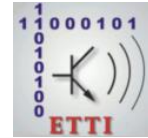




**UNIVERSITATEA POLITEHNICA  
DIN BUCUREȘTI**



**Școala Doctorală de Electronică, Telecomunicații  
și Tehnologia Informației**

Decizie nr. \_\_\_\_ din \_\_-\_\_-\_\_

# **REZUMAT TEZĂ DE DOCTORAT**

**Ing. Paul Liviu Aurel Diaconescu**

---

**CONTRIBUȚII LA OPTIMIZAREA ȘI  
APLICAȚIILE MODELULUI DEEP  
CONVOLUTIONAL NEURAL NETWORKS**

**CONTRIBUTIONS TO THE OPTIMIZATION  
AND APPLICATIONS OF DEEP  
CONVOLUTIONAL NEURAL NETWORK MODEL**

---

## **COMISIA DE DOCTORAT**

<b>Prof. Dr. Ing. Mihai CIUC</b> Univ. Politehnica din București	Președinte
<b>Prof. Dr. Ing. Victor NEAGOE</b> Univ. Politehnica din București	Conducător de doctorat
<b>Prof. Dr. Ing. Alexandru ISAR</b> Univ. Politehnica Timișoara	Referent
<b>Prof. Dr. Ing. Alexandru SERBANESCU</b> Academia Tehnică Militară „Ferdinand I”	Referent
<b>Conf. Dr. Ing. Anamaria RĂDOI</b> Univ. Politehnica din București	Referent

**BUCUREȘTI 2021**

# Cuprins

Lista tabelelor

Lista figurilor

<b>1. Introducere.....</b>	<b>1</b>
1.1. Prezentarea domeniului tezei de doctorat.....	1
1.2. Scopul tezei de doctorat.....	1
1.3. Conținutul tezei de doctorat.....	1
<b>2. Inteligența Artificială și Deep Learning.....</b>	<b>2</b>
2.1. Deep Learning. Concepte esențiale.....	2
2.1.1. Rețele Neurale Convoluționale (Convolutional Neural Networks CNN).....	2
2.1.2. Stratul convoluțional.....	2
2.1.3. Straturi de activare (Relu, Tahn, Sinusoidal).....	2
2.1.4. Stratul de punere în comun (Pooling).....	2
2.1.5. Stratul DropOut.....	2
2.2. Antrenarea unei DCNN.....	3
2.2.1. Back-propagation.....	3
2.2.2. Gradient Descent și rata de învățare. Oprește anticipată.....	3
2.2.3. Caracteristicile algoritmilor de antrenare.....	3
2.2.4. Supra-adaptare (Overfitting).....	3
2.2.5. Validarea antrenamentului (Training Validation).....	3
2.2.6. Eroare de antrenare (Training Loss).....	3
2.2.7. Metode de accelerare a antrenării.....	3
2.3. Aplicații CNN.....	3
2.3.1. Clasificarea.....	3
2.3.2. Transferul învățării.....	3
2.3.3. Detecția obiectelor.....	3
2.4. Selecția unor arhitecturi DCNN consacrate.....	3
2.5. Alte tipuri de rețele Deep Learning.....	3
2.6. Frameworks / Cadre de dezvoltare.....	3
2.6.1. Pytorch.....	3
2.6.2. Matlab.....	3
2.6.3. Github.....	3
2.7. Optimizarea performanței sistemului hardware.....	3
<b>3. Un sistem DCNN pentru detecția stării de ebrietate folosind analiza     imaginilor faciale termice în spectrul infraroșu.....</b>	<b>4</b>
3.1. Introducere.....	4
3.2. Modelul propus pentru detecția stării de ebrietate utilizând o arhitectură duală DCNN.....	4
3.3. Experimente.....	5
3.3.1. Setul de date cu imagini termice în spectrul infraroșu.....	5
3.3.2. Pregătirea parametrilor și antrenarea rețelei.....	6
3.4. Rezultate.....	6
3.5. Concluzii.....	6

<b>4. O arhitectură DCNN configurabila pentru clasificarea imaginilor hiperspectrale.....</b>	<b>7</b>
4.1. Introducere.....	7
4.2. Metoda Propusă.....	7
4.3. Rezultate experimentale.....	7
4.3.1. Setul de date.....	8
4.4. Rezultate.....	8
4.5. Concluzii.....	8
<b>5. Clasificarea imaginilor hiperspectrale folosind augmentarea elementelor de antrenare cu sisteme GAN.....</b>	<b>10</b>
5.1. Introducere.....	10
5.1.1. Sisteme GAN.....	10
5.1.2. Clasificarea imaginilor hiperspectrale.....	10
5.2. Metoda propusă.....	11
5.3. Rezultate experimentale.....	12
5.4. Concluzii.....	12
<b>6. Clasificarea solicitărilor de credit utilizând optimizarea modelului Deep Learning.....</b>	<b>13</b>
6.1. Introducere.....	13
6.1.1. Riscul de credit financiar.....	13
6.1.2. Soluții bazate pe modelul DCNN pentru predicția riscurilor asociate creditelor .....	13
6.2. Metoda Propusă.....	13
6.2.1. Mediul de simulare.....	13
6.2.2. Criterii de evaluare și costul creditului.....	13
6.2.3. Algoritmi de optimizare.....	13
6.2.4. Arhitectura DCNN.....	14
6.3. Experimente și rezultate.....	15
6.3.1. Setul de date.....	15
6.3.2. Performanțe și hiper-parametri optimi.....	15
6.3.3. Rezultate experimentale.....	15
6.4. Concluzii.....	15
<b>7. Tehnici de tip DCNN folosite pentru conducerea automată a vehiculelor în zone urbane.....</b>	<b>17</b>
7.1. Introducere.....	17
7.2. Transferul învățării.....	17
7.3. Sistemul DCNN YOLOv5.....	17
7.4. Metoda Propusă.....	17
7.4.1. Metoda propusă pentru transferul învățării.....	17
7.4.2. Criteriile de evaluare.....	17
7.4.3. Optimizarea parametrilor cu un algoritm genetic.....	17
7.4.4. Arhitectura software.....	18
7.5. Experimente și rezultate.....	18
7.5.1. Seturile de date.....	18
7.5.2. Transformarea atributelor.....	18
7.5.3. Rezultate experimentale.....	18

7.6. Concluzii.....	18
<b>8. Concluzii.....</b>	<b>20</b>
8.1. Rezultate obținute.....	20
8.1.1. Rezultate și concluzii referitoare la folosirea unui ansamblu DCNN pentru detecția stării de ebrietate folosind analiza imaginilor faciale termice în domeniul spectral infraroșu.....	20
8.1.2. Rezultate și concluzii referitoare la folosirea unei arhitecturi configurabile DCNN pentru clasificarea imaginilor hiperspectrale.....	20
8.1.3. Rezultate și concluzii referitoare la clasificarea imaginilor hiperspectrale folosind augmentarea elementelor de antrenare cu sisteme GAN.....	20
8.1.4. Rezultate și concluzii referitoare la clasificarea solicitărilor de credit utilizând optimizarea modelului Deep Learning.....	20
8.1.5. Rezultate și concluzii referitoare la tehnici de tip DCNN folosite pentru conducerea automată a vehiculelor în zone urbane.....	20
8.2. Contribuții originale.....	20
8.2.1. Contribuții la folosirea unui ansamblu DCNN pentru detecția stării de ebrietate folosind analiza imaginilor faciale termice în spectrul infraroșu.....	20
8.2.2. Contribuții la folosirea unei arhitecturi configurabile DCNN pentru clasificarea imaginilor hiperspectrale.....	20
8.2.3. Contribuții la clasificarea imaginilor hiperspectrale folosind augmentarea elementelor de antrenare cu sisteme GAN.....	21
8.2.4. Contribuții la clasificarea solicitărilor de credit utilizând optimizarea modelului Deep Learning.....	21
8.2.5. Contribuții la folosirea unor tehnici de tip DCNN folosite pentru conducerea automată a vehiculelor în zone urbane.....	22
8.3. Lista lucrărilor originale.....	22
8.4. Perspective de dezvoltare ulterioară.....	23
<b>Bibliografie.....</b>	<b>24</b>

