

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Animal Image Recognition

propusă de

Șerban-Mihai Botez

Sesiunea: *Iulie, 2019*

Coordonator științific

Lect. dr. Anca Ignat

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI
FACULTATEA DE INFORMATICĂ

Animal Image recognition

Șerban-Mihai Botez

Sesiunea: *Iulie, 2019*

Coordonator științific

Lect. dr. Anca Ignat

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP,
absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de
..... specializarea, promoția
....., declar pe propria răspundere, cunoscând consecințele falsului în
declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr.
1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elaborată sub îndrumarea dl. / d-na
_____, pe care urmează să o susțină în fața
comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată
prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la
introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări
științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei
lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie
răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am
întreprins-o.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Animal Image Recognition*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică. De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 27.06.2019

Absolvent Șerban-Mihai Botez

(semnătura în original)

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări,
Șerban-Mihai Botez.

Încheierea acestui acord este necesară din următoarele motive:

[Se explică de ce este necesar un acord, se descriu originile resurselor utilizate în realizarea produsului-program (personal, tehnologii, fonduri) și aportul adus de fiecare resursă.]

Iași, 27.06.2019

Decan *Adrian Iftene*

(semnătura în original)

Absolvent *Șerban-Mihai Botez*

(semnătura în original)

Cuprins

Cuprins	1
Motivație	3
Introducere	4
Contribuții	5
1. Descrierea problemei.....	6
1.1 Abordari Anterioare.....	7
2. Procesarea Imaginilor	8
2.1 Tipuri de imagini.....	8
2.2 Adâncimea culorilor și formatul imaginilor.....	9
2.3 Zgomotul în imagini	10
3. Clasificatori în cascadă	11
3.1 Algoritmul Viola&Jones	11
3.2 Antrenarea clasificatorilor și detecția pe imagini	14
4. Rețele Neuronale	17
4.1 Rețele neuronale convoluționale.....	17
4.2 Antrenarea unui model pentru recunoașterea animalelor.....	18

5. Descrierea soluției.....	23
5.1 Arhitectura proiectului	19
5.2 Prezentare interfață	20
5.3 Prezentare output.....	21
5.4 Rezultate finale	22
6. Concluzii	23
Bibliografie.....	24

Motivație

Vederea Artificiala (Computer Vision) este o ramura a inteligenței artificiale care se ocupa de procesarea imaginilor din lumea reala. Este compusa din tehnici low-level de procesare a imaginilor, cum ar fi eliminarea zgomotului sau redimensionarea imaginilor și tehnici high-level (recunoasterea modelelor și clasificarea lor). Recunoasterea faciala este cel mai folosit domeniu al vederii artificiale, fiind necesara in multe sisteme de securitate, controlul accesului si marketing.

Recunoașterea faciala pe animala este in mare parte folosită pentru aplicații de gasire a animalelor pierdute, in clasificarea automatizata a speciilor dar și in preluarea de imagini sau videoclipuri ale unor rase specifice in mediul lor natural, cu interacțiune umana cat mai limitata.

Chiar daca exista deja multe tehnici si implementari de recunoastere facială, nu s-a ajuns la un sistem cu o acuratețe de 100% deoarece majoritatea imaginilor sunt preluate din lumea reala. Astfel, exista posibilitatea continua de a imbunatați abordarile folosite, lucru care atrage tot mai mulți cercetatori si pasionați spre această ramura a inteligenței artificiale.

Introducere

Aceasta lucrare are ca scop construirea și prezentarea unei aplicații prin intermediul careia se pot detecta fețe și corpuri de animale în imaginile furnizate ca input, folosind clasificatori și modele deja antrenate. De asemenea, aplicația va permite utilizatorilor să își antreneze proprii clasificatori sau modele, în funcție de animalul ce se dorește a fi găsit.

În continuare se vor prezenta capitolele lucrării, cât și o descriere în mare pentru fiecare capitol:

