

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Cancer ScanNet

Aplicație Python pentru diagnosticarea Cancerului histopatologic

propusă de

Alexandru Chiorianu

Sesiunea: februarie, 2019

Coordonator științific

Conf. Liviu Ciortuz

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ

Cancer ScanNet

Aplicație Python pentru diagnosticarea Cancerului histopatologic

Alexandru Chiorianu

Sesiunea: februarie, 2019

Coordonator științific

Conf. Liviu Ciortuz

Avizat,
Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

**DECLARAȚIE privind originalitatea conținutului lucrării de
licență/diplomă/disertație/absolvire**

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: _____

_____ela
borată 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/diplomă/disertație/absolvire 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

Cuprins

1 Introducere	4
1.1 MOTIVAȚIE	4
1.2 Context	6
1.3 Obiective generale	7
2 State of the art	7
2.1 Conceptul	7
2.1.1 Tipul Convolutiv	8
2.1.2 Tipul Pooling	9
2.1.3 Tipul Normalizare	10
2.1.4 Tipul Conectat complet	11
2.1.5 Tipul Dropout	12
2.1.6 Funcții de activare	13
3 Metodologia abordată	14
4 Modele testate	19
4.1 AlexNet	19
4.2 Folosirea grayscale	21
4.3 Rețeaua neuronală proprie grayscale	22
4.4 Rețea neuronală proprie grayscale - filtre diferite	22
5 Contributii:	25
6 Imbunătățiri:	25
7 Bibliografie	27

1. Introducere

Cancerul este unul din principalele cauze ale mortalității la nivel global, cu predispoziție în zonele mai puțin dezvoltate economic, deoarece zonele sub dezvoltate nu dispun de o infrastructura medicala eficace pentru detectarea lui timpurie.

Cele mai eficace metode pentru detectarea lui timpurie sunt biopsiile, care constau în prelevarea de țesut și examinarea lui pentru determinarea celulelor maligne.

Analiza acestor țesuturi ca și persoanele cu experienta pentru clasificarea lor, este destul de costisitoare deoarece necesita persoane antrenate și special pregătite pentru identificarea celulelor în cauza, timp și bani care ar putea fi investiți în cercetarea de noi tratamente ca și metode de prevenție.

1.1 MOTIVAȚIE

Dintre toate formele acestuia cancerul pulmonar este cel mai răspândit tip de cancer la nivel mondial. Cu aproximativ 1.8 milioane de cazuri noi în fiecare an pentru ambele sexe, este cauza a 1.6 milioane de decese la nivel global.

Deși majoritatea tipurilor de cancer pulmonar nu sunt foarte agresive, acesta este vinovat de 1 din 5 victime. Din cele două tipuri de cancer (celule mici/celule mare) cancerul pulmonar cu celule mici este cu mult mai periculos cu o șansă de supraviețuire de aproximativ 5 ani, însemnând 5% față de 70-90% pentru cel cu celule mari.

Diagnosticarea timpurie, precum și indicarea riscului mărit de cancer asupra indivizilor mărește șansele de supraviețuire, conducând până la tratarea completă.

Considerând că un individ este predispus cancerului, orice metodă de tratare al acestuia este binevenită. Astfel, pot fi luate măsuri de combatere și încetinire a evoluției bolii în timp util.

Datorită evoluției rapide a tehnologiei, perioada în care cancerul reprezenta o sentință la moarte incontestabilă este cu mult depășită. Astfel, prin intermediul noilor aparate, programe și instrumente, această boală necruțătoare poate fi identificată, tratată și, în cazurile cele mai fericite, înlăturată.

În ziua de azi, putem beneficia de avantajele oferite de puterea computațională precum și CAD¹ (Computer Asisted Diagnostics) grație cărora putem ajunge la evitarea pierderii unor vieți omenești prețioase.

Conceptul pornește de la următoarele elemente: emergența rețelelor neuronale, creșterea puterii computaționale precum și scalarea calculelor matematice pe procesorul grafic.

Avantaje primare:

- rețelele neuronale pot învăța din relațiile complexe și modelele non lineare, care se dovedesc a fi un avantaj major în conturarea vieții reale;
- rețelele neuronale pot generaliza, i.e. plecând de la relațiile inițiale furnizate de datele de input, o rețea neuronală poate deduce relațiile ce nu sunt întâlnite în datele de input, în așa fel încât să generalizeze și să prezică datele nevăzute;
- rețelele neuronale nu sunt foarte constrictive din punct de vedere al datelor de input în ideea distribuției datelor, astfel, descurcându-se cu facilitate cu volatilitatea mare și variația non constantă.

Luând în considerare tema și avantajele deja menționate, se poate afirma despre cele două elemente că sunt într-o relație de reciprocitate.

¹ "Kuni Doi *Computer-Aided Diagnosis in Medical Imaging*"
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1955762/>. Accessed 6 Feb. 2019.

1.2 Context

Conceptul de CAD sau Computer Aided Diagnostics, apărut la începutul anilor 2000, a ajuns să fie din ce în ce mai utilizat, altfel încât, în ultimii ani s-a pus accentul pe dezvoltarea și aplicarea cât mai frecventă a acestuia.

