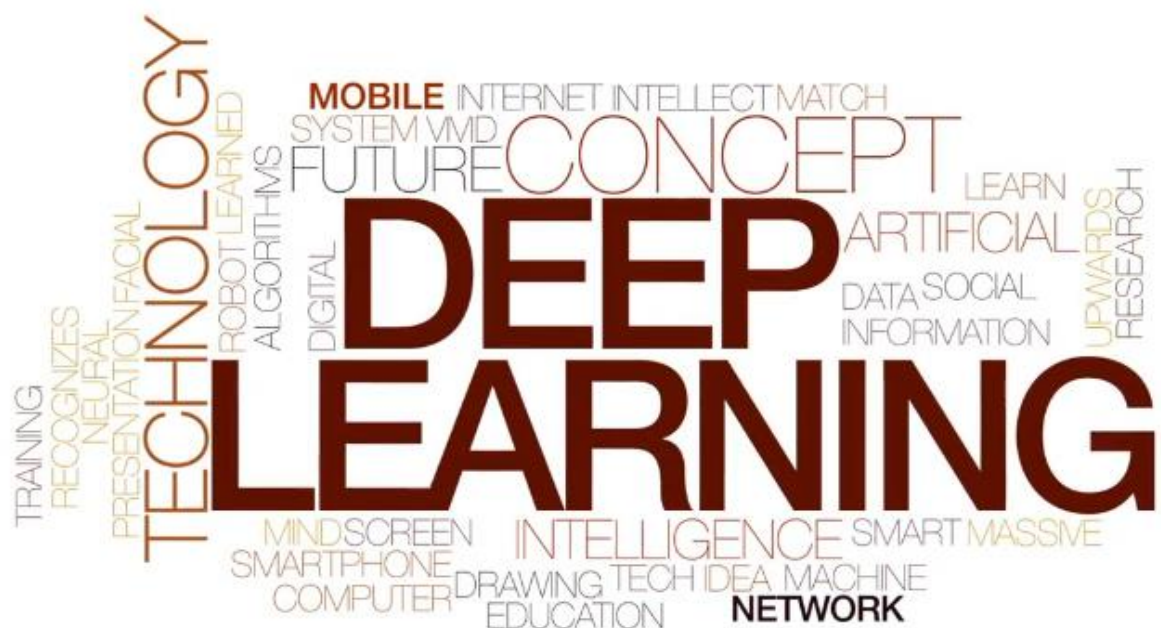




RAPORT DE CERCETARE NR. 2

Utilizarea rețelelor neuronale artificiale pentru clasificarea imaginilor digitale

Abstract

Segmentarea imaginilor aeriene este un proces care se bazează pur și simplu pe conținut spectral, iar acest lucru îl face un subiect foarte provocator.

În această lucrare, am folosit o rețea neuronală convoluțională (CNN) pentru a realiza segmentarea semantică a ortofotoplanurilor în șase clase, cum ar fi drumuri, clădiri, vegetație joasă, copaci, vehicule și altele. Pentru testarea procesului de segmentare folosim două seturi de date, ambele benchmark-uri ISPRS (Vaihingen și Potsdam). Arhitectura rețelei neuronale utilizate se bazează pe rețeaua convoluțională profundă SegNet, iar ponderile inițiale utilizate sunt cele obținute pe setul de date ImageNet aplicând arhitectura VGG16.

În Secțiunea 1, numită Introducere, definim noțiunile de rețea neuronală artificială, neuron natural și artificial și arhitecturile rețelei.

În Secțiunea 2, prezentăm metodele utilizate, începând cu rețeaua neuronală convoluțională și terminând cu arhitectura SegNet.

Secțiunea 3 este studiu de caz, unde definim seturile de date utilizate și analizăm rezultatele obținute.

În Secțiunea 4 sunt enumerate concluziile.

1. Introducere

Utilizăm rețele neuronale artificiale frecvent în zilele noastre, în aproape toate domeniile tehnologiei. Unele dintre cele mai cunoscute nume din tehnologia actuală, precum Google, Facebook, Microsoft, Amazon, investesc și folosesc inteligența artificială în propriile modele de afaceri. Unele aplicații cunoscute sunt asistenții virtuali (de ex. Alexa), recunoașterea vorbirii, recunoaștere facială, traducerea automată etc.

Rețelele neuronale profunde sunt utilizate cu succes în medicină, pentru analiza imaginii, designul medicamentelor, sisteme de diagnostic, de către producătorii de vehicule autonome etc. Aplicațiile de bază ale rețelelor neuronale artificiale, în fotogrammetrie și teledetecție, se referă la detectarea obiectelor (Fig. 1), segmentarea sau clasificarea imaginilor (Fig. 2) și detectarea schimbărilor în timp.



Fig. 1. Identificarea obiectelor soft Visual Profiler [1]

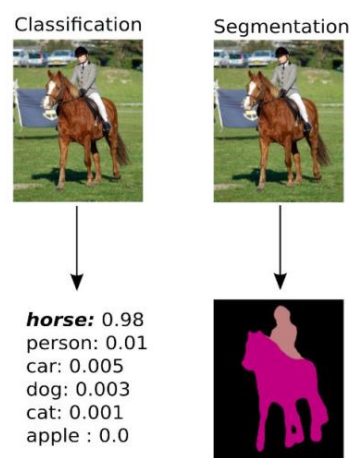


Fig. 2. Clasificarea și segmentarea imaginii [2]

Haykin definește rețeaua neuronală artificială ca fiind “procesor distribuit, masiv paralel, compus din unități simple de procesare cu abilitatea naturală de stocare a cunoștințelor noi pe baza experienței anterioare și a le face disponibile pentru utilizarea ulterioară” [3].

1.1.Principii de bază

Neuronul este o celulă excitabilă din punct de vedere electric care comunică cu alte celule prin conexiuni specializate numite sinapse, fiind unitatea morfo-funcțională a sistemului nervos. În corpul uman există 100 miliarde de neuroni, dispuși într-o rețea. Fiecare neuron are aproximativ 7000 de sinapse cu alți neuroni. Numărul de sinapse pentru un adult variază între 10^{14} și 5×10^{14} .