1. **Descrierea problemei.** În acest capitol se prezintă descrierea problemei abordate de această lucrare, dar și ideile principale ale problemei. Unele abordări și idei ale problemei inițiale nu au putut face parte din soluția finală. Ele vor fi prezentate în subcapitolul „Abordări anterioare”, împreună cu motivele pentru care s-au decis să se urmeze alte căi în rezolvarea problemei.
2. **Procesarea imaginilor.** În acest capitol se prezintă fundamentele procesării imaginilor și importanța acestora în obținerea unui model bine antrenat.
3. **Clasificatori în cascadă.** Acest capitol are rolul de a prezenta algoritmi folosiți pentru metoda cu clasificatori în cascadă, cât și cum funcționează de fapt această abordare pe câteva exemple practice.
4. **Rețele neuronale.** Acest capitol conține o scurtă introducere în conceptul de rețele neuronale, ce tip de rețele neuronale vor fi folosite pentru problema prezentată în prezenta lucrare cât și detalii despre procesul de crearea a unui model.
5. **Descrierea soluției.** În acest capitol se descrie arhitectura proiectului, împreună cu beneficiile fiecărei abordări alese și rezultatele finale obținute, însoțite de date statistice.
6. **Tehnologiile folosite.** În acest capitol sunt prezentate tehnologiile folosite, cât și o scurtă descriere pentru cele care nu sunt prezentate în programul de studii al facultății.
7. **Concluzii.** Acest ultim capitol aduce rezultatele tuturor abordărilor folosite, împreună cu beneficiile fiecărei abordări și opiniile personale legate de rezultatele lucrării. De asemenea, sunt prezentate și posibile direcții de viitor a proiectului și a problemei.

Contribuții

În această lucrare se dorește să se implementeze, antreneze și evalueze un model care este capabil să prezice animalele dintr-o imagine primită ca input. Există deja tehnici de depistare a anumitor obiecte dintr-o imagine, dar acestea funcționează cu o acuratețe bună doar pe obiecte care nu se mișcă foarte mult în diferite imagini (de exemplu steagul unei țări sau logo-ul unei companii), fie necesită o bază de cunoștințe în procesarea de imagini foarte amanunțită pentru a putea fi folosite de un public mai larg.

Elementul principal al acestei lucrări este reprezentat de crearea unei aplicații cu o interfață prietenoasă, care să poată fi folosită de persoane fără cunoștințe avansate în domeniu, dar și de persoane implicate în inteligența artificială și învățare automată. De asemenea, se încearcă ca modelul final să aibă o acuratețe cât mai ridicată împreună cu un timp rezonabil de antrenare pe imagini cât mai diversificate, pentru a nu se ignora partea practică proiectului.

Deoarece există enorm de multe cazuri în care un singur animal poate fi găsit într-o imagine, combinat cu multitudinea speciilor de animale ce prezintă trăsături diferite, modelul final poate fi mereu îmbunătățit prin folosirea de tehnologii diferite, combinarea diferitelor moduri de extragere a trăsăturilor sau procesarea imaginilor de antrenare.

Capitolul 1

Descrierea problemei

Problema care se dorește a se rezolva este recunoașterea animalelor dintr-o imagine furnizată de utilizator, folosind un model deja antrenat pe animale specifice. Pentru sistemul vizual uman, această problemă nu prezintă dificultăți atât timp cât se cunosc anumite trăsături ale speciei prezente în imagine. Pentru un computer se încearcă o abordare asemănătoare, bazându-se pe trăsături deja cunoscute luate din alte imagini unde se știe deja animalul prezent în imagine. Deoarece imaginile sunt stocate ca valori binare pentru fiecare pixel în parte, un computer are nevoie de o serie complexă de instrucțiuni pentru a recunoaște trăsăturile dintr-o imagine și a putea spune ce animal se găsește în acea imagine.

Metodele folosite sunt clasificatoare în cascadă și rețele neuronale pentru antrenare și detecție, folosind un set de imagini deja clasificate pentru obținerea trăsăturilor.

1.1 Abordari anterioare

În rezolvarea problemei s-au încercat mai multe tehnici pentru crearea modelului final, acestea având diferite motive pentru care nu au putut fi folosite în soluția finală. Câteva dintre aceste tehnici sunt: keypoint descriptors, image descriptors, local invariant descriptors.

Capitolul 2

Procesarea imaginilor

Înainte de a putea începe crearea unui model și antrenarea acestuia, trebuie să ne alegem un set de imagini care vor constitui setul de date de antrenament. În general, acest set de date trebuie să conțină cât mai multe imagini, pentru a ca modelul nostru să obțină cât mai multe trăsături pentru specia care se dorește a se găsi. De asemenea, imaginile dintr-un anumit set vor conține doar o specie de animale, pentru ca modelul să poată fi antrenat doar pentru acea specie și pentru a se reduce procentul detecțiilor fals pozitive (de exemplu, într-o imagine se detectează un câine ca fiind o pisică, pentru că aceste două specii au multe trăsături în comun).