Printre avantajele majore întâlnite în folosirea CAD regăsim atât reducerea timpului de lucru pentru clasificarea bolilor și a tipurilor de țesuturi, precum și distincția între celule maligne și benigne.

Deși prezintă un mare număr de baze de date, una dintre cele mai folosite și aplicate în ultimul timp este PCam² (Patch Camelyon). Scopul este unul foarte simplu: constituie cea mai relevantă variantă prin prisma atribuțiilor practice pe care le are. Și anume detectarea celulelor de tip malign.

În baza de date inițială putem detecta, din păcate, imagini duplicate indezirabile în procesul de antrenare și validare al unui clasificator. Pentru a combate această situație vom apela la Kaggle³ spre a folosi setul de date pentru competiția de detecție a cancerului prin histopatologie.

Totuși, având în vedere faptul că factorul de decizie este unul binar, iar dificultatea provocării nu este una chiar foarte mare, putem fi încrezători în obținerea unor avantaje foarte mari în ciuda dificultăților întâlnite. Datorită acestui factor, în ultimul timp, setul de date PCam a ajuns să aibă o importanță la fel de mare precum CIFAR-10⁴, i.e. clasificarea cifrelor scrise de mână.

Deși conceptul de rețea neuronală există de câteva decade, datorită scalării și măririi puterii computaționale, doar de curând a ajuns să fie implementat tot mai frecvent. În general, fiecare utilizare conduce la descoperirea unor noi avantaje și de asemenea, o putere de clasificare ridicată.

² "CAMELYON16 - Home." <https://camelyon16.grand-challenge.org/>. Accessed 6 Feb. 2019.

³ "Kaggle." <https://www.kaggle.com/>. Accessed 6 Feb. 2019.

⁴ "CIFAR-10" <https://www.cs.toronto.edu/~kriz/cifar.html>. Accessed 6 Feb. 2019.

1.3 Obiective generale

Obiectivele generale ale lucrării sunt reprezentate de mai multe etape. Astfel, se va începe prin prezentarea unei scurte introduceri în CAD, urmând prelucrarea datelor și interacțiunea dintre lumea reală și digitală. Totodată, vom dezvolta și tema rețelelor convoluționale în cadrul procesării imaginilor digitale.

Un prim pas va fi analiza informațiilor prezente în setul de date, în această manieră putem observa mai ușor caracteristicile cele mai importante pe care le putem clasifica.

Vom experimenta cu diferite tipuri de rețele neuronale abordând atât arhitecturi cât și hiper parametri pentru a obține un rezultat cât mai exact.

2. State of the art

2.1 Conceptul

Deși rețelele neuronale au existat conceptual de foarte mult timp, emergența lor a fost în anul 2009, când, în 8 competiții globale de recunoaștere a pattern-urilor și machine learning, algoritmi propuși de Juergen Schmidhuber au ieșit învingătoare.

Aceeași situație a fost întâlnită și în cazul lui Alex Graves, care prin intermediul bi-direcționalității și al multi-dimensiunilor rețelelor neuronale cu memorie de lungă și scurtă durată, a câștigat 3 competiții în recunoașterea scrisului de mână în anul 2009.

Din 2011, în schimb, a apărut conceptele dominante de rețele neuronale și deep learning, acestea numindu-se, în mod oficial, Rețele Neuronale Convoluționale aka. CNN. De reținut este faptul că acestea au venit cu conceptul de pune mai multe layers de tipul Convolutiv urmat de

Max Pooling, una având rolul de a mări spațiul de trăsături cât mai mult, iar cealaltă de a extrage trăsăturile cele mai importante.

2.1.1 Tipul Convolutiv

2.1.1.1 Intuitiv

Tipul Convolutiv, având rolul de a extrage cât mai multe trăsături din input-ul primit prin intermediul unei ferestre, va ajuta la vizualizarea și recunoașterea anumitor pattern-uri. În primele straturi din rețea, ele vor ajunge să identifice prima dată muchii, nuanțe de culoare, forme simple. Însă, cu cât avansăm în rețea layer-ele convolutive vor avea rolul de a recunoaște structuri mai complexe, pornind de la o roată și ajungând până la o persoană sau chiar clădire.

2.1.1.2 Analogie

Pentru a ajunge la o analogie cu creierul uman, nivelul convolutiv este că un singur neuron se uită la o porțiune mică din imagine și împărtășește parametrii cu cei din stânga și din dreapta lui, activarea acestora fiind transpusă în următorul layer.

2.1.1.3 Conectivitate locala

Cel mai mare avantaj al layer-ului convolutiv este faptul că nu toți neuronii sunt conectați între ei, ci doar pe regiuni mici, în așa fel încât modelul să poată fi gestionat mai ușor (mai puține calcule de realizat și mai puține greutateți).

În cazul unui layer conectat complet, putem lua ca exemplu setul de date CIFAR-10 (numere scrise de mână 0-9). Imaginile din input au o mărime de $32 \times 32 \times 3$, astfel 32 de octeți lungime, lățime și 3 canale de culoare RGB⁵, fapte ce ar duce la 3072 de greutateți pentru primul

⁵ "Image Segmentation Using Color Spaces in OpenCV + Python – Real"
<https://realpython.com/python-opencv-color-spaces/>. Accessed 7 Feb. 2019.

layer. Acest aspect nedorit prezintă de multe ori un număr mare de parametri ce poate duce în overfitting.

