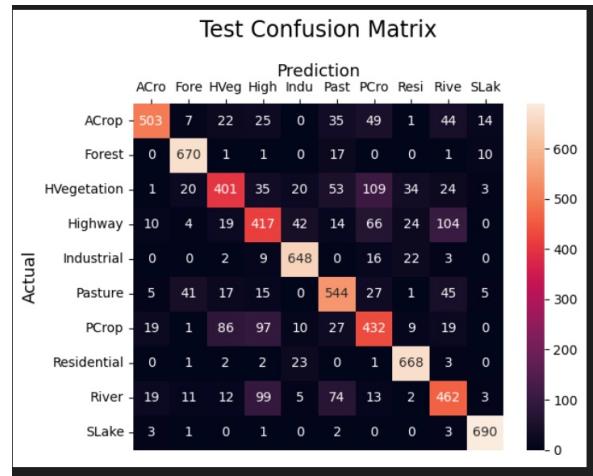


Figura 33: Matrice Confuzie CNN (Test).

5.3 Strategii de Fine-Tuning

Pentru această etapă, am utilizat modelul pre-antrenat pe Imagebits și l-am adaptat la setul Land Patches, testând ipoteza că trăsăturile învățate anterior pot stabiliza clasificarea noilor texturi.

5.3.1 Strategia A: Frozen Layers (Straturi Înghețate)

Am înghețat toate straturile convoluționale și am antrenat doar clasificatorul final timp de 50 de epoci.

- **Rezultat:** Acuratețe Test: **78.65%**. Aceasta este cea mai ridicată performanță obținută dintre toate experimentele.
- **Analiza Erorilor:** Această strategie a adus un beneficiu specific:
 - **Ameliorarea Geometriei (Highway vs River):** Confuzia dintre autostrăzi și râuri a scăzut comparativ cu antrenarea de la zero. Se pare că filtrele de

”muchii” și ”forme” învățate pe Imagebits sunt foarte robuste și au ajutat la distingerea mai clară a drumurilor.

- **Stagnare pe Vegetație (3 vs 7):** Totuși, confuzia dintre *Herbaceous Vegetation* și *PermanentCrop* a rămas neschimbată. Deoarece straturile de extracție a trăsăturilor au fost blocate, modelul nu și-a putut adapta filtrele de textură pentru a sesiza diferențele subtile dintre cele două tipuri de plante.

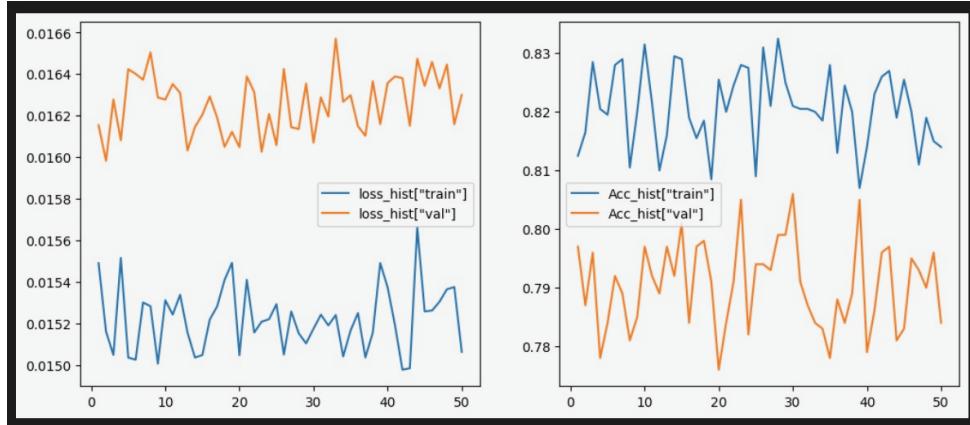


Figura 34: Curge Fine-Tuning (Frozen Layers).

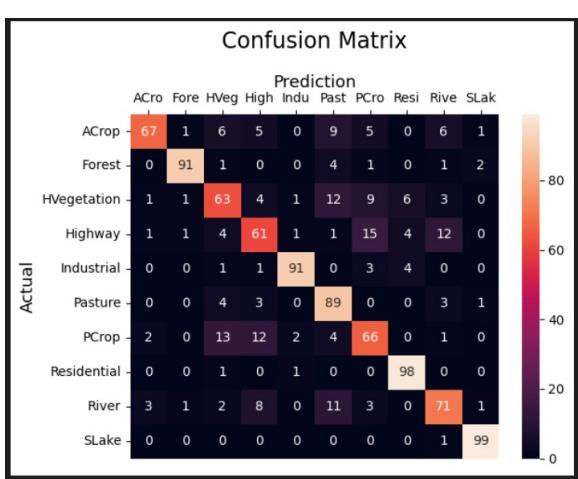


Figura 35: CM Frozen (Validare).