# Lista tabelelor

Tabelul 3.1 Arhitectura modelului DCNN .....	5
Tabelul 3.3 Rate de detecție a stării de ebrietate în cazul independent de subiect.....	6
Tabelul 4.4 Rezultatele preciziei de clasificare prin metoda Deep Learning pentru 200 de elemente de antrenament.....	8
Tabelul 4.5 Precizia rezultatelor clasificării SVM.....	9
Tabelul. 4.6 Precizia clasificării Deep Learning vs. SVM.....	9
Tabelul 5.1 Variante de alocare a pixelilor pentru fiecare clasă.....	12
Tabelul 5.4 Rezultatele clasificării în funcție de transformări și validare.....	12
Tabelul. 6.2 Arhitectura rețelei.....	14
Tabelul 6.6 Rezultatele preciziei de clasificare bazate pe modelul DCNN pentru 220 elemente de antrenament.....	16
Tabelul 7.3 Rezultate experimentale.....	18

# Lista figurilor

Fig. 3.1 Arhitectura ansamblului de doua rețele DCNN pentru detecția stării de ebrietate în cazul subiectului independent.....	5
Fig. 4.2 Arhitectura rețelei.....	7
Fig. 5.1 Arhitectura unui sistem GAN.....	10
Fig. 5.3 Arhitectura Generatorului și cea a Discriminatorului sunt similare.....	11
Fig. 5.4 Arhitectura DCNN pentru clasificare.....	11
Fig. 6.2 Arhitectura DCNN.....	14

# Capitolul 1

## Introducere

### 1.1 Prezentarea domeniului tezei de doctorat

### 1.2 Scopul tezei de doctorat

### 1.3 Conținutul tezei de doctorat

Lucrarea este împărțită în opt capitole în care se prezintă rețelele DCNN construite să susțină tehnici de prelucrare vizuală.

Capitolul 1 este un capitol introductiv unde domeniul, scopul și conținutul tezei sunt prezentate.

Capitolul 2 este un capitol despre Inteligența Artificială în general și Metoda Deep Learning în special, explicând componentele esențiale, mecanismele și interacțiunea lor.

Capitolul 3 prezintă o metodă nouă pentru detecția stării de ebrietate cu ajutorul camerelor termale și al unui ansamblu DCNN.

Capitolul 4 tratează folosirea unei DCNN cu o arhitectură originală în clasificarea imaginilor hiperspectrale.

Capitolul 5 introduce generarea unor pixeli hiperspectrali artificiali cu ajutorul tehnologiei GAN pentru clasificare pixelilor hiperspectrali.

Capitolul 6 prezintă coordonarea antrenării DCNN cu ajutorul unor algoritmi automați de optimizare pentru clasificarea unor cereri de credit.

Capitolul 7 tratează transferul învățării în contextul detecției de obiecte din domeniul traficului auto urban.

Capitolul 8 prezintă rezultatele relevante ale aplicării DCNN, concluziile lucrării, contribuțiile aduse și perspective de dezvoltare ulterioară pentru fiecare din aplicațiile prezentate în capitolele 3-7.

# Capitolul 2

## Inteligența Artificială și Machine Learning

Inteligența Artificială este inteligența demonstrată de mașini și roboți, o inteligență creată de oameni și care include automatizarea unor activități umane ca planificarea, învățarea, raționarea, reprezentarea, percepția, și într-o anumită măsură chiar inteligența socială și creativitatea.

Machine Learning este domeniul Inteligenței Artificiale care se ocupă de componenta de învățare a mașinilor și pune la dispoziția arhitecților de mașini inteligente o serie de metode de învățare supervizată și învățare nesupervizată.

### 2.1 Deep Learning – Concepte Esențiale

Deep Learning este o tehnică de învățare supervizată a Machine Learning, inspirată din funcționarea creierului uman. Așa cum creierul învață din exemple, rețelele neurale pot învăța prin expunerea la un număr mare de elemente, ale căror caracteristici sunt asimilate. Elementele pot fi imagini, text sau sunete. Pentru a permite învățarea unor obiecte sau structuri complexe, rețelele neurale Deep Learning folosesc un sistem de straturi de neuroni ale căror ponderi sunt dinamice, modificându-se pentru a se potrivi cu informația primită. Datorită numărului mare de straturi de neuroni, spunem despre aceste rețele că sunt adânci (deep).

Aplicațiile metodei pot fi condusul autonom, clasificarea unor elemente, traducerea unor texte, asistenții virtuali, detecția unor boli, recunoașterea fețelor, generarea de conținut media (fotografii, secvențe video, articole), analiza și predicția unor fenomene și evenimente, detecția fraudelor și chiar înlocuirea unui jucător uman în jocuri logice ca șah sau go. Tehnicile deep learning sunt folosite în toate aceste aplicații datorită rezultatelor bune obținute, rezultate care în unele cazuri depășesc rezultatele celor mai buni subiecți umani în aceste domenii. Aceste rezultate sunt suficient de motivante pentru viitoare încercări de creare a unor inteligente artificiale generale.

Un avantaj major al Deep Learning în față altor tehnici de Machine Learning este încorporarea pasului de extragere a caracteristicilor elementelor de intrare, pas care de obicei în alte tehnici de Machine Learning trebuie efectuat separat.

#### 2.1.1 Rețele Neurale Convoluționale (Convolutional Neural Networks - CNN)

#### 2.1.2 Stratul convoluțional

#### 2.1.3 Straturi de activare (Relu, Tahn, Sinusoidal)

#### 2.1.4 Straturi de punere în comun (Pooling)

## **2.2 Antrenarea unei DCNN**

Antrenarea unei DCNN este o problema complexă, incluzând următoarele componente:

- Găsirea unui set de date (pentru antrenarea rețelei) potrivit pentru obiectivul propus și împărțirea elementelor în elemente folosite pentru antrenament, elemente folosite pentru validarea antrenamentului și elemente folosite pentru test
- Folosirea unei arhitecturi potrivite obiectivului și setului de date
- Determinarea unui set de parametri de configurare, cei mai importanți dintre aceștia fiind considerați hiper-parametri
- Inițializarea ponderilor
- Folosirea unei funcții de eroare de antrenare, care calculează diferența dintre obiectivul rețelei și funcționarea propriu-zisă într-o epocă de antrenament
- Folosirea unui algoritm de optimizare a antrenării, algoritm ce determina modul în care se schimbă ponderile DCNN în funcție de ultima eroare determinată în timpul antrenării
- Productizarea rețelei. Implică instalarea pe un sistem hardware și software adecvat scopului propus.

Considerând complexitatea componentelor evidențiate anterior, este posibil ca antrenarea unei rețele să dureze zile sau chiar luni. Această antrenare este un proces iterativ, o parte din pași trebuie reluați pe măsură ce încep să apară rezultatele. Pentru a reduce timpul antrenării, se pot automatiza anumite mecanisme, astfel încât fiecare iterație de antrenare să aibă loc într-un timp cât mai scurt.

### **2.2.2 Back-propagation**

### **2.2.3 Gradient Descent și rata de învățare. Oprire anticipată**

### **2.2.4 Caracteristicile algoritmilor de antrenare**

### **2.2.5 Supra-adaptare (Overfitting)**

### **2.2.6 Validarea antrenamentului (Training Validation)**

### **2.2.7 Eroare de antrenare (Training Loss)**

### **2.2.7 Metode de accelerare a antrenării**

## **2.3 Aplicații CNN**

### **2.3.1 Clasificarea**

Una din aplicațiile deep learning este clasificarea. Clasificarea este un proces prin care o rețea neurală învață dintr-un set de elemente primit și apoi reușește să clasifice noi elemente. Clasificările pot fi nesupervizate, însemnând că rețeaua clasifică obiectele după anumite caracteristici comune într-un număr de clase definit de programator, sau supervizate, clasificări în care elementele de test sunt clasificate cu ajutorul feedbackului dat de cercetător în faza de antrenare.

### **2.3.2 Transferul învățării**

### **2.3.3 Detecția obiectelor**

## **2.4 Selecția unor arhitecturi DCNN consacrate**

## **2.5 Alte tipuri de rețele Deep Learning**

## **2.6 Cadre de dezvoltare**

## **2.7 Optimizarea performanței sistemului hardware**



# Capitolul 3

## Un ansamblu DCNN pentru detectia stării de ebrietate folosind analiza imaginilor faciale termice în spectrul infraroșu

### 3.1 Introducere

Această lucrare propune o metodă originală pentru detectarea stării de ebrietate folosind un ansamblu de două Deep Convolutional Neural Network (DCNN) pentru prelucrarea termică a imaginilor faciale în infraroșu ale subiecților care urmează să fie testați. Am ales varianta de detecție a stării de ebrietate independenta de subiect, recunoaștere care implică formarea de seturi de date pentru fiecare dintre subiecții J, folosind imaginile celorlalți (J-1) diferite de subiectul considerat pentru antrenament. Principalul obiectiv al cercetării noastre este de a obține o rată corectă de detecție de mai mult de 90% atât pentru starea de ebrietate cât și pentru cea sobră. Modelul este evaluat folosind un set de date de 400 de imagini termice în infraroșu.