În situația unui layer convolutional ajungem la un număr mai mic în cazul în care luăm, de exemplu, 32 de filtre de mărime 5×5 . Atunci fiecare neuron va avea 75 ($5 \times 5 \times 3$ RGB) de greutatea pentru un total de 2400 de greutatea.

În trecut, a fost preferită folosirea unor stride-uri cât mai mici de preferat 1. Totuși, în unele cazuri în care dimensiunea input-ului este foarte mare, se prefera folosirea stride-urilor cât mai mari, în defavoarea re-dimensionării care ar fi putut duce la pierderea unor informații cruciale.

De un lucru trebuie ținut cont, și anume asimetria dintre dimensiunile spațiale, lungimea și lățimea vor putea fi scalate prin intermediul convoluțiilor, dar adâncimea va rămâne aceeași în acest caz.

2.1.2 Tipul Pooling

2.1.2.1 Intuitiv

La un nivel intuitiv stratul de tip pooling are ca scop redimensionarea input-ului prin calcularea valorii maxime în raport cu mărimea unei ferestre.

Tot ce va face va fi ca din mărimea ferestrei să extragă valoarea maximă și să o trimită spre output.

La fel ca în cazul stratului convolutiv (adâncimea sau numărul de canale de culoare va fi păstrat constant între input și output).

În practică, rolul acestui layer este de a extrage porțiunile de interes din straturile convoluționale, de aceea se obișnuiește încheierea unuia sau mai multor straturi convoluționale cu un strat de tip pooling.

2.1.2.2 Analogie

Pentru o înțelegere mai ușoară, ne putem raporta la stratul de pooling ca un neuron ce așteaptă input-uri de la mai mulți neuroni vizuali și va extrage zona de interes. În cazul nostru, zona pe care vrem să o identificăm (de exemplu, o ușă sau o mașină).

2.1.2.3 Present

Este necesar să reținem faptul că straturile de pooling pot realiza și alte tipuri de funcții, cum ar fi pooling-ul mediu care extrage din fereastră valoarea medie la care se uită, sau normalizare L2.

În ultimii ani, a început un trend de eliminare al straturilor de pooling-ul în favoarea mai multor straturi convoluționale și pentru reducerea matricii de trăsături rezultate, să se folosească stride-uri mai mari în straturile convoluționale ulterioare celui de input.

2.1.3 Tipul Normalizare

2.1.3.1 Intuitiv

Rolul statului de normalizare este de a menține media activărilor între straturi cât mai aproape de 0, și a nu duce la overfitting bazat pe datele expuse până în momentul respectiv.

2.1.3.2 Analogie

Stratul de normalizare este folosit în special pentru a dezvolta un oarecare inhibitor în rețeaua neuronală și pentru a nu influența rezultatul final doar pe baza unui singur șir de activări din rețeaua neuronală.

Ideal se caută un model care să oglindească comportamentul creierului biologic deși până în momentul de față nu s-a găsit nimic care poate fi destul de apropiat.

2.1.3.3 Prezent

Până în prezent, în anumite cazuri, s-a demonstrat că în cazul rețelelor convoluționale folosirea straturilor de normalizare nu a adus un câștig foarte semnificativ datorită felului în care sunt construite. Deși în cazul straturilor complet conectate pe care le vom discuta mai târziu s-a putut demonstra un câștig destul de important.

2.1.4 Tipul Conectat complet

2.1.4.1 Intuitiv

Stratul respectiv se va găsi de obicei la sfârșitul rețelelor neuronale pentru a corela legăturile din trăsăturile extrase și modelele pe care vor să le definească.

Din punct de vedere computațional, stratul convoluțional și cel conectat pot fi schimbate între ele. Datorită conectivității locale din stratul convoluțional, cel conectat va avea în output foarte multe valori de 0 care nu vor imita conectivitatea locală din stratul convoluțional.

2.1.4.2 Analogie

Rolul stratului complet conectat este de acela de a privi (fiecare neuron are legături cu toate trăsăturile stratului precedent) și de a identifica un grup de caracteristici care va duce la activare.

Drept exemplu ele vor modela ultimul rând de neuroni din creieri responsabili ai creierului uman.

2.1.4.3 Prezent

În prezent straturile complet conectate sunt încă folosite, dar depinzând de cazuri vor folosi funcții de activare diferite, drept exemplu, stratul de output va avea același număr de neuroni cu numărul de clase care vor dori a fi clasificate, de exemplu pentru a clasifica din

imagini dacă este vorba de un câine și o pisica, stratul de output va avea doar 2 neuroni complet conectați su stratul dinaintea lui.

Deoarece stratul complet conectat este foarte dispus spre overfitting, se prefera a fi intercalat fie cu straturi de tip dropout fie cu straturi de tip normalizare, depinzând de situație.

2.1.5 Tipul Dropout

2.1.5.1 Intuitiv

Stratul respectiv este folosit în special pentru a evita cat mai mult overfitting-ul produs de stratul de tip conectat complet.

El este reprezentat printr-o probabilitate de a dezactiva un procentaj din activări în mod aleator din stratul precedent în cel antecedent.