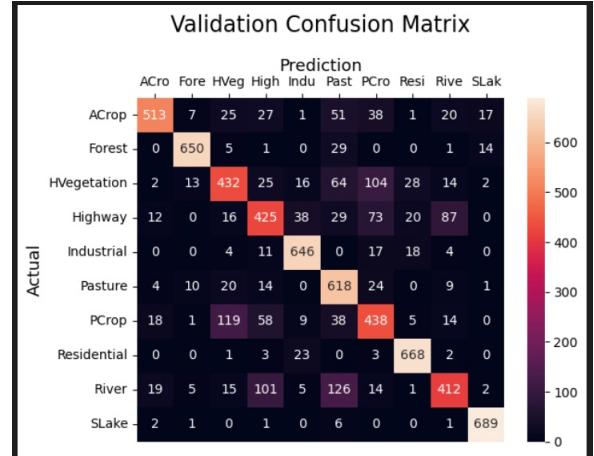


Figura 36: CM Frozen (Test) - Îmbunătățire strict structurală.

5.3.2 Strategia B: Differential Learning Rates (Diff LR)

Am dezghețat întreaga rețea, aplicând un Learning Rate foarte mic ($3e - 6$) pentru backbone și unul standard ($3e - 4$) pentru clasificator.

- **Rezultat:** Acuratețe Test: **78.44%**. Deși media este similară cu varianta Frozen, distribuția erorilor este mai echilibrată.
- **Analiza Erorilor (Fig. 39):** Această strategie a oferit o capacitate de adaptare superioară.

- **Rafinarea Texturii (3 vs 7):** Spre deosebire de varianta Frozen, aici s-a observat o usoară îmbunătățire a separării dintre clasele *Herbaceous Vegetation* (3) și *PermanentCrop* (7). Permitând filtrelor să se modifice puțin, rețeaua a început să învețe pattern-urile specifice de vegetație din noul set.
- **Menținerea Câștilegului Structural:** Simultan, modelul a păstrat și îmbunătățirea discriminării dintre *Highway* și *River*, demonstrând că Differential LR combină stabilitatea cunoștințelor anterioare cu flexibilitatea necesară noilor date.

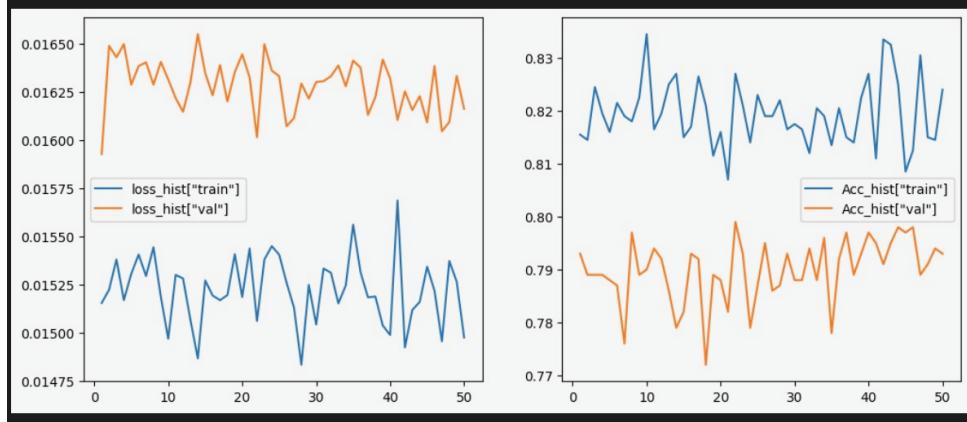


Figura 37: Curve Fine-Tuning (Differential LR).

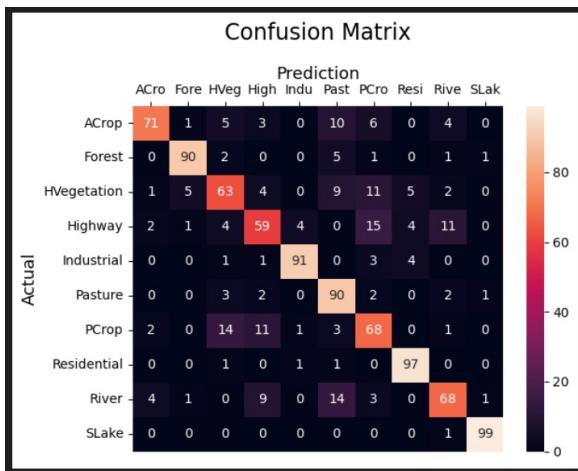


Figura 38: CM Diff LR (Validare).