### 3.2 Modelul propus pentru detectia stării de ebrietate utilizând o arhitectura duală DCNN

Rețelele Neurale Convoluționale sunt folosite în aplicații variate datorita capacității de a extrage modele și a capacității de folosire a acestor modele pentru recunoașterea și clasificarea obiectelor. Variații de DCNN includ diferite straturi cu diferite funcții: convoluție, dropout, activare, conectare totală (fully connected), soft-max, clasificare și altele.

Aceasta lucrare își propune să proceseze detectia stării de ebrietate din imagini faciale termice în spectrul infraroșu, folosind un sistem compus dintr-un ansamblu de 2 rețele DCNN. Cele 2 module DCNN ale ansamblului au câte doisprezece, respectiv zece straturi. Diferența e data de al treilea strat convoluțional (cu încă un strat dropout) folosit doar în una din cele doua rețele.

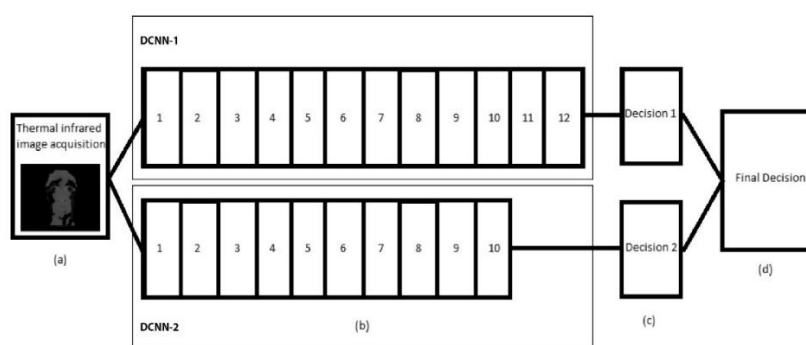
**Tab. 3.1** Arhitectura modulelor DCNN

Numar strat	DCNN-1	DCNN-2
1	Strat de intrare a imaginii	Strat de intrare a imaginii
2	Strat convoluțional 1	Strat convoluțional 1
3	Strat Dropout	Strat Dropout
4	Strat convoluțional 2	Strat convoluțională 2
5	Strat Dropout	Strat Dropout
6	Strat convoluțional 3	Strat Conectat Total
7	Strat Dropout	Strat de Activare
8	Strat Conectat Total	Strat Conectat Total
9	Strat de Activare	Strat Softmax
10	Strat Conectat Total	Strat de Clasificare
11	Strat Softmax	
12	Strat de Clasificare	

Cele doua rețele DCNN au fost antrenate separat, folosind aceleași valori pentru o parte din parametri, restul parametrilor fiind diferiți. Ca efect, avem doua rețele DCNN independente, antrenate independent, cu decizii independente. Mecanismul decizional final poate fi descris astfel:

Când ambele rețele DCNN ajung la aceeași decizie (stare de ebrietate sau stare treaza), consideram decizia lor ca decizie finala

Când cele doua rețele DCNN iau decizii diferite, decizia luata cu o încredere mai mare este selectata ca decizie finala.



**Fig. 3.1** Arhitectura ansamblului de doua rețele DCNN pentru detecția stării de ebrietate în cazul subiectului independent: (a) încărcarea imaginii termice infraroșii; (b) cele doua rețele DCNN de doisprezece, respectiv zece straturi; (c) deciziile independente ale celor doua rețele; (d) decizia finala rezultata din cele doua decizii independente

## 3.2 Experimente

### 3.2.1 Setul de date cu imagini termice în spectrul infraroșu

### 3.2.2 Pregătirea parametrilor și antrenarea rețelei

## 3.3 Rezultate

Rezultatele detecției independente de subiect sunt prezentate în Tab. 3.3, atât ca performanță pentru clasificarea fiecărui subiect, cât și ca performanță globală. Se poate observa o performanță minimă de 90% pentru subiectul numărul 7, o performanță maximă de 97.5% pentru subiecții 2, 3, 6, 8 și 10, cât și o performanță globală de 95.75%.

*Tab. 3.3 Rate de detecție a stării de ebrietate în cazul independenței de subiect*

Numărul subiectului testat pentru stare de ebrietate cu sistemul de rețele DCNN antrenat pe ceilalți 9 subiecți	DCNN 1	DCNN 2	Ansamblu de 2 DCNN
1	90.00%	95.00%	95.00%
2	97.50%	95.00%	97.50%
3	95.00%	92.50%	97.50%
4	95.00%	95.00%	95.00%
5	92.50%	90.00%	95.00%
6	97.50%	95.00%	97.50%
7	90.00%	87.50%	90.00%
8	92.50%	90.00%	97.50%
9	92.50%	92.50%	95.00%
10	90.00%	95.00%	97.50%
Rata de deviatie medie	93.25%	92.75%	95.75%
Deviatie standard	3.89%	3.69%	3.18%

# Capitolul 4

## O arhitectură DCNN configurabilă pentru clasificarea imaginilor hiperspectrale

### 4.1 Introducere

Imaginile hiperspectrale sunt imagini obținute prin colectarea și procesarea informațiilor din spectrul electromagnetic. Fiecare pixel al imaginii este o colecție de răspunsuri primite de la un punct material prin reflectarea undelor electromagnetice cu diferite forme de unda. Cu zeci sau sute de forme de unda și o mare rezoluție a imaginilor, clasificarea pixelilor devine un lucru greu de realizat pentru oameni. Metodele ce folosesc machine learning (ML) pot răspunde acestor complexități și include multe subdomenii care pot susține aceste clasificări. Intre metodele Machine Learning, metodele ce folosesc Deep Learning (DL) sunt foarte folosite în aceasta perioada și sunt îmbunătățite permanent. Unul din avantajele lor este faptul că imaginile hiperspectrale procesate de ele nu necesita îmbunătățiri sau transformări înainte de a fi folosite, înțelese și clasificate.

### 4.2 Metoda propusă

Clasificarea pixelilor poate fi realizată pe baza caracteristicilor pixelului clasificat și al pixelilor din jurul sau. Aceste opțiuni se numesc corelație spectrală și corelație spectrală-spațială. Numărul de pixeli vecini este ales considerând aranjarea pixelilor din jurul subiectului: un prim strat are 8 pixeli vecini, 2 straturi au 24 de pixeli, 3 straturi au 48 de pixeli. Alte lucrări au demonstrat o mai bună clasificare este obținută dacă sunt luați în considerare și vecinii. S-a demonstrat și că după un anumit prag, mai mulți vecini nu ajută clasificarea.

Arhitectura rețelei este bazată pe trasee paralele de straturi convoluționale și este prezentată în Fig. 4.2.

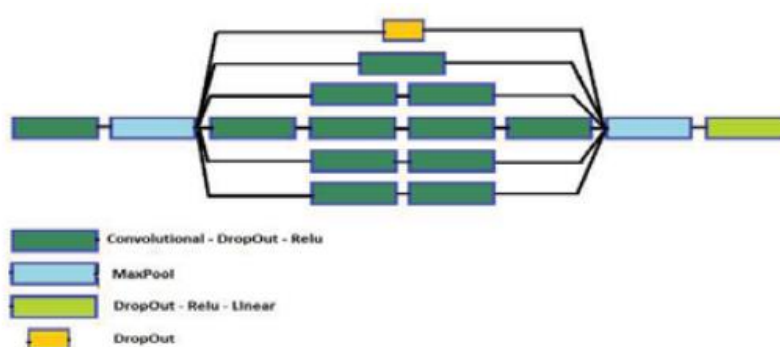


Fig. 4.2 Arhitectura rețelei

Modul în care se realizează antrenamentul este influențat puternic de către algoritmul de optimizare și de funcția de eroare de antrenare. Candidații încercați de noi pentru găsirea algoritmului optim de optimizare au fost: Stochastic Gradient Descent, Adaptive Moment Estimation și Adagrad, în timp ce funcțiile de eroare de antrenare încercate au fost Cross Entropy Loss, Negative Log Likelihood Loss, Connectionist Temporal Classification și Multi Margin Loss.