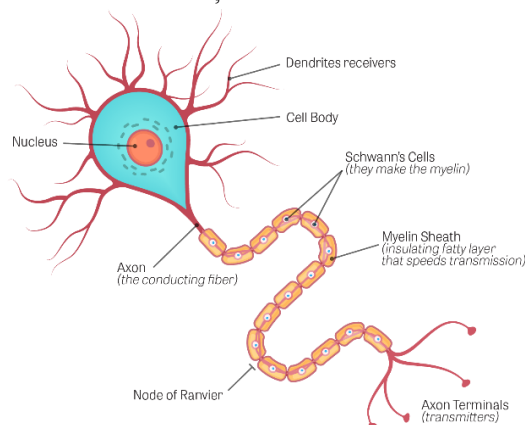


Fig. 3. Structură neuron uman (Sursă: iStock Photos, ©)

Componentele de bază ale unui neuron artificial sunt:

- un set de conexiuni/sinapse între unități; în general, fiecare conexiune este caracterizată de o pondere/putere;
- un sumator care adună semnalele de intrare, ponderate cu puterea sinaptică a neuronului respectiv;
- o funcție de activare, care are rolul de limitare a intervalului de amplitudine admisibil al semnalului de ieșire la o anumită valoare finită;
- un bias/polarizare aplicat din exteriorul rețelei, care are scopul de a crește sau micșora intrarea netă a funcției de activare, în funcție de valoarea acesteia (pozitivă sau negativă).

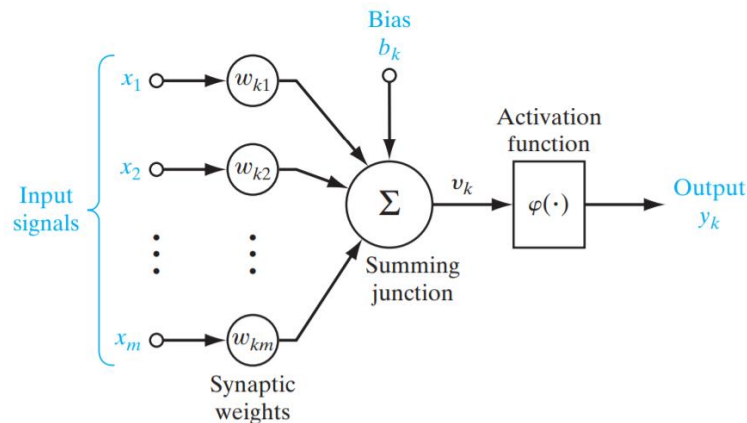


Fig. 4. Model nonlinear al neuronului k [3]

1.2. Arhitectura rețelelor neuronale artificiale

Modul în care sunt structurați neuronii unei rețele neuronale artificiale este legat de algoritmul de învățare folosit pentru antrenarea rețelei. În cele ce urmează, vă prezentăm trei arhitecturi sau structuri de rețea diferite.

a. Rețele feed-forward (uni-direcționale) unistrat

O rețea de tip feed-forward face ca stratul de intrare să se proiecteze direct pe un strat de ieșire de neuroni, dar nu invers. Noțiunea de unistrat se referă la calculul care se realizează într-un singur strat.

b. Rețele feed-forward (uni-direcționale) multistrat

O rețea de tip feed-forward multistrat are unul sau mai multe straturi ascunse, în care nodurile de calcul sunt numite unități ascunse. Adăugarea acestor straturi are ca scop obținerea unor rezultate îmbunătățite. Se poate afirma că rețeaua dobândește o perspectivă globală, datorită setului suplimentar de conexiuni sinaptice și dimensiunii suplimentare a interacțiunilor neuronale [4].

Rețelele feed-forward multistrat sunt mai lente, dar pot implementa funcții mai complexe decât cele unistrat.

c. Rețele recurente (feed-back)

O rețea neuronală recurentă prezintă minim o buclă de feed-back. Cea mai simplă arhitectură de tip recurent constă dintr-un unistrat de neuroni, unde fiecare neuron își transmite semnalul de ieșire înapoi intrărilor tuturor celorlalți neuroni.

Buclele de feed-back implică utilizarea pe aceste ramuri a unor elemente de timp sau de întârziere. Acestea determină un comportament neliniar-dinamic. Acest tip de arhitectură este cea mai complexă și în funcție de densitatea conexiunilor inverse există rețele total recurente (model Hopfield) și parțial recurente (model Elman sau Chua-Yang) [5].

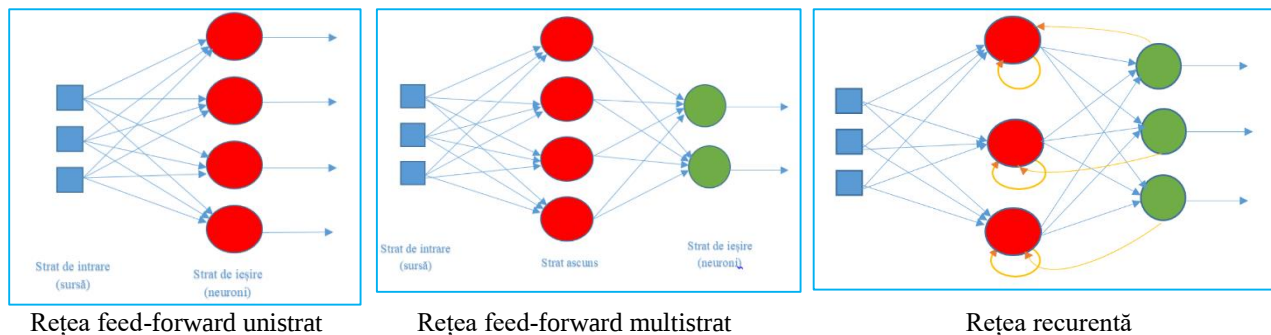


Fig. 5. Arhitecturi de rețele neuronale artificiale

2. Metode

2.1. Rețea neuronală convoluțională (CNN)

CNN este o rețea neuronală profundă, compusă din partea convoluțională, care extrage caracteristicile specifice ale datelor de intrare și le stochează sub formă de vector și unul sau mai multe straturi complet conectate de tip Perceptron multistrat, care realizează partea de clasificare [6].