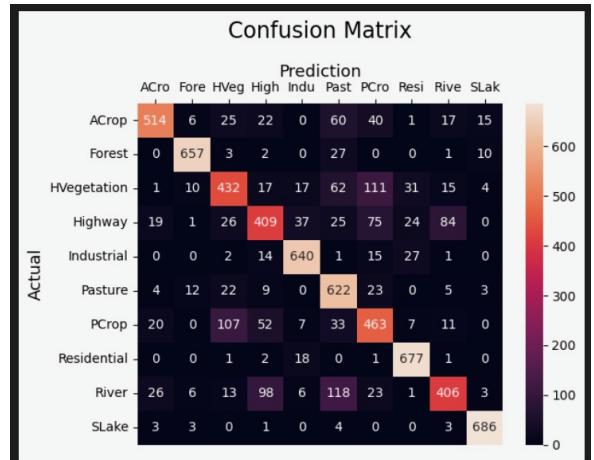


Figura 39: CM Diff LR (Test) - Rafinare suplimentară pe vegetație.

5.4 Rezultate Centralizate - Land Patches

Tabelul de mai jos compară performanța finală pe setul de testare pentru toate arhitecturile.

Tabela 2: Compararea metricilor de performanță pe setul de Test (Land Patches)

Model	Epoci	Strategie	Acc	F1-score	Precision	Recall
CNN (Base)	100	Antrenare de la zero	0.7764	0.7737	0.7757	0.7764
CNN (Frozen)	50	Fine-Tuning (Best)	0.7865	0.7836	0.7890	0.7865
CNN (Diff LR)	50	Fine-Tuning	0.7844	0.7823	0.7879	0.7844
MLP	100	Baseline	0.4732	0.4432	0.4874	0.4732

Concluzie Finală Pe setul Land Patches, Transfer Learning-ul (Strategy A: Frozen) a oferit cele mai bune rezultate, depășind antrenarea de la zero. Acest lucru confirmă că trăsăturile de nivel jos (muchii, colțuri) învățate pe un dataset generic sunt valoroase și transferabile chiar și într-un domeniu complet diferit (imagini satelitare), stabilizând antrenarea și îmbunătățind acuratețea finală.

6 Partea 2: Analiza de Sentiment pe Text în Limba Română

În a doua parte a proiectului, am abordat problema clasificării binare a sentimentului (Pozitiv vs. Negativ) folosind setul de date *Ro-Sent*. Obiectivul a fost construirea unui pipeline robust de procesare a limbajului natural (NLP) care să integreze curățarea textului, tokenizarea și utilizarea de embeddings pre-antrenate într-o arhitectură recurrentă.

6.1 Explorarea Datelor (EDA)

Setul de date conține 17.941 de mostre pentru antrenare și 11.005 pentru testare. Analiza preliminară a evidențiat caracteristicile structurale ale corpusului.

6.1.1 Echilibrul Claselor

O primă verificare esențială a fost distribuția etichetelor. Așa cum se observă în Figura ??, setul de date este **perfect echilibrat** atât pentru antrenare, cât și pentru testare. Acest lucru elimină necesitatea unor tehnici de re-eșantionare (oversampling/undersampling) și ne permite să utilizăm Acuratețea ca metrică principală de performanță, alături de F1-Score.

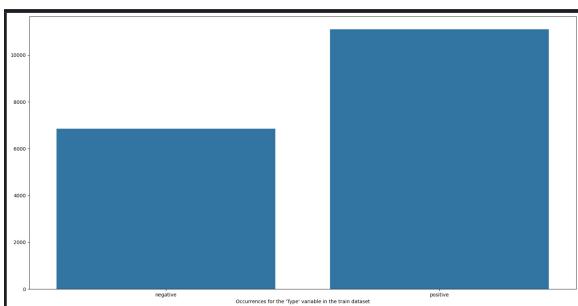


Figura 40: Distribuția claselor (Train) - Echilibrul perfect.

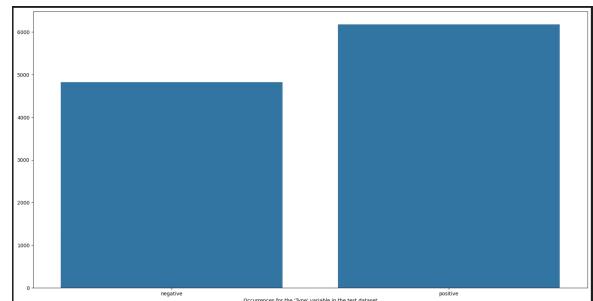


Figura 41: Distribuția claselor (Test).