Algoritmul de optimizare care a funcționat cel mai bine cu setul de date Pavia University a fost Adaptive Moment Estimation. Acest algoritm are o rată de învățare adaptivă.

Dintre funcțiile de eroare de antrenare, Multi Margin Loss și Cross Entropy Loss au avut rezultate mult mai bune decât Negative Log Likelihood Loss și Connectionist Temporal Classification. Aceste rezultate (mai bune pentru primele două funcții) au fost confirmate cu fiecare din algoritmi de optimizare menționați mai sus. Am concluzionat că aceste funcții de eroare de antrenare sunt potrivite pentru tipul de set de date folosit. În final, funcția Multi Margin Loss a fost folosită pentru antrenament (datorită vitezei mai bune) în timp ce funcția Cross Entropy Loss a fost folosită pentru o pre-antrenare a rețelei.

## 4.3 Rezultate experimentale

### 4.3.1 Setul de date

Pentru această lucrare a fost ales setul de date Pavia University.

### 4.3.2 Rezultate

*Tab. 4.4 Rezultatele preciziei de clasificare prin metoda Deep Learning pentru 200 de elemente de antrenament*

#	Vecinatate	Epoci	Algoritm de optimizare	Funcția de eroare de antrenare	Clasificare Corectă
1	0	123	ADAM	CELoss	78.6%
2	8	80	ADAM	CELoss	82.4%
3	24	1014	ADAM	CELoss	85.1%
4	24	815	ADAM	MMLoss	89.8%
5	24	1262	SGD	MMLoss	84.8%
6	48	2953	ADAM	MMLoss	83.6%

Scorul de clasificare este media scorurilor de clasificare corectă pentru fiecare clasă. Două din clase au fost clasificate 100% corect (clasele Metal și Umbre) sic el mai mic scor a fost obținut pentru Lunca (83%). Rezultatele din Tab. 4.4 au fost obținute în mediul de dezvoltare PyTorch. În matlab rezultatele clasificării au fost aproape apropiate, dar cu 1%-2% mai puțin bune.

Metoda Support Vector Machine (SVM) a fost folosită pentru comparația rezultatelor. Tab. 4.5 prezintă precizia rezultatelor obținute în clasificarea SVM bazate pe numere diferite de pixeli de antrenament și numere diferite de vecini.

**Tab 4.5** Precizia rezultatelor clasificarii SVM

<b>Numar / Procent de pixeli de antrenament din fiecare clasa</b>	<b>8 Vecini</b>	<b>24 Vecini</b>
4%	61.1%	61.8%
20%	77.5%	77.9%
200	51.3%	51.7%
4% + 200	76.0%	76.9%

Precizia clasificării prin metodele Deep learning și SVM au fost comparate pentru același număr de pixeli de antrenament și același număr de vecini în Tab.4.6

**Tab. 4.6** Precizia clasificarii Deep Learning vs. SVM

<b>Metoda</b>	<b>Numarul pixelilor de antrenament</b>	<b>Vecini</b>	<b>Scor de clasificare corecta</b>
SVM	200/clasa	24	51.7%
Deep Learning	200/clasa	24	89.8%

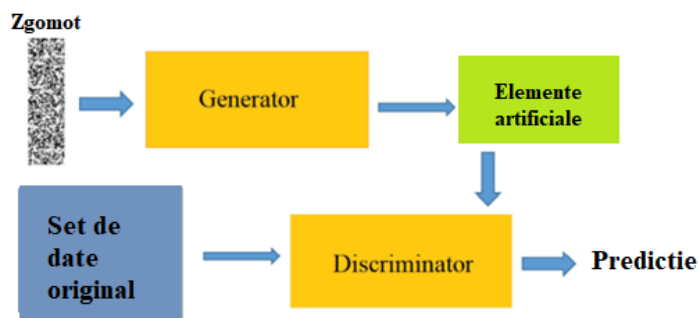
# Capitolul 5

## Clasificarea imaginilor hiperspectrale folosind augmentarea elementelor de antrenare cu sisteme GAN

### 5.1 Introducere

#### 5.1.1 Sisteme GAN

Sistemele GAN sunt folosite în mai mult de 1000 de aplicații ca generarea unor fețe umane, îmbătrânirea feței, transformarea textului în imagini sau predicția video. Noi elemente sunt generate dintr-o distribuție existentă de modele, păstrând caracteristicile modelelor existente. Generarea este realizată de un ansamblu de două rețele, una din ele numită Generator și cealaltă numită Discriminator. Aceste rețele sunt rețele nesupervizate și precizia rezultatelor lor este evaluată prin numărul de erori pe care îl fac.



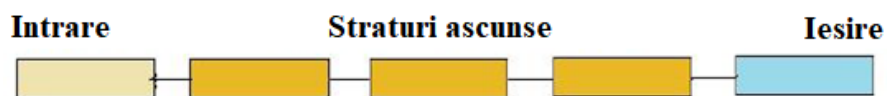
*Fig. 5.1 Arhitectura unui sistem GAN*

#### 5.1.2 Clasificarea imaginilor hiperspectrale

Scopul propus a fost creșterea preciziei de clasificare folosind diferite metode moderne bazate de Deep Learning. Metodă care a funcționat cel mai bine a fost o metodă care a folosit mai mulți pixeli pentru antrenament, acești pixeli suplimentari fiind obținuți cu GAN. Considerând că pixelii nu pot fi direct citiți sau înțeleși de oameni datorită complexității lor, generarea unor noi pixeli de calitate este un lucru deosebit de greu de realizat. Prin pixeli de calitate înțelegem pixeli care au aceleași caracteristici ca cele ale pixelilor originali, folosirea lor ducând la mărirea preciziei de clasificare a rețelei.

## 5.2 Metoda propusă

Pentru modelul GAN am folosit o arhitectura originala, asa cum este prezentata în Fig. 5.3.

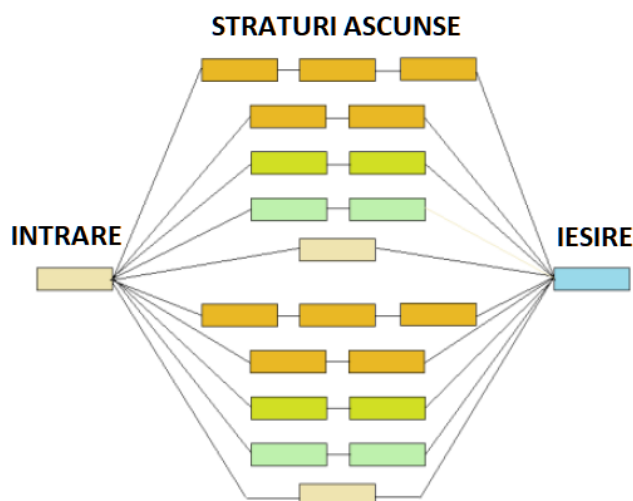


*Fig. 5.3 Arhitectura Generatorului și cea a Discriminatorului sunt similare*

Generatorul include 5 straturi convoluționale transpuse, primele 4 din ele (în bej și portocaliu) sunt fiecare urmărite de un strat de normalizare și un strat ReLU iar ultimul strat este precedat doar de un strat Tahn la sfârșit.

Discriminatorul include 5 straturi convoluționale, primele 4 din ele (în bej și portocaliu) sunt urmate de un strat de normalizare și un strat LeakyReLU. Ultimul strat convoluțional este urmat de un strat Sigmoid.

Arhitectură DCNN folosita pentru clasificarea pixelilor include trasee paralele: diferite tipuri de straturi convoluționale și straturi Gated Recurrent Layers (GRU). Toate straturi convoluționale (și cele GRU) sunt urmate de straturi de normalizare, straturi dropout, și straturi de activare LeakyReLU. Ultimele două straturi sunt un strat liniar și un strat de activare LogSoftmax. Cele 5 trasee de deasupra (Fig. 5.4) folosesc straturi convoluționale, în timp de ultimele 5 trasee folosesc straturi GRU. Culorile au fost folosite în felul următor: Bej: strat convoluțional, strat dropout, strat de activare, strat de normalizare; Portocaliu de 3 ori: strat convoluțional, strat dropout, strat de activare, strat de normalizare; Portocaliu de 2 ori: strat convoluțional, strat dropout, strat de activare, strat de normalizare; Verde: strat convolutional transpus, strat dropout, strat de activare, strat de normalizare; Verde deschis: strat convolutional transpus, strat dropout, strat de activare, strat de normalizare; Albastru (ultimul strat): strat liniar, strat dropout și ultimul strat de activare



*Fig. 5.4 Arhitectura DCNN pentru clasificare*



## 5.3 Rezultate experimentale

*Tab. 5.1 Variante de alocare a pixelilor pentru fiecare clasă*

Varianta de alocare pixeli	Pixeli originali pentru antrenament	Pixeli generați prin transformări clasice	Pixeli generați prin transformări GAN	Pixeli pentru validarea antrenamentului	Pixeli pentru test
1	300	0	0	0	42,476
2	200	0	0	100	42,476
3	300	500	0	0	41,976
4	200	500	0	100	41,976
5	300	0	500	0	42,476
6	200	0	500	100	42,476

Combinând cele 2 tipuri de transformări de pixeli (GAN și îmbunătățire clasică) cu folosirea sau nefolosirea validării, am obținut 6 tipuri de antrenament pentru care cele mai bune rezultate sunt prezentate în Tab. 5.4.