Prima aplicație a rețelelor neuronale CNN este reprezentată de arhitectura LeNet-5, introdusă de Yann LeCun, Leon Bottou, Yoshua Bengio și Patrick Haffner, în 1998. Aceasta a fost dezvoltată pentru recunoașterea literelor de mână din documente (Fig. 6.) și se folosește cu succes în procesul de clasificare a imaginilor.

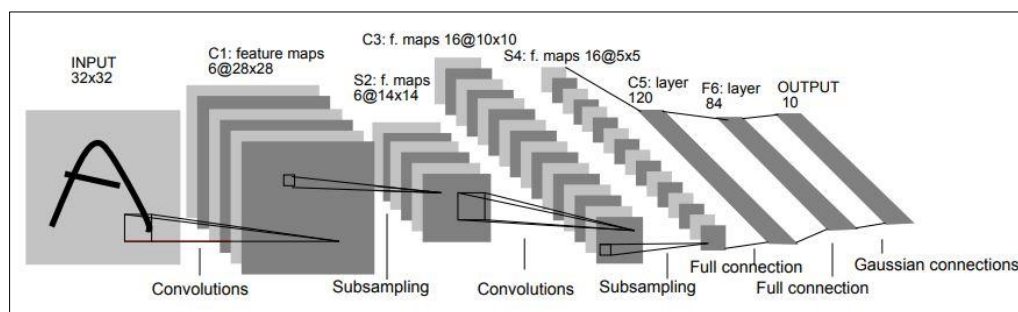


Fig. 6. Arhitectura LeNet-5 [7]

Pentru a descrie mai bine structura CNN trebuie să explicăm câteva noțiuni: stratul convoluțional, stratul de agregare și stratul final complet conectat.

Straturile convoluționale au la bază operația matematică prin care două funcții produc o a treia, care exprimă cum forma celei dintâi o modifică pe a doua. Scopul acestui strat este de a reduce dimensiunea imaginii de intrare. Imaginea va fi parcursă de la stânga la dreapta și de jos în sus, cu un filtru kernel (o matrice de numere) și se vor calcula valorile corespunzătoare din stratul de convoluție (multiplicarea valorilor din filtru cu valorile originale ale pixelilor din imagine și însumarea acestora). După parcurgerea întregii imagini, se obține harta de activare sau harta caracteristicilor.

Stratul de agregare (pooling) are rolul de a continua reducerea dimensiunii spațiale a imaginii, a timpului de procesare și de evidențiere a anumitor caracteristici. Hiper-parametrii folosiți sunt dimensiunea filtrului de agregare și pasul de glisare a acestuia peste matricea de intrare în stratul de agregare. Cele două tipuri de agregare folosite sunt max pooling și average pooling.

Stratul final complet conectat are rolul de a adapta ieșirea din partea convoluțională la un vector cu N dimensiuni, unde N este numărul de clase folosite în procesul de clasificare. Valorile din acest vector N reprezintă valoarea probabilității unei anumite clase.

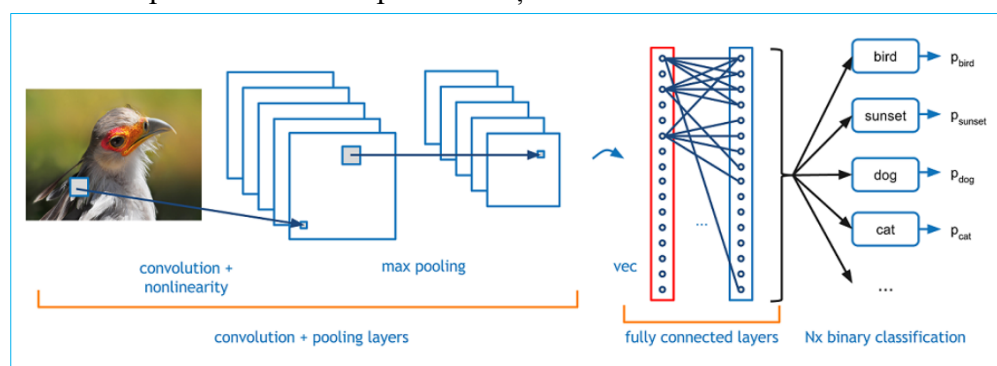


Fig. 7. Arhitectura CNN [8]

2.2.SegNet

SegNet este o rețea neuronală convoluțională profundă, dezvoltată în cadrul grupului Computer Vision și Robotica de la Universitatea Cambridge [9] și folosită pentru segmentarea semantică la nivel de pixel.

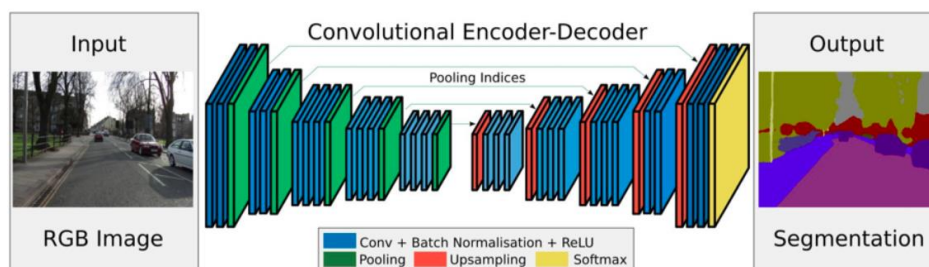


Fig. 8. Arhitectura SegNet [9]

Rețeaua este formată din straturi de codificatori, decodori și un strat final de clasificare la nivel de pixel.

Rețeaua de codificatori reprezentată în Fig. 9 este formată din 13 straturi convoluționale (corespunzătoare celor din rețeaua VGG16). Fiecărui strat codificator îi corespunde unul tip decodor. Dacă în faza de codificare se face o subeșantionare a datelor, în partea de decodare se aplică operația inversă și anume supraeșantionare, pentru ca imaginea finală să aibă aceeași dimensiune cu cea inițială.