6.1.2 Analiza Lungimii Textelor

Investigarea lungimii recenziilor (număr de caractere/cuvinte) a scos la iveală o distribuție **bimodală** interesantă, vizibilă în special în setul de antrenare (Figura 42):

- **Recenzi Scurte (Peak 1):** Există o concentrare mare de texte foarte scurte (sub 100 de caractere), probabil comentarii rapide de tipul "Excelent!" sau "Nu recomand.".
- **Recenzi Detaliate (Peak 2):** Al doilea vârf al distribuției apare în zona 600-800 de caractere, indicând utilizatori care argumentează detaliat opinia.

Această varianță mare (de la câteva cuvinte la paragrafe întregi) impune o strategie atentă de *padding*. Dacă stabilim o lungime fixă prea mare, recenziile scurte vor fi dominate de token-ul de padding (zero), ceea ce poate dilua semnalul pentru rețelele recurente.

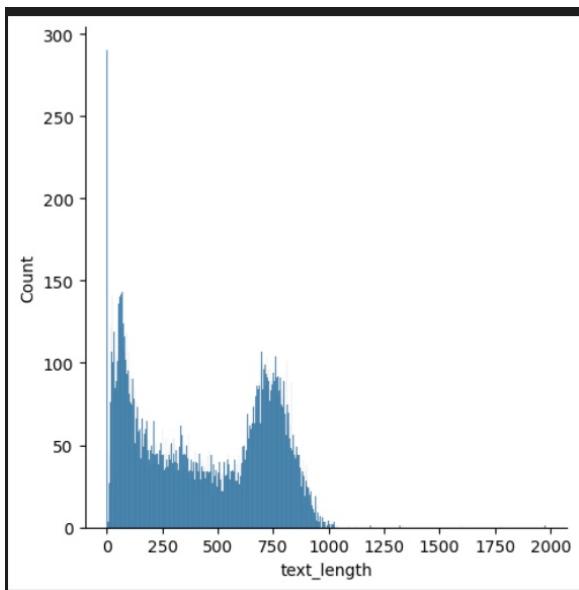


Figura 42: Histograma lungimii textelor (Train) - Distribuție bimodală.

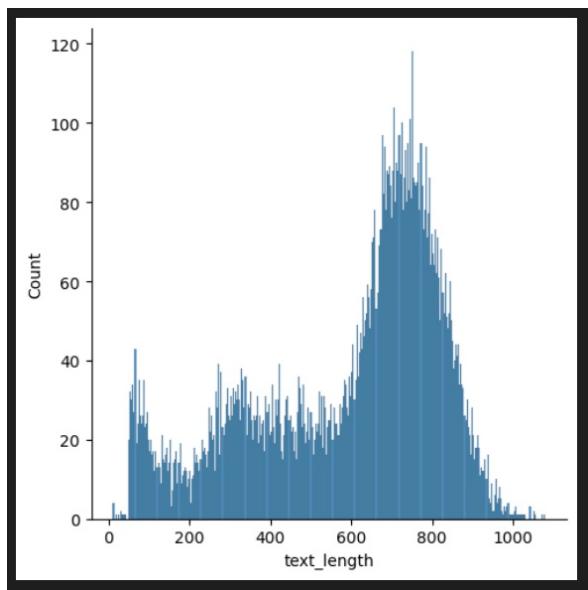


Figura 43: Histograma lungimii textelor (Test).

6.2 Pipeline de Preprocesare și Tokenizare

Pentru a transforma textul brut într-un format numeric acceptat de rețelele neurale, am implementat un pipeline de curățare și vectorizare.

6.2.1 Curățarea și Normalizarea Datelor

Înainte de tokenizare, am aplicat următoarele transformări asupra textului brut:

- **Lowercasing:** Conversia întregului text la litere mici pentru a reduce dimensiunea vocabularului.
- **Eliminarea Zgomotului:** Folosind expresii regulate (RegEx), am eliminat cifrele, semnele de punctuație și caracterele speciale care nu contribuie la sentiment.
- **Eliminarea Stopwords:** Am utilizat biblioteca NLTK pentru a filtra cuvintele de legătură din limba română (ex: "și", "la", "care"), păstrând doar termenii cu încărcătură semantică.

6.2.2 Tokenizare și Construirea Vocabularului

Pentru tokenizare am utilizat biblioteca **Spacy** cu modelul de limbă `ro_core_news_sm`. Procesul a generat un vocabular de **49.659 de cuvinte unice**. Fiecărui cuvânt i săa asociat un index numeric, rezervând indexul 0 pentru *Padding* și indexul 1 pentru cuvintele necunoscute (*Unknown*).