Setul de date de antrenament conține două seturi de imagini: setul de imagini pozitive și setul de imagini negative. Imaginile pozitive sunt imagini care conțin animalul care se dorește a fi găsit din diferite unghiuri, pentru a se obține o gamă cât mai largă de trăsături. Imaginile negative pot fi imagini de orice tip, dar trebuie obligatoriu să nu conțină animalul cerut. Este de preferat ca setul de imagini negative să conțină imagini din mediul inconjurător al animalului care trebuie găsit, pentru a se putea face cât mai bine diferența dintre trăsăturile animalului și fundalul imaginii.

Pentru ca antrenarea să se realizeze într-un timp rezonabil și pentru a se obține o acuratețe cât mai bună, se vor aplica diferite procesări pe imagini (redimensionare, decupare, transformare în alb-negru, eliminarea zgomotului). Următoarele subcapitole vor prezenta aceste concepte, pentru a se putea înțelege mai bine termenii și asocierile folosite în următoarele capitole.

2.1. Tipuri de imagini

Imaginile pot fi împărțite în două categorii, în funcție de metoda de stocare folosită: imagini vectoriale și imagini digitale (acestea mai sunt numite și imagii raster sau bitmap). Imaginile vectoriale sunt combinații puncte care urmează mai multe funcții matematice. Deoarece majoritatea imaginilor vectoriale sunt de fapt linii trase între puncte, aceste imagini sunt independente de rezoluție și pot fi mărite și micșorate fără pierderi de calitate.

Imaginile digitale (raster sau bitmap) sunt formate din pixeli așezați pe o grilă. Pixelul reprezintă cel mai mic element dintr-o imagine, fiecare pixel având cate o valoare într-un anumit interval, în funcție de tipul de imagine digitală care este reprezentată. Rezoluția unei imagini este determinată de numărul de pixeli pe lățime și înălțime. Atunci când o imagine este digitală este redimensionată, calitatea pixelilor scade deoarece rezoluția se schimbă și numărul de pixeli trebuie marit sau scazut pentru a se ajunge la rezoluția dorită. În continuare se va lucra și discuta doar despre imagini digitale, deoarece este cel mai popular tip de reprezentare a imaginilor.

2.2. Adâncimea culorilor și formatul imaginilor

Numărul de culori diferite dintr-o imagine diferă în funcție de adâncimea culorilor sau numărul de biți pe un pixel. Mai jos se găsește un tabel ce conține câțiva biți și culorile asociate lor.

Bits per pixel	Number of colors
1 bpp	2 colors
2 bpp	4 colors
3 bpp	8 colors
4 bpp	16 colors
5 bpp	32 colors
6 bpp	64 colors
7 bpp	128 colors
8 bpp	256 colors
10 bpp	1024 colors
16 bpp	65536 colors
24 bpp	16777216 colors (16.7 million colors)
32 bpp	4294967296 colors (4294 million colors)

Sursă tabel : https://www.tutorialspoint.com/dip/concept_of_bits_per_pixel.htm

De obicei, imaginile alb-negru (grayscale) conțin 8 biți pe un pixel, ceea ce înseamnă 256 culori diferite. Imaginile colorate conțin de obicei 16 sau 24 biți pe un pixel. Pentru o imagine

cu 8 biți, valoarea unui pixel pentru culoarea negru este 0, în timp ce valoarea pentru culoarea alb este 255. În general la antrenare, imaginile sunt convertite în imagini grayscale pentru a reduce timpul de antrenare.

În funcție de modul în care sunt folosite imaginile, ele pot fi salvate în diferite formate. Cele mai populare formate sunt JPG, PNG, TIFF și GIF. Pentru imaginile din setul de antrenare se recomandă să se folosească formatul JPG, deoarece acest format comprimă imaginea și minimizează spațiul de pe disc. Chiar dacă imaginea își pierde câteva detalii în timpul compresiei, aspectul final al imaginii nu este cu mult diferit față de o imagine fără compresie.

2.3.Zgomotul în imagini

Zgomotul în imagini reprezintă diferite variații ale culorilor, luminozitate sau mediu înconjurător. Deoarece zgomotul are un impact foarte puternic asupra procesării imaginilor, este esențial ca acesta să fie eliminat pentru a se putea crea un model final cât mai bun.

Zgomotul creat de mediul înconjurător se referă la animalul principal din imagine și la fundalul imaginii care nu conține acel animal. Pentru setul de imagini pozitive este esențială eliminarea zgomotului creat de mediul înconjurător pentru a se putea extrage trăsăturile animalului. Eliminarea zgomotului se realizează prin decuparea animalului sau animalelor care se doresc a fi găsite din imagini. Dacă acest zgomot nu este eliminat, există posibilitatea ridicată ca să se găsească rezultate fals-pozitive (de exemplu fundalul unei imagini poate fi clasificat ca un animal).