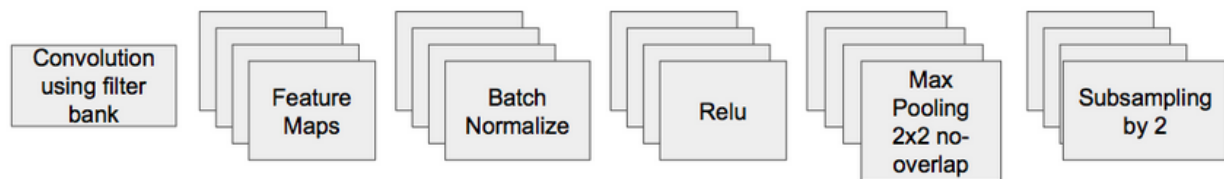


Fig. 9. Rețea codificatori [10]

Pentru a testa această arhitectură am folosit produsul demonstrativ, de pe site-ul <http://mi.eng.cam.ac.uk/projects/segnet/demo.php#demo> pe o imagine aleasă aleator din România, rezultatele obținute pentru cele 12 clase implicate ale SegNet pot fi observate în Fig. 10.

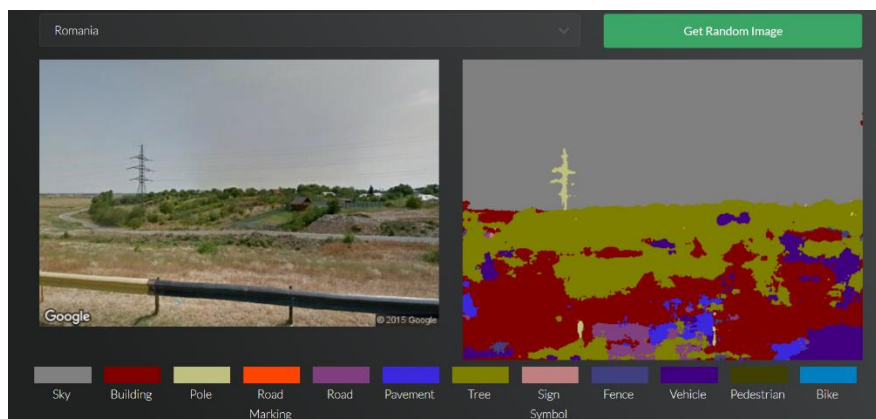


Fig. 10. Exemplu imagine clasificată folosind SegNet [11]

3. Studiu de caz

3.1. Seturi de date folosite

Primul set de date folosit este reprezentat de un subset al datelor achiziționate deasupra orașului Vaihingen an der Enz, Germania de către Asociația Germană de Fotogrammetrie și Teledetecție [12]. Imaginile au fost achiziționate cu camera Intergraph/ZI DMC, de compania RWE Power, în vara anului 2008 și au următoarele caracteristici:

- rezoluția pixelului la sol - 8 cm;
- rezoluție radimetrică - 11 biți;
- rezoluția spectrală - IR, R, G;

- număr imagini - 38.

Ortofotoplanurile generate sunt tip TOP (true-ortofotoplan) cu rezoluția spațială de 9 cm. Setul de date folosit este compus din true-ortofotoplanuri, imaginile clasificate (ground-truth) și imaginile clasificate cu contur erodat.

Clasele folosite pentru segmentare sunt DRUMURI, CLĂDIRI, VEGETAȚIE JOASĂ, POMI, VEHICULE și ALTELE.



Fig. 11. Exemplu o imagine din setul de date Vaihingen și corespondentele ei

Al doilea set de date folosit este reprezentat de un subset al datelor achiziționate deasupra orașului Potsdam, Germania de către *BSF swissphoto* [13], unde imaginile au următoarele caracteristici:

- rezoluția pixelului la sol - 5 cm;
- rezoluție radimetrică - 8 biți;
- rezoluția spectrală - IR, R, G / R, G, B / R,G, B, IR;
- număr imagini - 38.

Ortofotoplanurile generate sunt tip TOP (true-ortofotoplan), cu rezoluția spațială de 5 cm. Clasele folosite pentru segmentare sunt aceleași ca la setul de date Vaihingen.

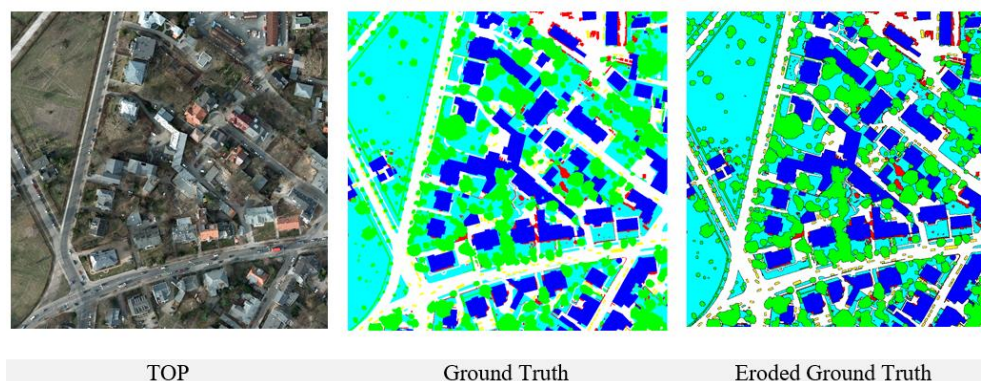


Fig. 12. Exemplu o imagine din setul de date Potsdam și corespondentele ei

3.2.Arhitectura rețelei neuronale utilizate

Pentru testarea segmentării imaginilor utilizând rețele neuronale artificiale, am folosit codul Python, disponibil pe pagina <https://blesaux.github.io/teaching/DL4RS> [49]. Rularea

codului se face în mediul Google Drive. Codul reprezintă o implementare a bibliotecii Pytorch pentru segmentarea imaginilor aeriene pe baza FCN, folosind arhitectura SegNet [15].

Inițial în cod sunt setați parametrii (diferiți în funcție de fiecare set de date), cum ar fi dimensiune fereastră, pas, clasele folosite la segmentare, setul de date folosit etc. Pentru manipularea setului de date se folosește scikit-image. Dacă imaginea Ground-Truth este stocată în format RGB trebuie să definim paleta de culori care poate face maparea ID-ului clasei la culoarea din paleta RGB. În cadrul codului trebuie definite funcțiile ajutoare, care vor fi apelate în continuare (ex: definire matricei de confuzie, scorului F1, coeficientului kappa). Apoi, se încarcă toate imaginile în memorie și se aplică o eșantionare aleatoare. În cadrul acestui pas se face și augumentarea și normalizarea datelor. După acest pas se definește arhitectura SegNet. Se inițializează rețeaua și ponderile folosind abordarea He. Opțional, se pot încărca ponderile pre-antrenate pe setul ImageNet cu arhitectura VGG16. Se încarcă rețeaua pe GPU, pentru facilitarea procesului de antrenare și testare și se divizează datele în set învățare și set testare. În procesul de optimizare se folosește algoritmul Stochastic Gradient Descent pentru optimizarea ponderilor și se antrenează codificatorul la jumătate din rata de învățare a decodului. Pasul următor se referă la antrenarea rețelei care se poate realiza în una sau mai multe epoci. După ce procesul de antrenare s-a încheiat, se încarcă ponderile finale și se testează rețeaua folosind un pas rezonabil, de ex. jumătate sau un sfert din dimensiunea ferestrei. Rezultatele segmentării sunt compuse din imaginile clasificate și indicatorii de acuratețe.