6.2.3 Embedding Layer (FastText)

Pentru a surprinde relațiile semantice dintre cuvinte, am evitat antrenarea unui strat de embedding de la zero (care ar fi necesitat un corpus mult mai mare). Am utilizat **FastText pre-trained embeddings** (`cc.ro.300.bin`), care oferă vectori de dimensiune 300 pentru limba română.

6.2.4 Standardizarea Lungimii (Global Padding)

Deoarece recenziile au lungimi variabile (de la câteva cuvinte la paragrafe întregi), rețelele neurale necesită o dimensiune fixă a intrării pentru a putea procesa datele în batch-uri.

- Am calculat lungimea maximă a secvențelor luând în considerare simultan seturile de **Antrenare, Validare și Testare**.
- Această valoare maximă globală a fost impusă tuturor propozițiilor. Secvențele mai scurte au fost completate (padded) cu indexul 0 la final, iar cele (teoretic) mai lungi ar fi fost trunchiate.
- Această abordare garantează consistența dimensiunilor tensorilor ($BatchSize \times MaxLength$) pe tot parcursul experimentului, eliminând riscul erorilor la inferență pe date noi.

6.2.5 Matricea de Embedding

Pentru reprezentarea semantică a cuvintelor, am utilizat **FastText Pre-trained Embeddings** (`cc.ro.300.bin`):

- Am construit o matrice de dimensiune ($VocabSize \times 300$).
- Pentru fiecare cuvânt din vocabularul nostru, am copiat vectorul corespunzător din FastText.
- **Tratarea cuvintelor necunoscute (OOV):** Cuvintele care nu au fost găsite în FastText (ex: cuvinte care nu se regăseau în setul de antrenare) au fost inițializate cu un vector de valori aleatorii uniforme în intervalul $[-1, 1]$. Această abordare permite rețelei să învețe o reprezentare proprie pentru conceptul de "necunoscut" în timpul antrenării.

7 Antrenarea și Evaluarea Modelelor Recurente

După preprocessarea textului și generarea embedding-urilor folosind FastText, am implementat și comparat trei arhitecturi recurente pentru a determina cea mai eficientă metodă de captare a contextului semantic în limba română.

7.1 Setup Experimental

Toate modelele au fost antrenate utilizând aceiași hiperparametri pentru a asigura o comparație echitabilă:

- **Optimizator:** Adam ($lr = 3e - 4$, Weight Decay= $1e - 4$).
- **Loss Function:** CrossEntropyLoss cu `label_smoothing=0.1` pentru regularizare.
- **Scheduler:** ReduceLROnPlateau (factor 0.5, patience 4).
- **Arhitectură:** Embedding (300 dim) \rightarrow RNN/LSTM (Hidden Size 256) \rightarrow Dropout (0.3) \rightarrow Linear Output.

7.2 Arhitectura Simple RNN (Baseline)

Primul experiment a utilizat o rețea recurrentă simplă (Vanilla RNN).

Analiza Performanței:

- **Rezultat:** Acuratețe Test: **75.44%**.
- **Limitări:** Așa cum se observă în Figura 44, modelul converge, dar se plafonează rapid. Performanța inferioară este cauzată de problema dispariției gradienților (*vanishing gradient problem*), specifică RNN-urilor simple. Modelul uită informația de la începutul recenziei până ajunge la final, pierzând contextul necesar pentru texte lungi (distribuția bimodală observată la EDA).

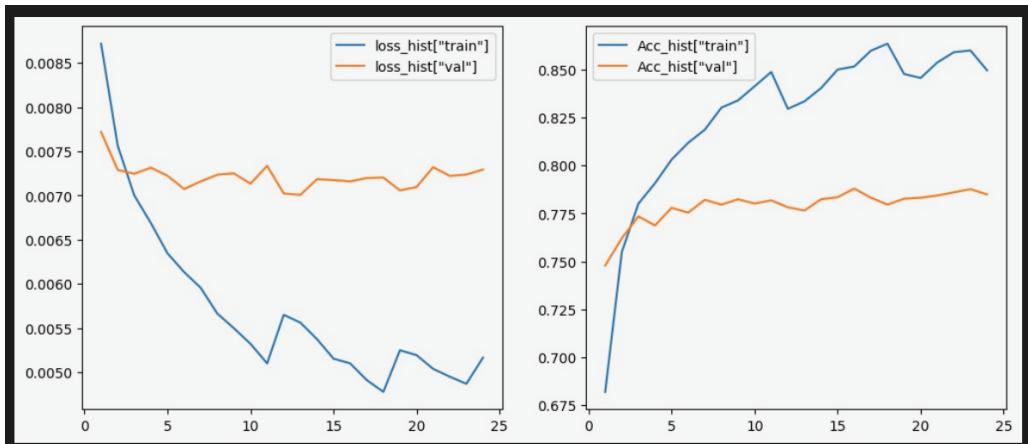


Figura 44: Curbele de antrenare pentru Simple RNN.