Zgomotul creat de luminozitate sau variații de culoare se împarte în două categorii : zgomot gaussian și zgomot salt & pepper. Pentru obținerea unui model cât mai bine antrenat, se recomandă ca imaginile din setul de date de antrenament să nu conțină zgomot deoarece acesta poate scădea calitatea trăsăturilor extrase.

Capitolul 3

Clasificatori în cascadă

Un clasificator este un algoritm sau un set de funcții matematice care implementează un clasificator ceea ce are ca scop împărțirea instanțelor dintr-un set de date în mai multe categorii, în funcție de anumite instanțe ale caror categorii se cunosc.

Termenul „în cascadă” al unui clasificator reprezintă modalitatea prin care acesta este construit, prin ansamblarea clasficatorilor simpli pentru a se crea un clasificator final. Clasificatorul final este construit din mai mulți clasificatori simpli sau de bază (numiți stagii) care sunt aplicați pe anumite regiuni, în funcție de cat de multe trăsături se găsesc în acea regiune. Clasificatorii de bază sunt clasificatori bazați pe arbori de decizie ce conțin măcar doua nivele, input-ul fiind trăsăturile extrase din setul de date de antrenare. Pentru ansamblarea clasificatorilor simpli se folosește algoritmul discret.

3.1. Algoritmul Viola & Jones

Apărută în 2001, lucrarea „Rapid Object Detection using a Boosted Cascade of Simple Features” publicată de Paul Viola și Michael Jones conține a abordare bazată pe învățare automată pentru problema detecției obiectelor în timp real cu o acuratețe mare și timp de procesare mic pentru imagini.

Algoritmul prezentat în lucrare are patru etape care se execută succesiv pentru a extragerea trăsăturilor, antrenare și ansamblare. Chiar dacă algoritmul a fost creat inițial pentru detectarea fețelor umane, el funcționează pentru fețele animalelor cât și pentru obiecte simple (cu o acuratețe mai slabă, deoarece trăsăturile sunt extrase într-un mod specific fețelor).

1. Selectarea și extragerea trăsăturilor

Deoarece trăsăturile sunt esențiale în clasificarea imaginilor, procesul de creare a unui clasificator va începe cu selectarea și extragerea trăsăturilor. Motivele pentru care se lucrează cu trăsături și nu direct cu pixeli sunt performanța și imposibilitatea de a se face antrenarea pe un set atât de variat, deoarece fiecare pixel prezintă informații despre culoare și intensitate. În continuare vom discuta despre cele mai folosite tipuri de trăsături, Haar și LPB.

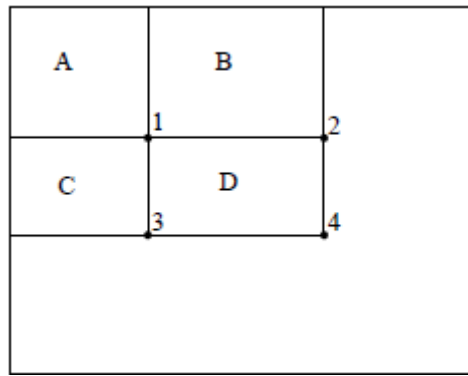
Trăsăturile de tip Haar sunt forme geometrice dreptunghiulare care sunt aplicate imaginilor din setul de antrenament și care selectează și extrag zonele cât mai bogate în trăsături din imagini. Câteva trăsături importante sunt : localizarea nasului și a gurii, tipul urechilor, forma nasului și a gurii. Alte trăsături mai pot fi culoarea regiunilor din împrejurul ochilor sau diferențele de culoare dintre diferite părți ale feței, dar aceste trăsături nu sunt neapărat valide la clasificarea animalelor deoarece culorile diferite ale pielii și părului pot afecta negativ clasificatorul.

Trăsăturile de tip LBP (local binary pattern) sunt grupări de pixeli clasificați prin verificarea vecinătăților pixelilor din imaginilor transformate în grayscale după niște praguri stabilite și scrierea valorilor de 0 pentru când valoarea vecinului este mai mică decât cea a pixelului după care se lucrează și 1 atunci când valoarea este mai mare.

În aplicația prezentată se folosesc trăsături de tip Haar deoarece acestea o acuratețe mai ridicată față de trăsăturile LPB (15-20%). Chiar dacă antrenarea unui clasificator de tip Haar necesită o durată mai lungă, acuratețea clasificatorului este esențială pentru o cât mai bună funcționare iar antrenarea se realizează o singură dată, rezultatul fiind salvat într-un fișier pentru a putea fi folosit mai târziu la detecție.