3.3.Rezultate

Pentru testarea și compararea rezultatelor, am folosit diferite subseturi de date din benchmark-urile amintite: două încercări pentru Vaihingen și una pentru Potsdam. Pentru aceste trei seturi de date, am rulat codul folosind numere diferite de epoci.

Test 1 - Vaihingen (2 imagini pentru antrenare și una pentru test)

Tile-urile folosite pentru învățare sunt top_mosaic_09cm_area1.tif și top_mosaic_09cm_area23.tif, iar pentru testare top_mosaic_09cm_area5.tif.

Matricea de confuzie și valorile indicatorilor de acuratețe obținuți în urma procesului de antrenare sunt prezentate în Tabelul 1 și Tabelul 2. Este important de specificat că a fost rulată o singură epocă.

Tabel 1. Matricea de confuzie rezultată în procesul de antrenare

CLASE	DRUMURI	CLĂDIRI	VEGETAȚIE JOASĂ	POMI	VEHICULE
DRUMURI	1128357	323097	56151	7899	293
CLĂDIRI	192253	2085525	19065	3860	1847
VEGETAȚIE JOASĂ	17399	3690	112131	130589	0
POMI	5691	685	9178	277082	0
VEHICULE	32731	8498	2057	529	1131

Tabel 2. Indicatori de acuratețe în procesul de antrenare

CLASE	SCOR F1 (%)	OA (%)	KAPPA (%)
DRUMURI	78.02	81.55	69.18
CLĂDIRI	88.29		
VEGETAȚIE JOASĂ	48.50		
POMI	77.76		
VEHICULE	4.69		
ALTELE	-		

După ce antrenarea rețelei s-a încheiat, se încarcă ponderile finale și se testează rețeaua folosind un pas rezonabil. Valorile indicatorilor de acuratețe rezultați în urma procesului de testare sunt detaliați în Tabelul 3.

Tabel 3. Indicatori de acuratețe în procesul de testare

CLASE	SCOR F1 (%)	OA (%)	KAPPA (%)
DRUMURI	78.90	82.14	70.14
CLĂDIRI	88.62		
VEGETAȚIE JOASĂ	49.39		
POMI	78.13		
VEHICULE	3.83		
ALTELE	-		

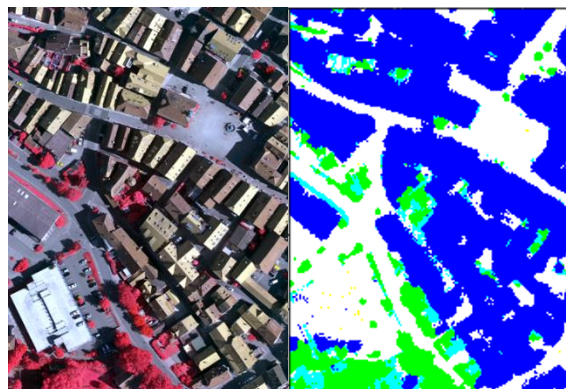


Fig. 13. Vizualizare true-ortofotoplan *top_mosaic_09cm_area5.tif* și rezultatul clasificării (o epocă)

A fost rulat întregul proces și pentru 2 respectiv 10 epoci, iar rezultatele sunt afișate în Tabelul 5. Se poate observa o îmbunătățire a indicatorilor de acuratețe direct proporțională cu numărul de epoci folosite în procesul de antrenare. În Tabelul 5 este detaliat scorul F1 obținut pentru clasa CLĂDIRI, acestea au fost segmentate optim, cu obținerea unui scor F1 de 91.76 %, pentru 10 epoci.

Tabel 4. Indicatori de acuratețe funcție de numărul de epoci Test 1

Nr epoci	OA (%)	SCOR F1 (%) - CLĂDIRI	KAPPA (%)	Timp necesar în procesul de antrenare
1	82.14	88.61	70.13	10 min
2	85.98	91.06	76.55	15 min
10	87.30	91.76	79.19	60 min

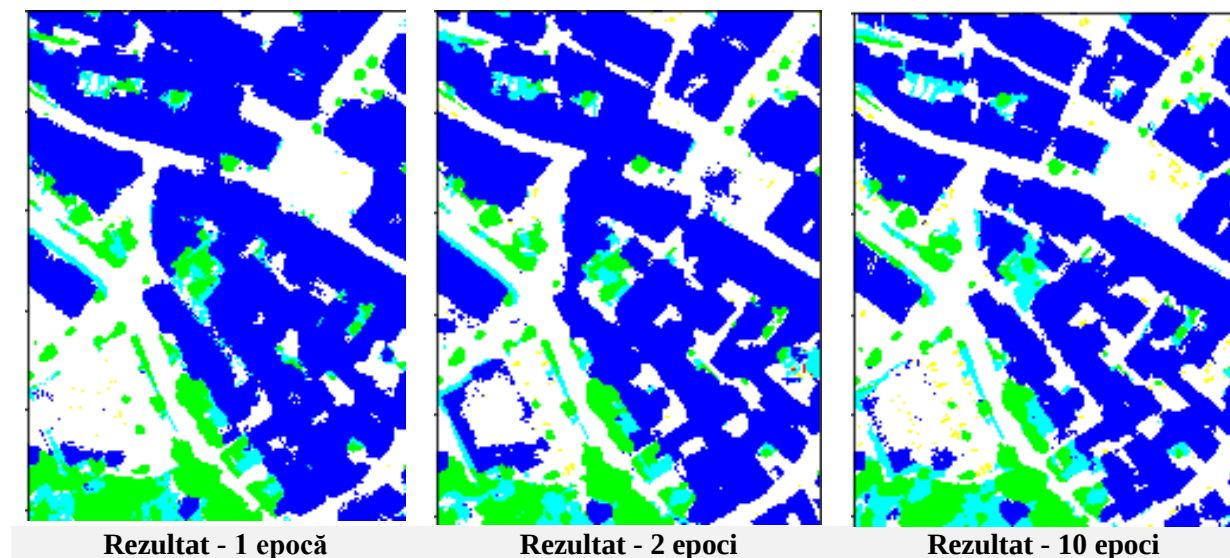


Fig. 14. Vizualizare rezultate clasificare Test 1

Test 2 - Vaihingen (4 imagini pentru antrenare și două pentru test)