El poate fi de mai multe tipuri, dropout spațial sau simplu, în cazul primului tip el este folosit în special pentru stratul convolutiv, el având rolul dezactivării totale a unor matrici de trăsături⁶ și nu a pixelilor respectivi.

2.1.5.2 Analogie

Stratul dropout nu are un analog exact în creierul uman ci ține mai mult de felul în care comunică neuronii între ei. Cea mai apropiată analogie este faptul că unii neuroni din creierul uman, atunci când se activează, nu au o comunicare liniară ci mai mult prin intermediul semnalelor de vârf.

2.1.5.3 Present

În prezent stratul de tip dropout continuă să fie folosit și intercalat cu cel conectat pentru a limita dezavantajul produs de stratul conectat, acela fiind overfitting-ul.

Depinzând de cazuri și de nivelul de zgomot din input, tipul spațial dropout nu mai este neapărat dorit a fi intercalat cu cel convolutiv, justificarea fiind că anumite trăsături pot fi

⁶ "Convolutional Feature Maps - Kaiming He."

http://kaiminghe.com/iccv15tutorial/iccv2015_tutorial_convolutional_feature_maps_kaiminghe.pdf.

Accessed 7 Feb. 2019.

constant reprezentate în aceleași zone spațiale. Spre exemplu, când ne uităm la o mașină, știm unde ar trebui să se afle roțile sau geamurile, lucru care ne ajută în a identifica clasa respectivă, dar tipul spațial dropout poate nega unele mape de trăsături esențiale.

În cadrul acestei lucrări, dacă studiem îndeajuns cum se manifestă cancerul între celule, putem observa că variația este foarte mare de la poziție până la orientare.

În concluzie, pentru această lucrare, se va dori intercalarea cu straturile convolutive pentru a încerca maximizarea identificării unei celule de către mai multe filtre din stratul care analizează imaginea.

2.1.6 Funcții de activare

2.1.6.1 Intuitiv

Funcțiile de activare au rolul de a modela valorile de output dintr-un neuron aflat în rețeaua neuronală. În rețelele neuronale sunt folosite pentru a imita comportamentul unui neuron biologic. În altă ordine de idei, ea va dicta dacă neuronul respectiv se va activa și va propaga informația mai departe sau nu.

În practică, se dorește a fi prezent după straturile de normalizare pentru a modifica greutatea stratului respectiv în proporții cât mai puțin destructive, care să provoace moartea anumitor neuroni din stratul de activare.

2.1.6.2 Analogie

Spre deosebire însă de creierul biologic care are diferite tipuri de activări, de tip spontan, regulat sau rapid, în cadrul rețelelor neuronale aceste tipuri de activări vor fi menținute la un nivel matematic care poate fi modelat.

2.1.6.3 Prezent

În cadrul rețelelor neuronale convenționale, atât unele dintre cele mai dorite funcții de activare cât și cele mai folosite sunt de tip ReLu⁷, Rectified Linear Unit, iar tot comportamentul lor este dat de $\max(0,x)$. Unul din cel mai mari avantaje aduse de această funcție oarecum simplă, este că, în practică, a fost demonstrat că converge foarte repede la minimul local (locul în care se înregistrează cele mai puține pierderi).

Deși are un comportament care duce la o învățare foarte rapidă în rețelele neuronale, un avantaj destul de important, funcția respectivă duce la un proces de învățare foarte rapid. În detrimentul acestui fapt, unii neuroni pot ajunge să “moară”, datorită modificării greutateților în procesul de învățare în așa fel încât output-ul să fie constant pe 0, un aspect ce nu este de dorit.

În unele cazuri, acest proces poate fi atât de avansat încât până la 40% din rețeaua neuronală va ajunge “moartă”, iar nici unul din neuronii respectivi nu se vor activa indiferent de input.

Cu un pas de învățare destul de mic, acest dezavantaj poate fi combătut prin alterarea procesului de învățare, astfel încât să fie cât mai încet, în acest fel, greutatețile input-urilor să fie modificate într-un ritm cât mai mic.

Totuși, folosind această practică, se va ajunge la un număr mai mare de repetări ale antrenamentului pentru rețeaua neuronală în cauză.