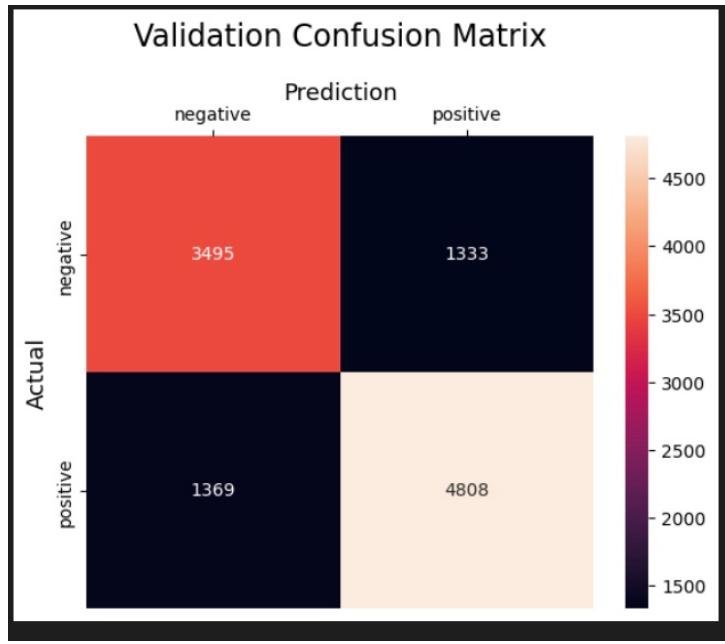


Figura 45: Matricea de Confuzie - Simple RNN (Test). Se observă un număr semnificativ de erori (1333 False Positives și 1369 False Negatives).

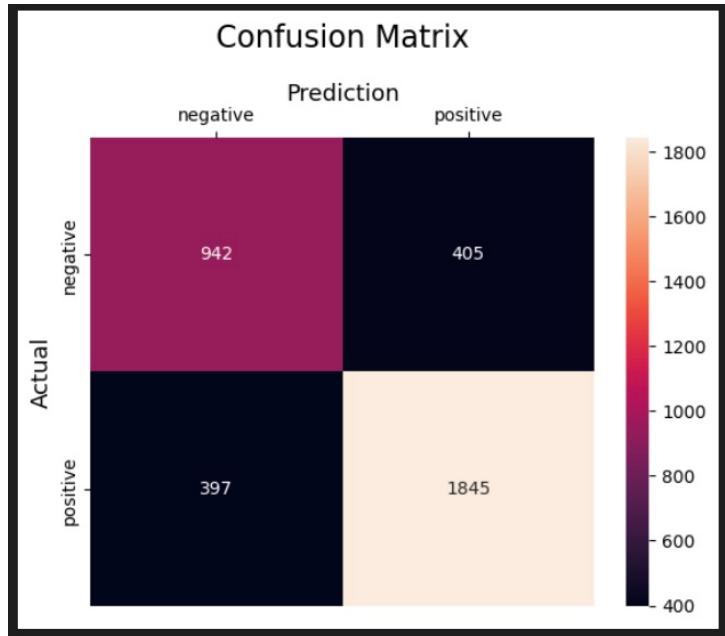


Figura 46: Matricea de Confuzie - Simple RNN (Validare).

7.3 Arhitecturi LSTM (Long Short-Term Memory)

Pentru a remedia deficiențele RNN-ului, am trecut la arhitectura LSTM, care utilizează porți (gates) pentru a controla fluxul de informație și a menține dependențe pe termen lung.

7.3.1 LSTM Bidirecțional vs. Unidirecțional

Am testat două variante: una care parcurge textul doar de la stânga la dreapta (Unidirectional) și una care îl parcurge în ambele sensuri simultan (Bidirectional).

Analiza Comparativă:

- **Salt de Performanță:** Ambele modele LSTM au depășit RNN-ul cu aproximativ **6-7%**, demonstrând capacitatea superioară de a înțelege fraze complexe și negații (ex: "Nu pot spune că mi-a plăcut" - unde "nu" schimbă sensul lui "plăcut").
- **Câștigătorul Surprinzător:** Contrar intuiției initiale, varianta **LSTM Unidirectional** a obținut cea mai bună acuratețe pe setul de test (**82.72%**) față de cea Bidirectională (**81.83%**).
- **Explicație Posibilă:** Deși varianta bidirectională are acces la mai mult context, ea are și dublul numărului de parametri. Pe un set de date de dimensiune medie (18k exemple), acest lucru poate duce la un ușor overfitting (modelul devine prea complex), în timp ce varianta unidirectională a generalizat mai bine.