Tile-urile folosite pentru învățare sunt top_mosaic_09cm_area5.tif, top_mosaic_09cm_area21.tif, top_mosaic_09cm_area23.tif și top_mosaic_09cm_area28.tif, iar pentru testare top_mosaic_09cm_area7.tif și top_mosaic_09cm_area30.tif.

Tabel 5. Indicatori de acuratețe funcție de numărul de epoci pentru Testul 2

CLASE	2 EPOCI			10 EPOCI			20 EPOCI		
	SCOR F1 (%)	OA (%)	KAPPA (%)	SCOR F1 (%)	OA (%)	KAPPA (%)	SCOR F1 (%)	OA (%)	KAPPA (%)
DRUMURI	82.21	78.09	70.74	87.83	86.20	81.63	87.53	85.76	81.03
CLĂDIRI	84.32			91.83			91.27		
VEGETAȚIE	58.43			75.37			74.64		
JOASĂ	81.12			87.10			86.71		
POMI	4.93			72.96			71.75		
VEHICULE	-			-			-		
ALTELE	-			-			-		

Valorile obținute pentru indicatorii de acuratețe urmează un trend ascendent până la folosirea unui număr de aproximativ 12 epoci, apoi trendul devine descendent (Fig. 15). Putem concluziona că pentru subsetul de date folosit rezultatele optime s-au obținut antrenând rețeaua cu 10 epoci.

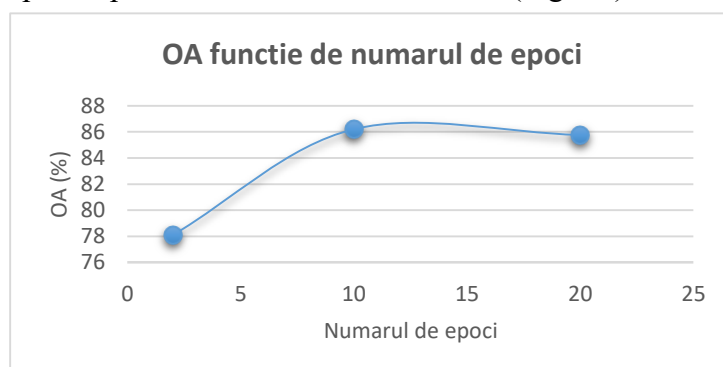


Fig. 15. Grafic OA funcție de numărul de epoci

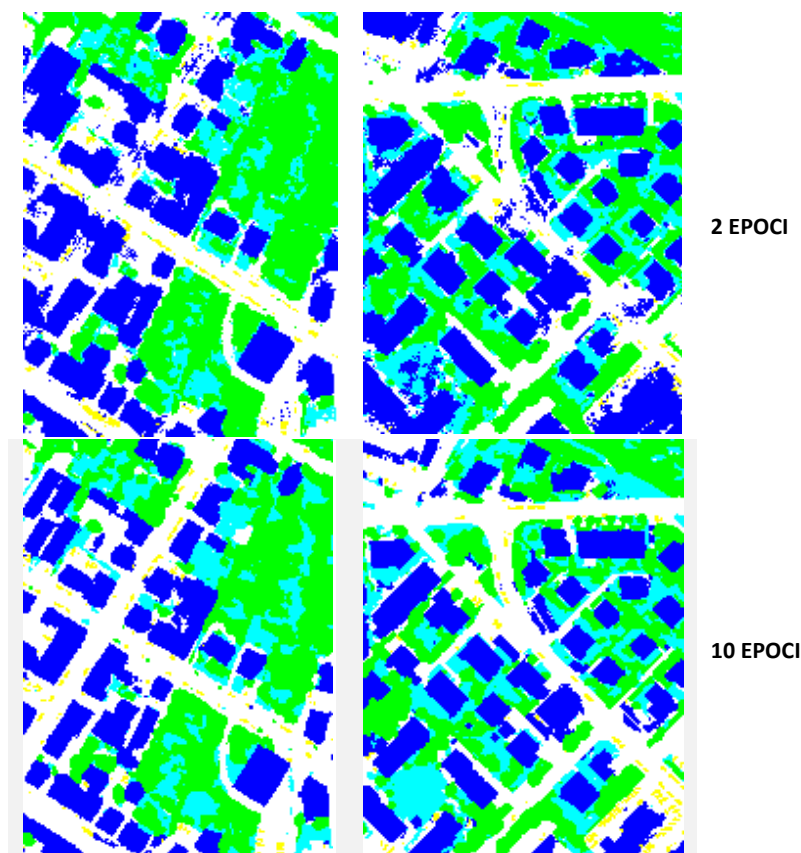


Fig. 16. Rezultatele comparative obținute pentru segmentarea celor două imagini Test 2

Test 3 - Potsdam (3 imagini pentru antrenare și una pentru test)

Tile-urile folosite pentru învățare sunt top_potsdam_2_10_RGB.tif, top_potsdam_2_11_RGB.tif și top_potsdam_2_13_RGB.tif, iar pentru testare top_potsdam_2_14_RGB.tif.

Matricea de confuzie și valorile indicatorilor de acuratețe în procesul de antrenare, folosind 5 epoci, sunt prezentate în Tabelul 6.