2. Crearea unei imagini integrale

Intr-o zonă de rezoluție 24 x 24 dintr-o imagine, numărul trăsăturilor dreptunghiulare este peste 160.000 iar pentru a se putea calcula cât mai repede trăsăturile rectangulare se folosește o reprezentare temporară a imaginii numită imagine integrală. O imagine integrală la un anumit punct de coordonate x și y conține suma pixelilor de deasupra și din dreapta punctului respectiv.



Fie figura de mai sus, folosind o imagine integrală, valoarea imaginii integrale în punctul 1 este suma pixelilor din chenarul A. Pentru locația 2 valoarea este $A + B$, pentru locația 3 valoarea este $A + C$ iar pentru locația 4 valoarea este $A + B + C + D$. Suma pixelilor din chenarul D poate fi calculă ca fiind $4 + 1 - (2 + 3)$, unde fiecare locație a fost calculată precedent.[3]

3. Antrenarea folosind AdaBoost

Algoritmul Adaboost este folosit într-o variantă modificată pentru a selecta un set restrâns de trăsături și pentru a antrena clasificatorul. De exemplu pe o zonă de rezoluție 24×24 care conține peste 160.000 trăsături, numărul trăsăturilor pastrate este între câteva sute și câteva mii. Alegerea trăsăturilor se face în funcție de cât de bine acestea fac separarea între imaginile pozitive și imaginile negative.

Pentru trăsăturile alese în primele stagii ale antrenării, rata erorii este între 0.1 și 0.3 (deoarece valorile, dar cu cât se avansează în stagii, obținerea trăsăturilor unice devine tot mai complicată iar rata erorii poate crește până la 0.4. În funcție de modul în care au fost procesate imaginile (mai ales cele din setul de imagini pozitive), această rată poate crește și afecta negativ clasificatorul, prin eliminarea unor trăsături care puteau să ajute la detecție, dacă imaginea era procesată corect.

4. Crearea clasificatorului în cascadă

Deoarece fiecare stagiou este un clasificator puternic, toate trăsăturile sunt grupate în diferite stagii cu diferite trăsături. Fiecare stagiou are ca scop clasificarea dacă o anumită trăsătura reprezintă fața unui animal sau nu, iar trăsăturile ce nu îndeplinesc această condiție sunt imediat eliminate.

Antrenarea și crearea clasificatorului se încheie de obicei atunci când s-a ajuns la numărul de stagii precizate, dar și atunci când se ajunge la potențialul statistic al clasificatorului (fie din cauză că sunt prea puține date sau parametrii antrenării au fost setați pentru a accepta o rată a clasificărilor false prea mare).

3.2. Antrenarea clasificatorului și detecția pe imagini

Pentru o mai ușoară folosire, aplicația prezintă o interfață care ajută utilizatorul să facă antrenarea și detecția animalelor folosind clasificatoare în cascadă.

În imaginea de mai sus se găsește meniul principal, împreună cu panoul prin care se poate face antrenarea. Înainte de a putea începe, utilizatorul trebuie să precizeze parametrii necesari antrenării, împreună cu path-urile către diferite foldere și fișiere. Parametrii sunt :

Data : Conține path-ul către un folder numit „data” unde va fi salvat output-ul programului. Output-ul constă în clasificatorul final, un fișier cu parametrii introduși la antrenare și câte un fișier ce conține progresul pentru fiecare stadiu în parte.

Vector :

Background :

Nr. of positive images :

Nr. of negative images :

Number of stages :

Width number :

Height number :

După ce s-au introdus toți parametrii, se poate apăsa butonul „confirm” pentru a începe antrenarea. Este de menționat faptul că antrenarea nu poate începe dacă nu s-au introdus toți parametrii sau nu au fost introduși corect. Utilizatorul va primi diferite mesaje de eroare, în funcție de greșelile făcute.

Rezultatul final este un clasicator ce poate fi folosit mai târziu pentru detecție.

Capitolul 4

Rețele neuronale

Descriere rețele neuronale.

4.1 Rețele neuronale convoluționale

4.2 Antrenarea unui model pentru recunoașterea animalelor

Capitolul 5

Descrierea soluției

Descrierea soluției, arhitectura proiectului, descrierea interfeței, prezentarea output-ului, rezultate finale.

Descriere soluție.

Arhitectura proiectului

Descriere interfață

Prezentare output

Rezultate finale

Concluzi

Bibliografie

- Paul Viola & Michael Jones, **Rapid object detection using a Boosted Cascade of Simple Features**, 2001 [1]
- Paul Viola & Michael Jones, **Robust Real-Time Face Detection**, 2003 [2]
- <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b> [3]