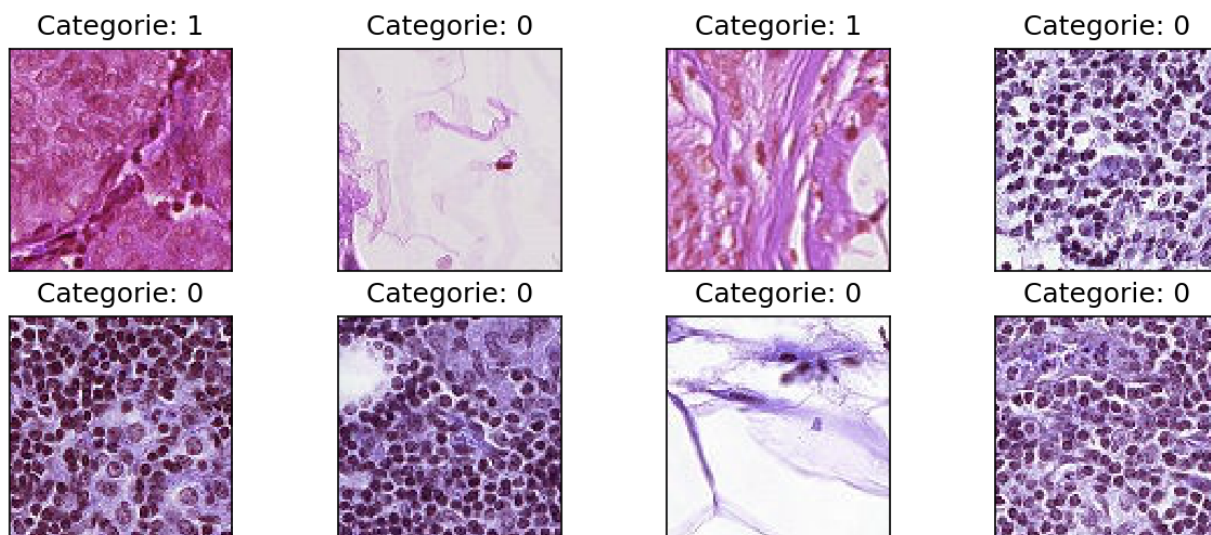
3. Metodologia abordată

Prima etapă consistă într-o scurtă analiză de date pentru a dobândi un comportament imediat, i.e. de la intensitatea pixelilor până la distribuția acestora în imagini, continuând cu gradul de luminozitate.

Pasul următor este constituit atât de observarea aspectului setului de date cât și de valoarea de adevăr anexată imaginii. În consecință, prezența celulelor metastazice este indexată cu 1, iar lipsa acestora cu 0.

⁷ "Rectifier (neural networks) - Wikipedia." [https://en.wikipedia.org/wiki/Rectifier_\(neural_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)). Accessed 7 Feb. 2019.

Dar pentru început va trebui sa vizualizam datele, pentru o mai buna înțelegere a acestora, asa ca putem selecta un eșantion de 8 imagini aleatoare la mărimea originala de 96x96x3 în speranța indentificării anumitor trăsături.



Din păcate la o primă observație, nu putem distinge anumite trăsături între cele două categorii, motiv pentru care vom opta pentru extragerea fie a anumitor statistici, fie a diferențelor dintre cele două categorii.

În următorul stadiu am ales să vizualizăm distribuția datelor în cadrul setului de date.

În cadrul acestei etape, ținta noastră este observarea canalelor de culoare spre a vedea media acestora.

Pentru a nu parcurge de două ori datele, odată pentru statistica per categorie, odată per total, vom aborda calcularea mediei pixelilor pe fiecare canal concomitent.

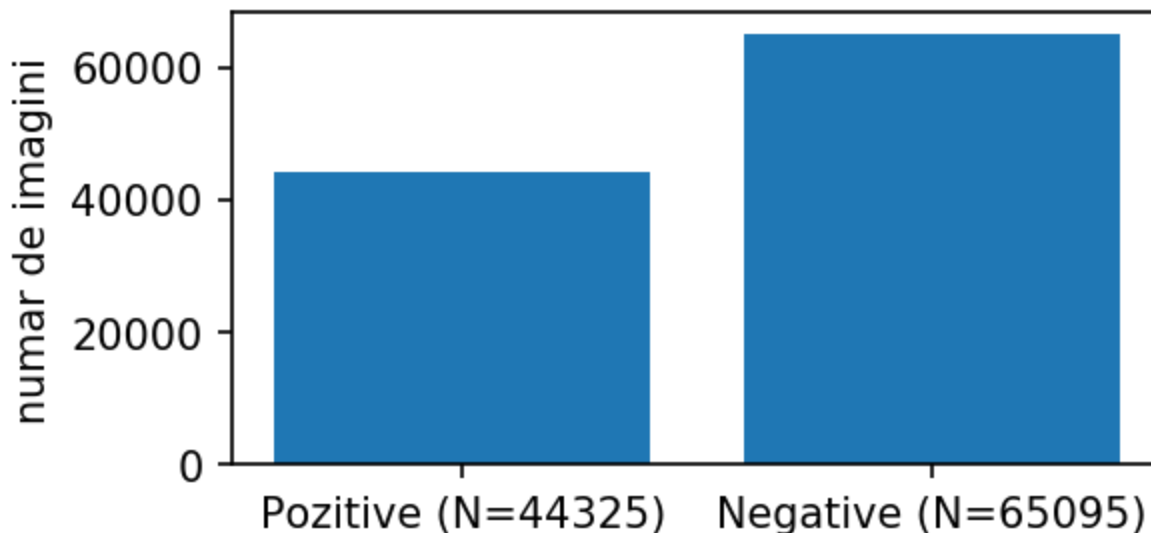
Valoarea obținută va fi folosită mai târziu pentru a face zero centering în jurul valorilor consumate de rețeaua neuronală.

Aceste valori sunt:

- Pentru canalul roșu: 177.36
- Pentru canalul verde: 138.84
- Pentru canalul albastru: 179.0

- Intensitatea pixelilor în canalul gri este: 155.25

În continuare, ne vom uita peste particularitățile categoriilor privind canalele de culoare.



După o primă observație a setului de date, se poate nota o discrepanță între rezultate. În cazul de față, ar fi oportun să avem un set de date cât mai echilibrat, dar având în vedere că nu este posibil fără să ținem cont de riscul de a introduce dubluri, îl vom balansa atribuind greutatea diferitelor categorii pe care vrem să le clasificăm.