*Tab. 5.4 Rezultatele clasificării în funcție de transformări și validare*

Varianta de alocare a pixelilor	Alocarea pixelilor pentru un tip de transformare și validarea	Precizia clasificării
1	Fără îmbunătățire / Fără validare	87.43%
2	Fără îmbunătățire / Validare	92.94%
3	Îmbunătățire clasică / Fără validare	84.14%
4	Îmbunătățire clasică / Validare	86.76%
5	Îmbunătățire GAN / Fără validare	89.79%
6	Îmbunătățire GAN / Validare	95.32%

# Capitolul 6

## Clasificarea solicitărilor de credit utilizând optimizarea modelului Deep Learning

### 6.1 Introducere

#### 6.1.1 Riscul de credit financiar

Această lucrare se ocupă de evaluarea riscului de credit asociat cu împrumuturile acordate persoanelor fizice și nu se ocupă de riscul creditelor acordate agenților economici. Pentru scopul acestei lucrări, contextul macroeconomic este considerat stabil.

#### 6.1.2 Soluții bazate pe modelul DCNN pentru predicția riscurilor asociate creditelor

Scopul este obținerea unei metode mai bune decât tehnicile clasice de genul încercă și obține eroare (try and error) pentru obținerea unei arhitecturi bune sau a folosirii intuiției pentru selectarea super-parametrilor. Arhitectura și valorile super-parametrilor trebuie să fie rezultatul unor selecții realizate de către algoritmi, în concordanță cu setul de date și obiectivele proiectului.

### 6.2 Metoda Propusă

#### 6.2.1 Mediul de simulare

#### 6.2.2 Criterii de evaluare și costul creditului

#### 6.2.3 Algoritmi de optimizare

Pentru aplicația noastră, următorii algoritmi de optimizare au fost folosiți cu scopul minimizării funcției obiectiv:

- Algoritmul Gradient Boosted Regression Trees (GBRT) pornește un set de simulări bazate pe super-parametri aleatori într-un interval definit, apoi încearcă să găsească o combinație îmbunătățită de super-parametri construind structuri de tip arbori, apoi adăugând crengi (de soluții) arborilor celor mai interesanți
- Algoritmul Decision Trees (DT) este caracterizat de faptul că spațiul valorilor posibile este împărțit în regiuni distincte ai valorilor super-parametrilor. Fiecare observație este folosită pentru a calcula o medie a regiunii.
- Algoritmul Uniform Sampling (US) împarte spațiul valorilor fiecărui super-parametru în regiuni și încearcă soluții din fiecare regiune. Spațiul este împărțit într-un număr de regiuni dar de numărul de încercări cerute.

· Algoritmul Random Search [RS] este un algoritm complet aleator care reține cele mai bune soluții obținute din încercări aleatoare. Și-a dovedit o eficiență mai bună decât Uniform Sampling.

· Algoritmul Gradient Descent (GD) folosește algoritmi de optimizare a ponderilor rețelelor neurale. Acești algoritmi se găsesc în librăriile Pytorch: ADAM, Adagrad, Adadelta și alții. Acești algoritmi pornesc de la un punct în spațiul n-dimensional al soluțiilor și încearcă diferite posibilități iar în funcție de rezultatul obținut pentru o încercare își schimbă direcția și distanța de deplasare către noi valori ale super-parametrilor.

· Algoritmul Optimizarea Bayes (Bayesian Optimization - BO) este unul dintre cei mai folosiți algoritmi pentru Deep Learning. Probabilitatea a posteriori a unui model bazată pe o dovadă existența, este proporțională cu șansa acelei dovezi înmulțită cu o probabilitate a priori. Cea mai mare diferență dintre acest algoritm comparat cu cele 2 anterioare, este că toate valorile anterioare sunt luate în considerare pentru căutarea unei soluții. Algoritmul de optimizare lucrează în felul următor:

- Observarea unui număr de simulări anterioare, considerat informație existența
- Predicția a unei funcții pe baza informației existente cu un grad de incertitudine
- Modificarea continuă a funcției pe baza noilor soluții are ca efect reducerea incertitudinii

## 6.2.4 Arhitectura DCNN

Arhitectură optimă a rețelei este prezentată în Fig. 6.2.

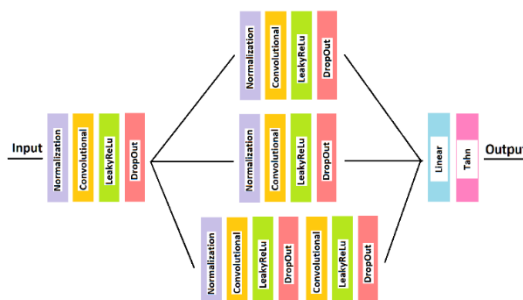


Fig. 6.2 Arhitectura DCNN

Straturile folosite în arhitectura optima DCNN sunt prezentate în Tab 6.2.

*Tab. 6.2 Arhitectura rețelei*

Număr strat	Strat
1 – 4	Batch Normalization - Conv - LeakyRelu - Dropout
5 – 8	Batch Normalization - Conv - LeakyRelu - Dropout
9 – 12	Batch Normalization - Conv - LeakyRelu - Dropout
13 – 19	Batch Normalization - Conv - LeakyRelu – Dropout - Conv - LeakyRelu - Dropout
20 – 21	Linear – Tahn

## 6.3 Experimente și rezultate

### 6.3.1 Setul de date

Această lucrare folosește setul de date German Credit, conținând 1000 înregistrări cu 20 de atribute fiecare. Fiecare înregistrare reprezintă informațiile oferite de către o persoană care a cerut credit și este clasificată ca o oportunitate bună sau slabă pentru oferirea creditului. Există 700 de înregistrări pre-clasificate ca bune și 300 de înregistrări pre-clasificate ca slabe.

### 6.3.2 Performante și hiper-parametri optimi

#### 6.3.3 Rezultate experimentale

Șase algoritmi de optimizare au fost folosiți pentru găsirea super-parametrilor optimi pentru antrenarea DCNN. În general, s-a pornit cu intervale largi de valori pentru fiecare hiper-parametru și după câteva runde de observație pentru fiecare algoritm, intervalele au fost micșorate. De exemplu pentru parametrul rată de învățare, căutarea a pornit cu un interval larg (10-8, 0.9) ce a fost redus la intervalul (10-3, 0.1) după observații care au arătat că învățarea nu are loc mai ales pentru valori foarte mici ale intervalului inițial.

O parte din algoritmi s-au dovedit mai buni în explorarea inițială (Random Search, Uniform Sampling) iar alții (Gradient Boosted Decision Trees and Gradient Descent Optimization) s-au dovedit mai buni la exploatarea unor zone cu rezultate bune. Algoritmii mai buni în zona de exploatare s-au dovedit și mai stabili ca răspuns la valori inițiale ale ponderilor rețelei, considerând strategiile de inițializare prezentate anterior în această lucrare.

Algoritmul de optimizare Random Search a fost cel mai ușor de folosit. Doar a primit intervalele de valori pentru fiecare super-parametru și a ales aleator diferite combinații de parametri. A fost rapid pentru obținerea unor rezultate acceptabile și îl recomandăm pentru a fi folosit la începutul dezvoltării DCNN.

Algoritmul de optimizare Uniform Sampling s-a dovedit a fi foarte dependent de distanță dintre 2 selecții, așa că obținerea unor rezultate bune pe intervale inițiale mari a durat foarte mult timp. Așa cum era de așteptat, chiar și după descoperirea unor valori interesante, algoritmul trece mai departe, trecând în sub-intervale lungi neinteresante. Ca și algoritmul Random Search, acest algoritm poate fi folosit în etapă de început a dezvoltării DCNN.