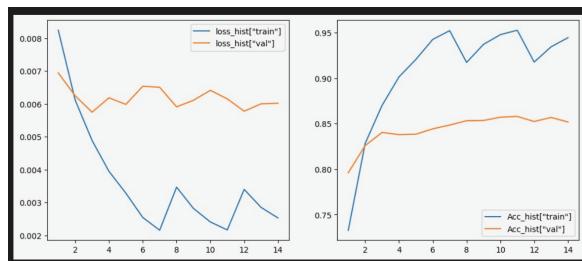


Figura 47: Curgea LSTM Bidirectional.

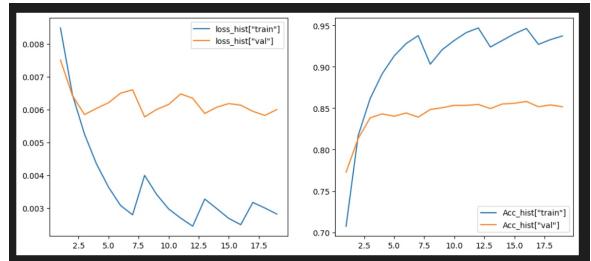


Figura 48: Curgea LSTM Unidirectional.

Analiza Matricei de Confuzie (LSTM Unidirectional - Fig. 52): Comparativ cu RNN, LSTM-ul a redus drastic erorile. Diagonala principală este mult mai pronunțată, iar echilibrul dintre Precision și Recall (ambele 0.82) indică un model robust, care nu favorizează o clasă în detrimentul celeilalte.

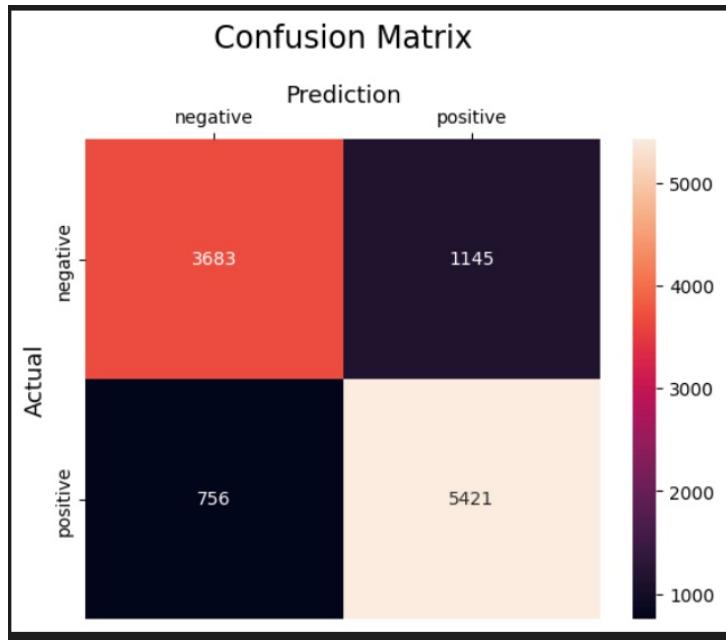


Figura 49: Matricea de Confuzie - LSTM Unidirecțional (Best Model). Se observă un număr mai mic de erori (1145 False Positives (față de 1333 la RNN) și 756 False Negatives (față de 1369 la RNN)).

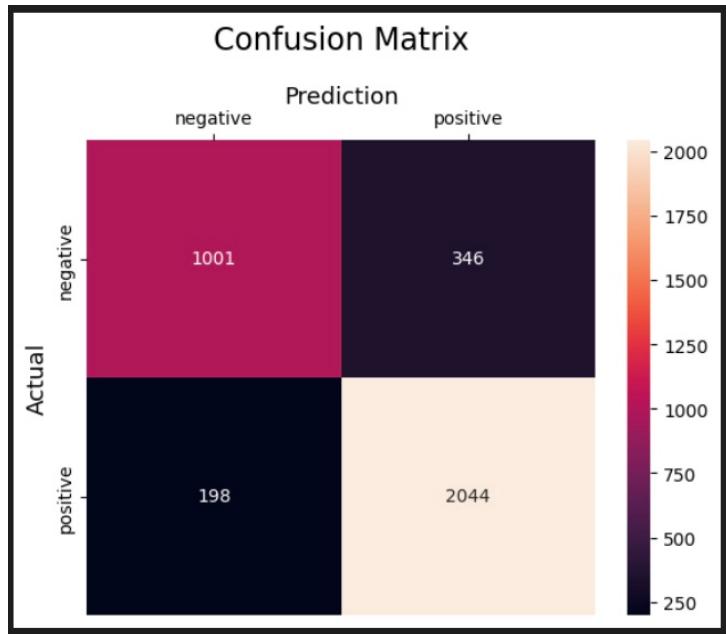


Figura 50: Matricea de Confuzie - LSTM Unidirecțional (Best Model) - pentru validare