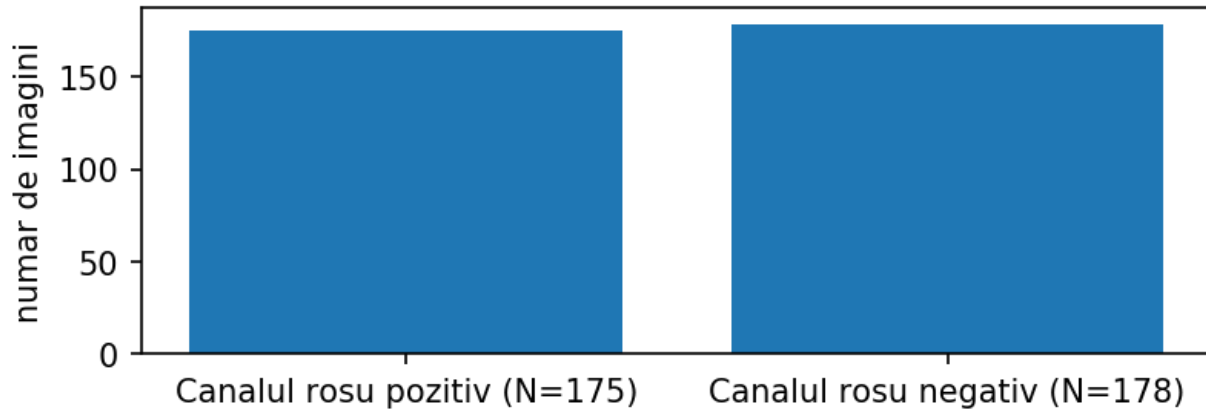
În urma clasificării greșite a unei imagini ilustrând o celulă malignă, aceasta va avea un quantum de aproximativ 1.25 mai mare decât clasificarea greșită a unei poze ce nu prezintă o celulă canceroasă.

Înainte de a începe procesul de analiză a canalelor de culoare și a distribuției acestora în cadrul setului de date, este necesară mențiunea conform căreia se caută celule maligne doar pe partea centrală a imaginii de 32 pe 32 pixeli și 3 canale de culoare.

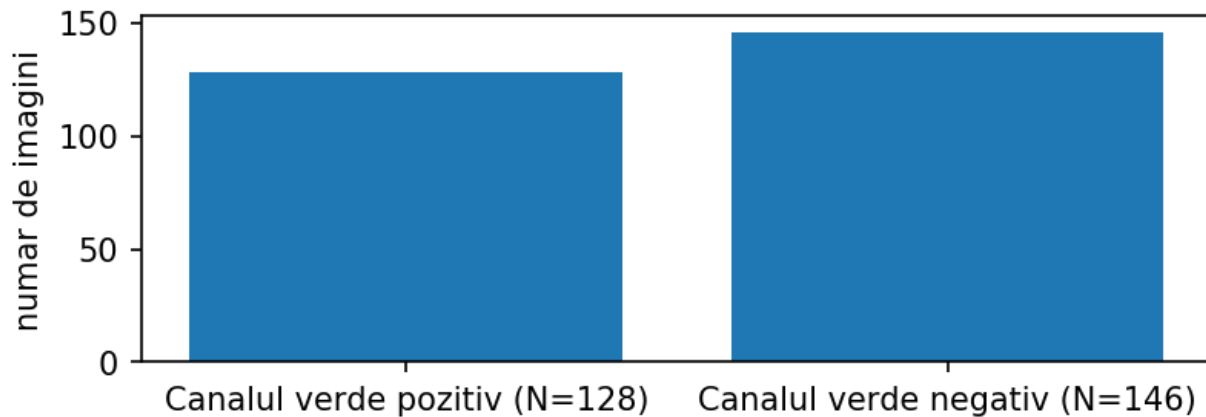
În acest mod, înaintea oricărei analize, vom tăia imaginile astfel încât doar cei 32 de pixeli din centrul imaginii, zona noastră de interes, să fie folosiți în următoarele etape.

Următorul pas este reprezentat atât de observarea distribuției pixelilor cât și de intensitatea acestor în canalul gri. Această etapă ne va conduce atât spre identificarea posibilității de a extrage anumite trăsături din intensitățile lor precum și la distribuția pe canalele de culoare.

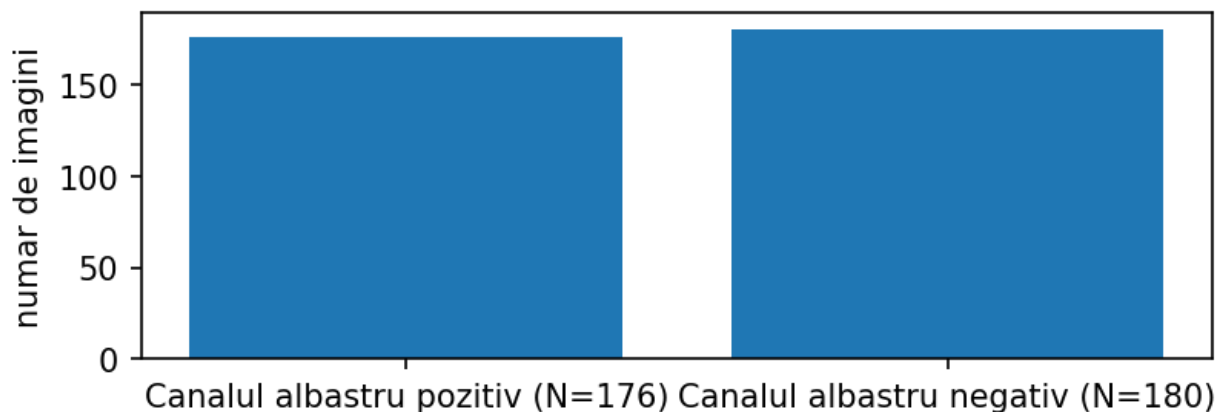
După cum se poate observa, distribuția este în felul următor:



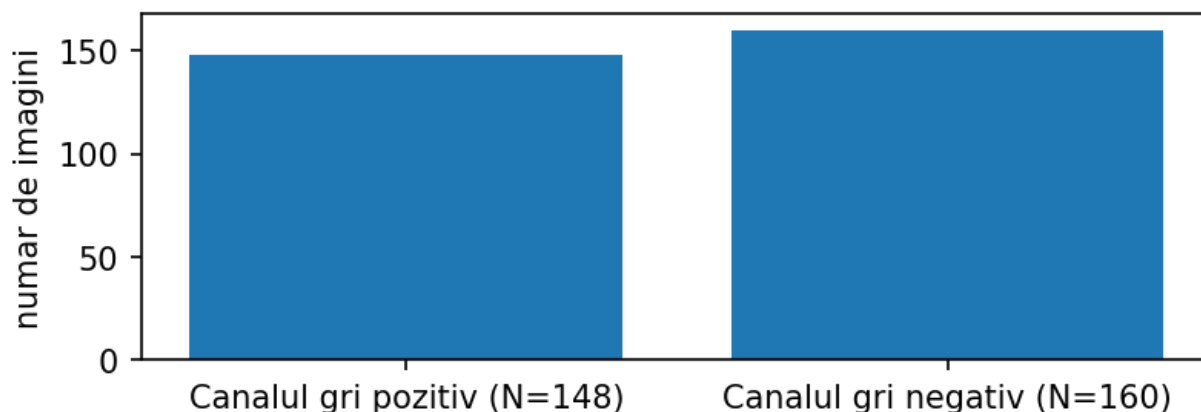
Pentru canalul rosu distribuția pixelilor pare să fie îndeajuns de apropiată de monstrele pozitive și cele negative. Totuși, rețeaua neuronală ar trebui să folosească și acest canal în procesul de decizie având în vedere că nu dispunem de foarte multe informații.



Pentru canalul verde se poate identifica o diferență destul de mare, fapt care ne-ar putea ajuta în clasificarea mult mai eficientă a monstrelor.



Pentru canalul albastru de culoare din imagini este prezentă o situație similară ca în cazul canalului roșu, media pixelilor fiind destul de apropiată.



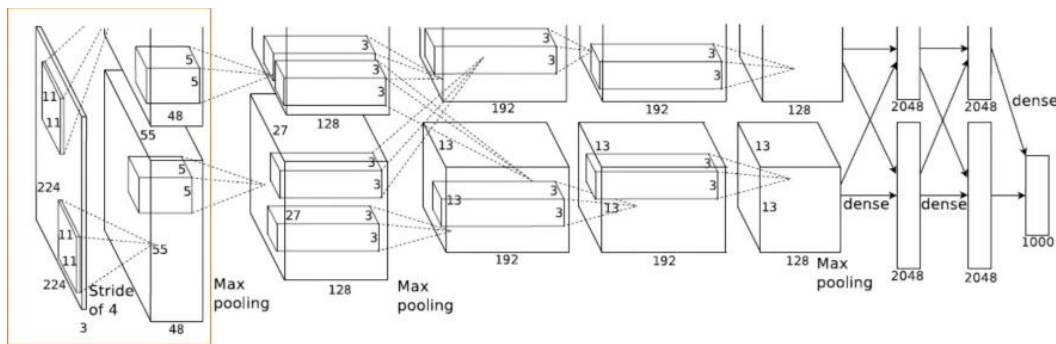
Într-un final, între intensitatea pixelilor din cadrul monstrelor rezultă o discrepanta destul consistentă ceea ce conduce totodată la o diferență între intensitatea pixelilor dintre monstrele negative și cele pozitive.

Merită menționat faptul că în momentul în care am calculat media, ne-am folosit de numărul pixelilor totali din fiecare clasă și nu de numărul pixelilor totali din ambele clase.

Pentru a ne folosi de antrenamentul cu batch-uri cât mai mare, ne putem ajuta de un generator de batch-uri care să amestece valorile între ele după fiecare epocă de învățare (tot timpul se dorește amestecarea valorilor pentru o durată cât mai îndelungată, deoarece atunci când o rețea neuronală va vedea o posibilitate de overfitting, va profita de ea).

4. Modele testate

4.1 Precum AlexNet



Una din arhitecturile de rețele neuronale convolutive care au schimbat percepția, a fost AlexNet⁸, de aceea ca o primă încercare, am utilizat adaptarea unei structuri asemănătoare cu AlexNet la cazul nostru.