Algoritmul de optimizare Gradient Boosted Regression Trees a prezentat o viteză relativ mare de convergență către rezultate bune de precizie a clasificării, apoi a durat foarte mult pentru obținerea unor rezultate și mai bune. O supraîncărcare a procesorului s-a observat în timpul de definirea noi ramuri pentru arbori. Algoritmul decide ce ramuri trebuie selectate pentru dezvoltare și noi am observat o dependență a rezultatului final de alegerile precedente și alegerile inițiale. Ca efect, algoritmul are rezultate mai bune pentru exploatare decât pentru explorare.

Algoritmul de optimizare Decision Trees a convers greu către rezultate acceptabile în cele mai multe dintre simulări. Acest algoritm este pretabil unor planuri n-

dimensionale mai line, în care cel puțin teoretic va converge mai rapid decât altele către rezultate foarte bune.

Cele mai bune rezultate au fost obținute folosind algoritmul Bayesian Optimization. Am remarcat rezultate consistente atât cu intervale scurte de valori cât și cu intervale mai lungi de valori și în final am obținut și cea mai bună precizie datorită lui. Acest algoritm de optimizare este chiar rapid în a converge către o eroare de evaluare bună (redușă) de credit, comparat cu ceilalți algoritmi la un număr similar de rulări. Aduce un mic overhead procesorului datorită calculării funcției după fiecare iterație și complexitatea lui crește de asemenea după fiecare iterație, deoarece adaugă din ce în ce mai multe rezultate precedente. Un alt avantaj al acestui algoritm este că flexibilitatea în termeni de explorare versus exploatare, cea ce îl face o bună alegere în diferite faze ale dezvoltării DCNN.

**Tab. 6.6** Rezultatele preciziei de clasificare bazate pe modelul DCNN pentru 220 elemente de antrenament.

#	Epoci	Algoritm de optimizare	Rata de învățare inițială	Optimizare ponderi DCNN/ Funcție de eroare de antrenare DCNN	Eroare de evaluare a cererilor de credit	MAR
1	86	GDO	0.001	ADAM / CEL	137	7%
2	20	DT	0.057	Adagrad / NLL	170	11%
3	85	BO	0.170	AdamW / MMLoss	<b>103</b>	<b>3%</b>
4	18	GBRT	0.050	Adam / NLL	152	10%
5	142	RS	0.756	AdamW / MMLoss	167	11%
6	267	US	0.065	AdamW / MMLoss	155	9%

# Capitolul 7

## Tehnici de tip DCNN folosite pentru conducerea automată a vehiculelor în zone urbane

### 7.1 Introducere

Construind pe eforturi anterioare legate de Machine Learning pentru detecția pietonilor și rețele neurale pentru navigare automată, în această lucrare aducem este adusă o contribuție pentru dezvoltarea sectorului auto prin introducerea unei tehnici noi de folosire a detecției de obiecte bazate pe DCNN în domeniul auto.

### 7.2 Transferul învățării

### 7.3 Sistemul DCNN YOLOv5

### 7.4 Metoda propusă

#### 7.4.1 Metoda propusă pentru transferul învățării

#### 7.4.2 Criteriile de evaluare

Mecanismul de detecție al obiectelor este bazat pe două componente: identificarea obiectelor pentru o imagine dată și predicția coordonatelor sale. Acestea pot fi măsurate folosind indicatorii de Găsire („Recall” – cât de bine sunt găsite obiectele) și Precizie (câte de bune sunt predicțiile coordonatelor).

Pentru scopul nostru de a măsura performanța DCNN pentru detecția obiectelor, am folosit mean Average Precision (mAP@0.5), un indicator compus din Găsire și Precizie. mAP@0.5 este un indicator de medie (calculat ca o arie acoperită pe graficul având ca axe Găsirea și Precizia) unde raportul dintre Intersecția și Uniunea ferestrelor reale și detectate este mai mare decât 0.5. O valoare mai mare decât 0.5 pentru raportul dintre Intersecție și Uniune este considerată ca reprezentând o detecție bună a obiectului. mAP@0.5 se calculează pentru toate obiectele și reprezintă o caracteristică a DCNN, nefiind relevând pentru obiecte singulare.

#### 7.4.3 Optimizarea parametrilor printr-un algoritm genetic

În căutarea hiper-parametrilor potriviți pentru setul de date BDD100K, am folosit o metodă pentru evoluția hiper-parametrilor. Acesta folosește algoritmi genetici ca mutația și cross-over-ul pentru modificarea valorilor hiper-parametrilor astfel încât să se obțină rezultate mai bune ale obiectivului definit.

Algoritmul genetic are nevoie de definirea unui număr de rulări și a unui număr de epoci pe rulare. Apoi, după fiecare rulare, algoritmul creează un set nou de hiper-parametri și pornește o nouă rulare cu numărul predefinit de epoci.

#### 7.4.4 Arhitectura software

YOLOv5 DCNN este scrisă în Python și rulată în frameworkul Pytorch. Experimentele noastre au fost rulate în Ubuntu Linux 20.04, folosind compilatorul Python 3.9 (beta) și librăriile NVIDIA CUDA 11.2.

### 7.5 Experimente și rezultate

#### 7.5.1 Seturile de date

„Obiecte obișnuite în context” (Common Objects în Context - COCO 2017) constituie setul de date folosit pentru dezvoltarea YOLOv5 DCNN. Setul de date include 80 de tipuri de obiecte în 1.5 milioane de instanțe.

Am folosit transferul învățării pentru recunoașterea obiectelor din setul de date O baza de imagini și filme legate de condus cu o mare diversitate (Large-scale Diverse Driving Video Database) produs de Universitatea Berkeley din Statele Unite ale Americii. Acest set de date include 100,000 de imagini de o mare diversitate incluzând multe orașe, multiple tipuri de vreme, multiple momente ale zilei (incluzând luminozitate scăzută și vizibilitate scăzută), multiple tipuri de scenă (incluzând obiecte suprapuse).

#### 7.5.2 Transformarea atributelor

#### 7.5.3 Rezultate experimentale

După încercarea a numeroase tehnici pentru un transfer eficient al învățării, cea mai potrivită tehnică pentru obiectivul nostru a fost următoarea:

Încercând să obținem un  $mAP@0.5$  cât mai bun, am modificat funcția de fitness în această direcție, acordându-i indicatorului  $mAP@0.5$  prioritate totală sau aproape totală. Împotriva intuiției, cele mai bune rezultate au fost obținute însă când am poziționat funcția fitness pe o direcție echilibrată:

$$[\text{Precizie, Găsire, } mAP@0.5, mAP@0.5:0.95] = [0.2, 0.2, 0.4, 0.4]$$

O instanță YOLOv5 DCNN deja antrenată cu COCO a fost folosită într-un proces iterativ de antrenare condus de algoritmul genetic pentru găsirea celor mai buni hiper-parametri pentru antrenare. Acest lucru s-a realizat selectând hiper-parametri care au condus la cele mai bune rezultate ale  $mAP@0.5$  după 6 epoci, apoi de la cele mai bune seturi de hiper-parametri s-a trecut la simulări de câte 50 de epoci conduse de același algoritmul genetic. Cele mai bune seturi de parametri au fost apoi folosite pentru rulări de 100 de epoci din care s-a obținut și cel mai bun scor global.