Tabel 6. Matricea de confuzie rezultată în procesul de antrenare

CLASE	DRUMURI	CLĂDIRI	VEGETAȚIE JOASĂ	POMI	VEHICULE	ALTELE
DRUMURI	3705817	663047	501458	141697	2591	10234
CLĂDIRI	90835	1711595	169894	41510	8754	331776
VEGETAȚIE JOASĂ	1234399	46820	11660323	1074105	1559	5260
POMI	737909	245355	3751823	7416668	12769	27179
VEHICULE	13377	44144	7029	1512	85334	9178
ALTELE	7661	35768	22244	4284	11085	5322

În urma analizei valorilor din matricea de confuzie se observă următoarele probleme:

- clasa ALTELE generează cele mai mari confuzii în procesul de învățare;
- se crează unele confuzii semnificative între clasele DRUMURI și VEGETAȚIE JOASĂ;

- clasa VEHICULE nu este foarte bine diferențiată de celelalte clase.

Antrenare rețelei este un procesul de lungă durată; în cazul subsetului folosit și în funcție de numărul de epoci, timpul necesar a variat între o oră și 24 ore.

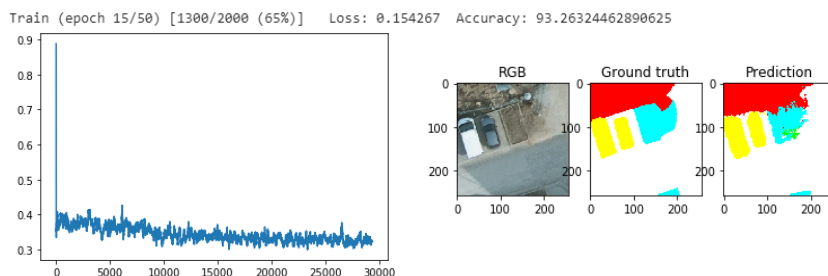


Fig. 17. Exemplu din timpul antrenării rețelei

În Fig. 17 se poate observa acuratețea on-the-fly de 93.26% obținută pentru antrenarea rețelei pe patch-ul din imagine, în cadrul epocii numărul 15, din totalul de 50 de epoci.

Ca parte a algoritmului de optimizare, eroarea pentru starea curentă a modelului trebuie estimată în mod repetat. Aceasta necesită alegerea unei funcții de eroare, denumită convențional funcție de pierdere (loss), care poate fi utilizată pentru a estima pierderea în cadrul modelului, astfel încât ponderile să poată fi actualizate pentru a reduce pierderea la următoarea evaluare. Funcția pe care dorim să o minimizăm sau să o maximizăm se numește funcție sau criteriu obiectiv. Când se dorește minimizarea acesteia se definește funcție de cost sau de pierdere [16].

Tabel 7. Scorul F1 pe clase funcție de numărul de epoci pentru Testul 3

CLASE	F1 SCORE (%)				
	5 EPOCI	10 EPOCI	20 EPOCI	30 EPOCI	50 EPOCI
DRUMURI	71.14	82.26	89.08	88.68	86.33
CLĂDIRI	70.75	77.06	94.22	95.06	95.52
VEGETAȚIE JOASĂ	77.95	81.66	88.42	88.07	86.41
POMI	71.89	79.81	87.66	85.91	85.16
VEHICULE	61.04	71.30	87.90	86.66	89.09
ALTELE	2.30	0.58	14.79	18.65	22.14

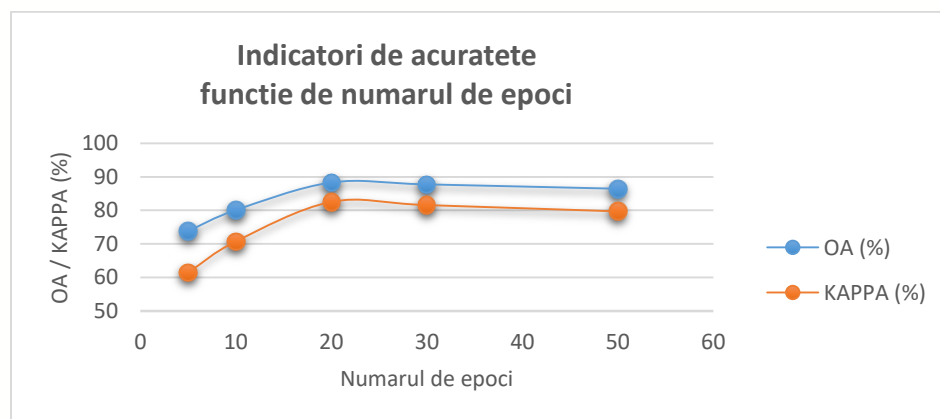


Fig. 18. Grafic indicatori de acuratețe funcție de numărul de epoci Test 3

Din Fig. 17 și Fig. 18 se poate observa o scădere a acurateții segmentării după 20 de epoci, dar o îmbunătățire a scorului F1 pentru clasa CLĂDIRI. Pentru clădiri putem afirma că cele mai bune rezultate s-au obținut pentru 50 de epoci. Timpul necesar pentru rularea procesului de antrenare pentru 50 de epoci a fost de 6 ore.

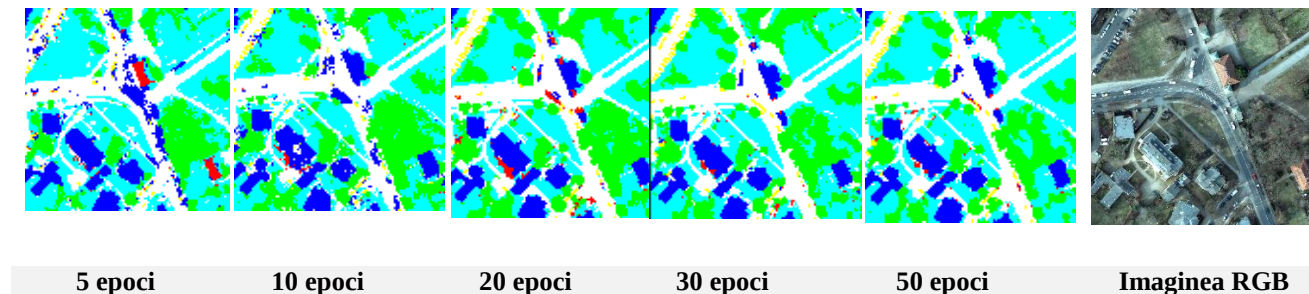


Fig. 19. Vizualizarea rezultatelor clasificării (imagine detaliu)

4. Concluzii

În zilele noastre, arhitectura AlexNet [17] a fost depășită de arhitecturi mult mai eficiente, dar reprezintă pasul cheie spre rețelele profunde actuale.