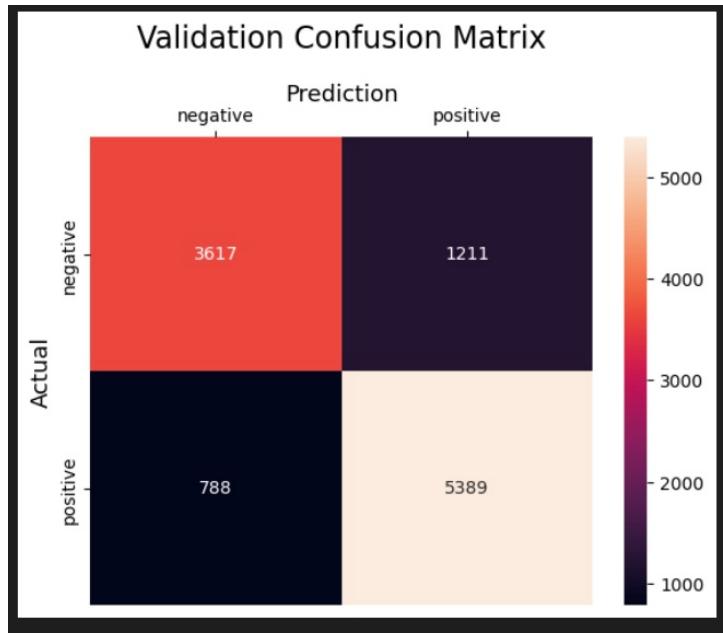


Figura 51: Matricea de Confuzie - LSTM Bidirecțional . Se observă și aici un număr mai mic de erori, chiar dacă mai multe ca la Unidirectionale (1211 False Positives (față de 1333 la RNN) și 788 False Negatives (față de 1369 la RNN)).

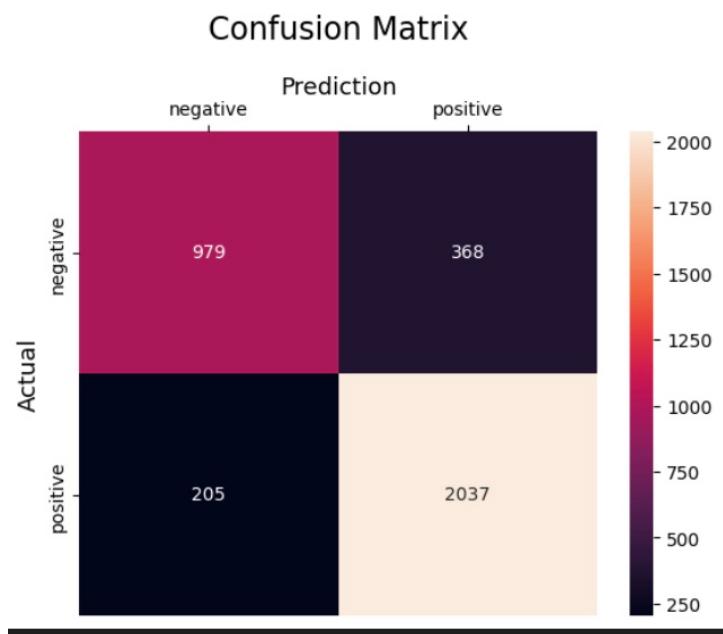


Figura 52: Matricea de Confuzie - LSTM Bidirecțional pentru validare.

7.4 Rezultate Centralizate - NLP

Tabelul de mai jos sintetizează performanța modelelor pe setul de testare.

Tabela 3: Compararea performanței modelelor de Analiză a Sentimentului

Model	Epoci	Configurație	Acc	F1-score	Precision	Recall
Simple RNN	50	0.7544	0.7509	0.7507	0.7511	
LSTM Bidirectional	50	0.8183	0.8135	0.8188	0.8108	
LSTM Unidirectional	50	0.8272	0.8228	0.8276	0.8202	

Concluzie Finală NLP Utilizarea embedding-urilor pre-antrenate (FastText) combinate cu o arhitectură LSTM a permis obținerea unei acurateți de peste 82% pe limba română. Diferența majoră față de RNN subliniază importanța memoriei pe termen lung în procesarea textului, unde sentimentul este adesea determinat de relația dintre cuvinte aflate la distanță în frază.