Cele mai relevante rezultate obținute de noi sunt prezentate în Tab 7.3

**Tab. 7.3 Rezultate experimentale**

Indicator de evaluare	mAP@0.5	Timp de antrenament	Timpul mediu de detecție pe imagine
Referința: Rețea <b>YOLOv5s</b> originala, testata cu COCO, rezoluție 640 / Rețea <b>YOLOv5s6</b> originala, testata cu COCO, rezoluție 1280	55.4 / 61.9	N/A	0.010s / 0.024s
Rețea <b>YOLOv5s6</b> antrenata cu BDD100K pentru 100 epoci, rezoluție 1280, testata cu BDD100K	62.6	65.5 ore	0.024s
Rețea <b>YOLOv5s</b> antrenata cu BDD100K pentru 100 epoci, rezoluție 640, testata cu BDD100K	59.6	18.1 ore	0.010s
Rețea <b>YOLOv5s</b> antrenata cu BDD100K pentru 100 epoci, rezoluție 1280, testata cu BDD100K	68.7	65.2 ore	0.024s
Rețea <b>YOLOv5s</b> antrenata cu BDD100K pentru 100 epoci cu 24 straturi nemodificate, rezoluție 640, testata cu BDD100K	19.6	17.3 ore	0.010s
Rețea <b>YOLOv5s</b> antrenata cu BDD100K pentru 100 epoci folosind algoritmul de optimizare ADAM în loc de SGD, rezoluție 640, testata cu BDD100K	30.2	17.5 ore	0.010s
Rezultate din alte articole [98] obținute pentru detecția obiectelor folosind setul de date BDD100K	45.7	N/A	N/A
Rezultate din alte articole [99] obținute pentru detecția obiectelor cu YOLO (antrenata de la zero, fără transferul învățării) cu setul de date BDD100K	18.6	N/A	N/A

Se poate observa că am obținut rezultate mai bune pentru detecția obiectelor din setul de date BDD100K în comparație cu rezultatele obținute de YOLOv5 cu setul de date COCO. Credem că explicația constă în faptul că numărul de clase de obiecte este mai mic în cazul BDD100K decât în cazul COCO. Pentru o aplicație specializată, un număr mai mic de clase de obiecte este realist, astfel încât rezultatul nostru poate fi considerat foarte bun.



# Capitolul 8

## Concluzii

### 8.1 Rezultate obținute

### 8.2 Contribuții originale

#### 8.2.1 Contribuții la folosirea unui ansamblu DCNN pentru detecția stării de ebrietate folosind imagini infraroșii termale faciale

Contribuțiile metodei pentru diagnoză stării de ebrietate din fotografiile termice ale fetelor umane sunt următoarele:

- Folosirea unui ansamblu de DCNN care are o performanță de detecție mai bună decât a unei singure DCNN și o flexibilitate mai mare derivată din faptul că cele 2 DCNN pot fi antrenate pe sisteme hardware independente.
- Metoda Deep Learning elimina nevoia de a folosi tehnici diferite pentru selecția caracteristicilor și clasificarea fotografiilor.

Aceste contribuții au fost prezentate în lucrarea:

V. E. Neagoie, **P. Diaconescu**, “An Ensemble of Deep Convolutional Neural Networks for Drunkenness Detection Using Thermal Infrared Facial Imagery,” Proc. 13th International Conference on Communications (COMM2020), Bucharest, Romania, Electr Network, June 19-20, 2020, pp. 147-150, WOS:000612723900026

#### 8.2.2 Contribuții la folosirea unei arhitecturi configurabile DCNN pentru clasificarea imaginilor hiperspectrale

În *Capitolul 4* am prezentat o metodă de a adapta arhitectură și hiperparametrii DCNN pentru un set de date și un obiectiv ales.

Am comparat 2 medii de dezvoltare (Matlab și Pytorch), am arătat impactul modificării algoritmului de optimizare (SGD sau ADAM), impactul pierderilor (CELoss sau MMLoss), impactul vecinătăților și al numărului de epoci asupra procentului de clasificare corectă a pixelilor hiperspectrați. De asemenea am comparat metodă deep learning cu metodă SVM și am obținut o performanță de clasificare foarte bună (89.8%).

Aceste contribuții au fost prezentate în lucrarea:

**P. Diaconescu**, V. E. Neagoie, “A Highly Configurable Deep Learning Architecture for Hyperspectral Image Classification,” Proc. IEEE 13th International Symposium on Applied Computational Intelligence And Informatics (SACI 2019), Timisoara, ROMANIA, May 29-31, 2019, pp.197-200, WOS:000610436600035

### 8.2.3 Contribuții la clasificarea imaginilor hiperspectrale folosind îmbunătățirea elementelor de training cu sisteme GAN

Contribuția prezentată în capitolul 5 constă în folosirea GAN pentru îmbunătățirea rezultatelor de clasificare ale unei DCNN prin folosirea unor elemente de antrenament create artificial pe baza caracteristicilor elementelor existente. Implementarea GAN este originală și este adaptată tipului de elemente folosite (pixeli spectrali cu diferite vecinătăți). Importanța contribuției este semnificativă și datorită faptului că achiziția mai multor elemente prin metodă clasică (fotografie satelitară) este foarte scumpă și necesită o pregătire prealabilă impresionantă).

Altă contribuție este folosirea optimizării Bayes pentru selecția automată a hiperparametrilor DCNN. Rezultatele au fost interesante și ne-intuitive, spre exemplu precizia clasificării a fost mai bună în cazul folosirii straturilor GRU decât în cazul folosirii straturilor convoluționale clasice.

Aceste contribuții au fost prezentate în lucrarea

V. E. Neagoe, **P. Diaconescu**, “CNN Hyperspectral Image Classification Using Training Sample Augmentation with Generative Adversarial Networks,” Proc. 13th International Conference on Communications (COMM2020), Bucharest, Romania, Electr Network, June 19-20, 2020, pp.515-519, WOS:000612723900091

### 8.2.4 Contribuții la clasificarea solicitărilor de credit utilizând optimizarea modelului Deep Learning

Folosirea algoritmilor de optimizare prezentată în capitolul 6 aduce contribuții semnificative ușurinței cu care instituțiile financiare pot utiliza tehnici moderne și eficiente pentru calcularea riscului asociat creditului financiar.

Algoritmii de optimizare au căutat o combinație optimă a:

- 6 tipuri de algoritmi de optimizare pentru găsirea arhitecturii optime sau superparametrilor optimi
- 9 tipuri de algoritmi de optimizare a gradientilor pentru antrenarea rețelei
- 8 tipuri de straturi de activare
- Diferite momente de oprire a antrenării
- Antrenare cu diferite precizii a datelor

Contribuția esențială este demonstrația că oricând se pot adăuga componente moderne pentru calcularea riscului financiar iar algoritmii de optimizare vor determina dacă aceste componente sunt potrivite și au un impact pozitiv.

Aceste contribuții au fost prezentate în lucrarea

**P. Diaconescu**, V. E. Neagoe, “Credit Scoring Using Deep Learning Driven by Optimization Algorithms,” Proc. 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI-2020), Bucharest, Romania, June 25-27, 2020, pp.1-6, WOS:000627393500021

## 8.2.5 Contributii la folosirea unor tehnici de tip Deep Learning folosite pentru conducerea automata a vehiculelor in zone urbane

În *Capitolul 7* am arătat cum din combinația unei DCNN de top și a unui set de date specializat într-un domeniu se pot atinge rezultate foarte bune într-un timp scurt dacă se folosește (transferă) învățarea generalizată inițială ca baza pentru specializarea DCNN în noul domeniu.

În funcție de obiectivul ales, modificarea funcției de fitness a YOLOv5 poate avea un efect pozitiv. Rezultatul obținut de noi este cel mai bun rezultat publicat până acum pentru această pereche DCNN - set de date.

Altă contribuție este oferirea unei părți de cod ca Open Source pentru ca oricine să poată accelera transferul învățării.

Aceste contribuții au fost prezentate în lucrarea:

[5] **P. Diaconescu**, V. Neagoe, „A Deep Learning Approach to Autonomous Driving in Urban Environment”, accepted by Scientific Bulletin of the Politehnica University of Bucharest, Series C, Electrical Engineering and Computer Science, to be indexed in Web of Science

## 8.3 Lista lucrărilor originale

[1] **P. Diaconescu**, V. E. Neagoe, “A Highly Configurable Deep Learning Architecture for Hyperspectral Image Classification,” Proc. IEEE *13th International Symposium on Applied Computational Intelligence And Informatics (SACI 2019)*, Timisoara, ROMANIA, May 29-31, 2019, pp.197-200, WOS:000610436600035

[2] V. E. Neagoe, **P. Diaconescu**, “An Ensemble of Deep Convolutional Neural Networks for Drunkenness Detection Using Thermal Infrared Facial Imagery,” Proc. 13th International Conference on Communications (COMM2020), Bucharest, Romania, Electr Network, June 19-20, 2020, pp. 147-150, WOS:000612723900026

[3] V. E. Neagoe, **P. Diaconescu**, “CNN Hyperspectral Image Classification Using Training Sample Augmentation with Generative Adversarial Networks,” Proc. 13th International Conference on Communications (COMM2020), Bucharest, Romania, Electr Network, June 19-20, 2020, pp.515-519, WOS:000612723900091

[4] **P. Diaconescu**, V. E. Neagoe, “Credit Scoring Using Deep Learning Driven by Optimization Algorithms,” Proc. *12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI-2020)*, Bucharest, Romania, June 25-27, 2020, pp.1-6, WOS:000627393500021

[5] **P. Diaconescu**, V. Neagoe, „A Deep Learning Approach to Autonomous Driving in Urban Environment”, accepted by Scientific Bulletin of the Politehnica University of Bucharest, Series C, Electrical Engineering and Computer Science, to be indexed in Web of Science

## 8.4 Perspective de dezvoltare ulterioară

Pe baza cercetării de până acum, se pot iniția o serie de dezvoltări.