Rețelele neuronale profunde sunt utilizate cu precădere în prezent, pentru că au fost întrunite cele trei criterii de bază: seturile de date sunt mari (big data), capacitățile hardware date de GPU sunt impresionante și componenta software prezintă tehnici îmbunătățite și modele noi.

Alegerea modelului arhitectural și antrenarea rețelei se bazează pe încercări și experiența proprie dobândită. Din toate etapele unui proces complex folosind rețele neuronale, procesul de antrenare are durata cea mai lungă.

Unul dintre avantajele de bază ale rețelelor neuronale este capacitatea de a modela atât funcții liniare cât și neliniare.

Rețelele neuronale au capacitatea de a funcționa datorită numărului mare de neuroni chiar dacă sistemul prezintă deteriorări.

Rețelele neuronale convoluționale au devenit un standard în aplicațiile bazate pe analiza de imagini, performanțele lor putând egala performanța umană în domeniu. Este bine ca straturile convoluționale să utilizeze filtre de dimensiuni mici (3×3 sau 5×5), pas egal cu 1 și padding cu zero astfel încât să nu fie alterate dimensiunile spațiale ale intrării.

Valorile din matricea de confuzie rezultată în procesul de antrenare oferă informații valoroase cu privire la neclaritățile obținute în diferențierea claselor.

Pentru Testul 1 - Vaihingen, clădirile au fost segmentate cel mai precis după 10 epoci de antrenare a rețelei, cu obținerea unui scor F1 de 91.76%.

Pentru Testul 2 - Vaihingen, elementele au fost segmentate cel mai precis după 10 epoci de antrenare a rețelei, pentru clasa CLĂDIRI s-a obținut scorul F1 de 91.83%, pentru clasa DRUMURI s-a obținut scorul F1 de 87.83%, pentru clasa POMI s-a obținut scorul F1 de 87.1%. Din graficul reprezentat în Fig. 52. se poate deduce că rezultatele cele mai bune se pot obține pentru 12 epoci; după depășirea acestui număr de epoci precizia segmentării scade.

Pentru Test 3 - Potsdam, clasa ALTELE generează cele mai mari confuzii în procesul de învățare, din acest motiv scorul F1 obținut în cazul acestei clase variază între 2.3% și 22.14%, în funcție de numărul de epoci. Pentru acest set de date se observă o scădere a acurateții segmentării după 20 de epoci, dar o îmbunătățire a scorului F1 pentru clasa CLĂDIRI. Astfel, pentru clădiri cele mai bune rezultate s-au obținut pentru 50 de epoci (95.52%).

Mulțumiri

Setul de date Vaihingen a fost pus la dispoziție de către Societatea Germană de Fotogrammetrie, Teledetecție și Geoinformații (DGPF). Setul de date Potsdam a fost pus la dispoziție de către BSF Swissphoto. Pentru testarea segmentării imaginilor utilizând rețele neuronale artificiale, am folosit codul Python, disponibil pe pagina <https://blesaux.github.io/teaching/DL4RS>.

Bibliografie

1. <https://www.video-inform.com/visual-profiler/>, Vizitat: 15.02.2020
2. Lefèvre, S., Le Saux, B., Landrieu, L.: DEEP LEARNING FOR REMOTE SENSING - EDUSERV COURSE, EuroSDR, 2019
3. Haykin, S.: NEURAL NETWORKS AND LEARNING MACHINES, Third Edition, Pearson, ISBN-13: 978-0-13-147139-9, ISBN-10: 0-13-147139-2, 2008
4. Churchland, P. S. & Sejnowski, T. J.: THE COMPUTATIONAL BRAIN. Computational neuroscience. The MIT Press, 1992
5. http://staff.fmi.uvt.ro/~daniela.zaharie/cne2014/curs/cne2014_slides5.pdf, Vizitat: 14.01.2020
6. Vrejoiu, M. H.: REȚELE NEURONALE CONVOLUȚIONALE, BIG DATA ȘI DEEP LEARNING ÎN ANALIZA AUTOMATĂ DE IMAGINI, Romanian Journal of Information Technology and Automatic Control, Vol. 29, No. 1, pp. 91-114, 2019
7. LeCun, Y., Bottun, L., Bengio, Y., Haffner, P.: GRADIENT-BASED LEARNING APPLIED TO DOCUMENT RECOGNITION, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278-2324, 1998
8. <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>, Vizitat: 20.01.2020
9. Badrinarayanan, V., Kendall, A., Cipolla, R.: SEGNET: A DEEP CONVOLUTIONAL ENCODER-DECODER ARCHITECTURE FOR IMAGE SEGMENTATION, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 12, pp. 2481-2495, doi: 10.1109/TPAMI.2016.2644615, 2017
10. <https://towardsdatascience.com/summary-of-segnet-a-deep-convolutional-encoder-decoder-architecture-for-image-segmentation-75b2805d86f5>, Vizitat: 09.02.2020

11. <http://mi.eng.cam.ac.uk/projects/segnet/demo.php#demo>, Visited 09.02.2020
12. Cramer, M.: THE DGPF TEST ON DIGITAL AERIAL CAMERA EVALUATION – OVERVIEW AND TEST DESIGN, Photogrammetrie – Fernerkundung – Geoinformation 2 (2010), pp. 73-82, 2010
13. <http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html>, Vizitat: 16.02.2020
14. <https://blesaux.github.io/teaching/DL4RS>, Vizitat: 25.02.2020
15. Audebert, N., Le Saux, B., Lefèvre, S.: BEYOND RGB: VERY HIGH RESOLUTION URBAN REMOTE SENSING WITH MULTIMODAL DEEP NETWORKS, ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier, Vol. 140, pp.20-32, 2018
16. Goodfellow, I., Bengio, Y., Courville, A.: DEEP LEARNING (ADAPTIVE COMPUTATION AND MACHINE LEARNING SERIES), MIT Press, 2017
17. Krizhevsky, A., Sutskever, I. & Hinton, G. E.: IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS, Advances in neural information processing systems, Vol. 25(2), DOI: 10.1145/3065386, 2012