Legat de ansamblul DCNN pentru detecția stării de ebrietate folosind imagini infraroșii termale faciale, se poate testa schimbarea DCNN folosite cu rețele DCNN mai adânci, create prin încercări sau folosirea unor DCNN cunoscute, apărute după publicarea articolului nostru.

În domeniul recunoașterii imaginilor hiperspectrale, DCNN-ul propus în lucrarea „O arhitectură configurabilă DCNN pentru clasificarea imaginilor hiperspectrale” poate fi îmbunătățit cu ajutorul unor algoritmi de optimizare care să aleagă arhitectură rețelei și hiper-parametri într-un mod automat. Alte seturi de date cu hărți diferite sau cu aceleași hărți provenite de la diferiți sateliți (folosind un număr diferit de benzi spectrale pentru aceeași harta) pot fi adăugate. În domeniul clasificării imaginilor hiperspectrale folosind îmbunătățirea elementelor de training cu sisteme GAN, se pot testa și alte tipuri de rețele generative și se pot rula mai multe simulări (presupunând accesul la un sistem hardware mai puternic), deoarece numărul de rulări cu scop generativ a fost limitat de durata mare a simulării.

Lucrarea „Scoruri de credit folosind învățarea adâncă condusă de algoritmi de optimizare” ar putea beneficia de introducerea unor seturi noi de date și a noi algoritmi de optimizare a hiper-parametrilor. S-ar putea încerca selectarea automată a unui algoritm de optimizare în funcție de setul de date și de obiectivul propus.

În domeniul dezvoltării tehnicilor de tip Deep Learning folosite pentru conducerea automată a vehiculelor în zone urbane, se pot utiliza măsuri diferite de măsurare a detecției obiectelor și de analiză a erorilor de detecție.

# Bibliografie

- [1] A. Krizhevsky, Ilya, Sutskever, G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, 2012, <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf> (Reteaua AlexNet)
- [2] D. Ciresan, U. Meier, and J. Schmidhuber. *Multi-column deep neural networks for image classification*. Arxiv preprint arXiv:1202.2745, 2012.
- [3] B. Hunicka, H. Laurell, H. Bergman, *Psychosocial characteristics of drunk drivers assessed by addiction severity index, prediction of relapse*, Scandinavian Journal of Public Health, vol. 38, no 1, pp. 71–77, 2010.
- [4] D.A. Socolinsky, A. Selinger, J.D. Neuheisel, *Face recognition with visible and thermal infrared imagery*, Computer Vision and Image Understanding, vol. 91, pp. 72-114, 2003.
- [5] V.E. Neagoe, A.D. Ropot, *Concurrent self-organizing maps for pattern classification*, Proc. First IEEE International Conference on Cognitive Informatics, p. 304-312, 2002.
- [6] V.E. Neagoe, A.P. Barar, N. Sebe, P. Robitu, *A deep learning approach for subject independent emotion recognition from facial expressions*, Proc. 1st International Conference on Image Processing and Pattern Recognition (IPPR13), pp. 93-98, 2013.
- [7] V.E. Neagoe, S.V. Carata, *Subject independent drunkenness detection using pulse-coupled neural network segmentation of thermal infrared facial imagery*, Int. J Math. Comput. Methods, pp. 305-312, 2016.
- [8] V.E. Neagoe, S.V. Carata, *Drunkenness diagnosis using a neural network-based approach for analysis of facial images in the thermal infrared spectrum*, Proc. 6th IEEE International Conference on E Health and Bioengineering (EHB), pp. 165-168, 2017.
- [9] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, L. Carin, *Deep learning with hierarchical convolutional factor analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no 8, pp. 1887-1901, 2013.
- [10] N. Srivastava, G. Hinton, A. Krizhevsky, A. Sutskever, R. Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.
- [11] Y. LeCun, Y. Bengio, G. Hinton, *Deep learning*, Nature, pp. 436-444, 2015.
- [12] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, Massachusetts: MIT Press, 2016.
- [13] Q. Gao, S. Lim, *Hyperspectral Image Classification Using Convolutional Neural Networks and Multiple Feature Learning*, Remote Sensing Journal, 2018
- [14] V.E. Neagoe, V. Chirilă-Berbentea, *A Novel Approach for Semi-supervised Classification of Remote Sensing Images using a Clustering-based Selection of Training Data according to their GMM Responsibilities*, Proceedings of 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS2017), Fort Worth, Texas, USA, July 23–28, 2017, pp. 4730-4733
- [15] V. E. Neagoe, A. D. Ciotec, *A New Approach for Accurate Classification of Hyperspectral Images Using Virtual Sample Generation by Concurrent Self-Organizing Maps*, Proc. IEEE International Geoscience & Remote Sensing Symposium (IGARSS2013), Melbourne, Australia, 21-26 July, 2013, pp. 1031-1034

- [16] M. Bejiga, F. Melgani, *An adversarial approach to cross-sensor hyperspectral data classification*, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 3583-3586, July 22-27, 2018.
- [17] S. Fang, D. Quan, S. Wang, L. Zhang, L. Zhou, *A two-branch network with semi-supervised learning for hyperspectral classification*, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 3868 – 3871, July 22-27, 2018.
- [18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair et al., *Generative adversarial nets*, arXiv:1406.2661,
- [19] V.E. Neagoe, A.D. Ciotec, L. Bruzzone, *A weakly-supervised change detection technique for SAR images based on deep learning and synthetic training data generated by an ensemble of self-organizing maps*, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2019), Yokohana, Japan, pp.1669-1672, July 28-August 2, 2019.
- [20] V. E. Neagoe, R. Iatan, I.F. Iatan, *A nonlinear neuro-fuzzy model for prediction of daily exchange rates*, Proc. 5th International Symposium on Soft Computing for Industry, held at the World Automation Congress (WAC'04), Seville, June 28-July 1, 2004, published in : Soft Computing with Industrial Applications, Vol. 17, Book Series: TSI Press, pp. 573-578
- [21] T. Damrongsakmethee, V. Neagoe, *A Neural NARX Approach for Exchange Rate Forecasting*, in Proc Electronics, Computers and Artificial Intelligence (ECAI), 11th edition, Pitesti, 2019, pp. 1-6
- [22] V. Neagoe, A. Ciotec, G. Cucu, *Deep convolutional neural networks versus multilayer perceptron for financial prediction*, in Proc 2018 International Conference on Communications (COMM), 14-16 June 2018, pp 201-206
- [23] J. Bergstra, D. Yamins, D. Cox, *Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures*, Proc. of the 30th International Conference on Machine Learning (ICML) 2013
- [24] J. Bergstra, Y. Bengio, *Random Search for Hyper-Parameter Optimization*, Journal of Machine Learning Research 13, pp. 281-305, 2012
- [25] A. M. Turing, *Computing Machinery and Intelligence*, Mind, Volumul 49, pp. 433-460, Octombrie 1950
- [26] Dheekonda, Raja & Panda, Sampad & Khan, Md & Hasan, Mohammad & Anwar, Sohel *Object Detection from a Vehicle Using Deep Learning Network and Future Integration with Multi-Sensor Fusion Algorithm*. 10.4271/2017-01-0117.
- [27] A.D. Ciotec, V. E. Neagoe, A. P. Bărar, *Concurrent Self-Organizing Maps for Pedestrian Detection in Thermal Imagery*, Scientific Bulletin of the Polytechnic University of Bucharest, Series C, Vol. 75, Iss. 4, 2013, ISSN 2286-3540.
- [28] V. Neagoe, M. Vâlcu, and B. Sabac, *A Neural Approach for Detection of Road Direction in Autonomous Navigation*, in: Computational Intelligence, Theory and Applications, (ed. B. Reusch), Elsevier, Berlin-New York, 1999, pp. 324-333.
- [29] V.E. Neagoe, C.T. Tudoran, *A Neural Machine Vision Model for Road Detection in Autonomous Navigation*, Scientific Bulletin of the Politehnica University of Bucharest, Series C - Electrical Engineering, No 2, 2011, pp. 167-178.