Demn de menționat este că, atunci când structura AlexNet a apărut, a fost folosită pentru clasificarea imaginilor 2D, scopul fiind reducere destul de substanțială față de cazul nostru memoria necesară pentru reprezentarea în memorie a rețelei neuronale. Din această cauză, de la numărul de parametri antrenabili în AlexNet (aproximativ 60 de milioane), în cazul nostru imaginile de input nu sunt la fel de mari precum cele folosite de AlexNet, așa că vom rămâne cu mult mai puțini parametri la final.

Modelul folosit ca o primă încercare este următorul:

Construirea a 3 blocuri ascunse de neuroni cu următoarea structură:

Conv->Batch->Relu->Conv->Batch->Relu->Dropout->Pooling

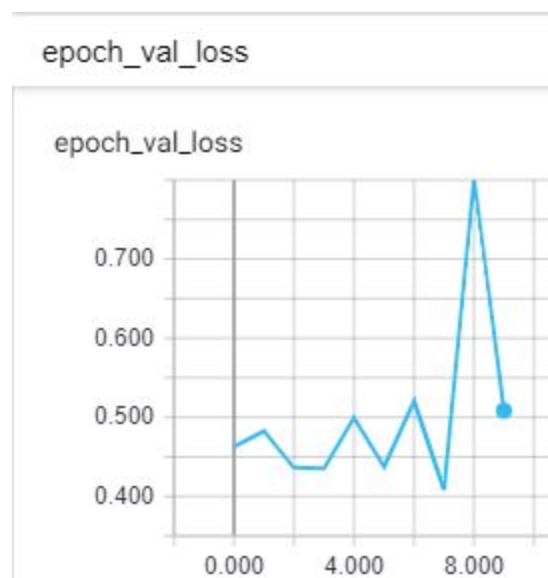
De reținut este că această structură s-a folosit doar de canalele Roșu, Verde, Albastru, din imagini și nu a fost luată în calcul și intensitatea pixelilor.

⁸ "AlexNet - Stanford Vision Lab." 18 May. 2015, http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf. Accessed 7 Feb. 2019.

Fiecare filtru de convoluție va fi de 3x3 pe parcursul întregii rețele, astfel vom observa cu timpul că folosirea diferită a mărimilor filtrelor pe parcursul rețelei neuronale poate aduce foarte mari îmbunătățiri.

Ca idee principală se preferă filtrele cât mai mici, deoarece este demonstrat că generalizează mult mai bine față de filtrele mari care tind să se ducă spre overfitting destul de repede.

O structură foarte simplă de rețea neuronală a ajuns după doar 8 repetări la o pierdere de validate de 0.43 și o precizie de 81.88% în cadrul validării.



Ceea ce trebuie însă menționat neapărat este faptul că după a 8 repetare, rețeaua neuronală are un spike foarte mare în pierderea la validare.

Cu toate că la a 10-a repetare pare să se normalizeze, modelul respectiv nu ar trebui învățat mai mult de 8 repetări, spike-urile destul de avansate în antrenament denotă de cele mai multe ori overfitting-ul clasificatorului.

Vom păstra structura generală a nodurilor ascunse în dezvoltarea structurilor ulterioare, variind mărimea filtrelor sau numărul lor pe parcurs ce avansăm.

Un lucru notabil este că în următoarele iterații ale clasificatorului se va renunța la stratul de tip pooling pentru a păstra cât mai mult din trăsăturile extrase.

Având în vedere și dimensiunile mici nu este neapărată nevoie de Pooling, care are ca rol principal micșorarea dimensiunilor propagate în rețea.

4.2 Folosirea grayscale

În aceasta arhitectură am introdus un al 4-lea canal de culoare, inițial pe aceeași arhitectură folosită anterior, fără nici un fel de modificare asupra structurii dinainte, cu excepția numărului de canale de culoare din 3 în 4.

Cu așteptarea că unele filtre vor extrage trăsături, muchii și cercuri din canalul gri (lucru care din păcate nu s-a întâmplat), comportamentul rețelei neuronale pe parcursul învățării tindea spre metodologia “pune tot pe roșu”, astfel alegând o singură categorie necontestantă de input.

Comportament destul de neobișnuit având în vedere că grayscale-ul pare să ne transmită destul de multe informații despre categoriile noastre.

Pe aceeași structură am încercat atât varierea numărului de filtre și mărimii lor cât și ajustarea șansei de dropout, însă nimic nu a adus un alt comportament al clasificatorului.

4.3 Rețeaua neuronală proprie grayscale

După cum spune și titlul am împărțit rețeaua neuronală în 2 părți, una responsabilă de canalele Rosu, Verde, Albastru, și încă una care să se descurce strict cu canalul gri.

Pentru aceasta abordare și a uni rezultatele din ambele părți, trebuie să ne asigurăm că până la finalul lor, vom ajunge la aceleași mărimi (număr de matrice de trăsături, înălțime, lățime) singurul lucru care poate fi variat va fi adâncimea, în cazul nostru canalele de culoare.

Am ales această abordare deoarece pe canalul gri avem foarte multe șanse de a detecta muchii și alte formațiuni care ne pot spune anumite informații despre tipul celulei.

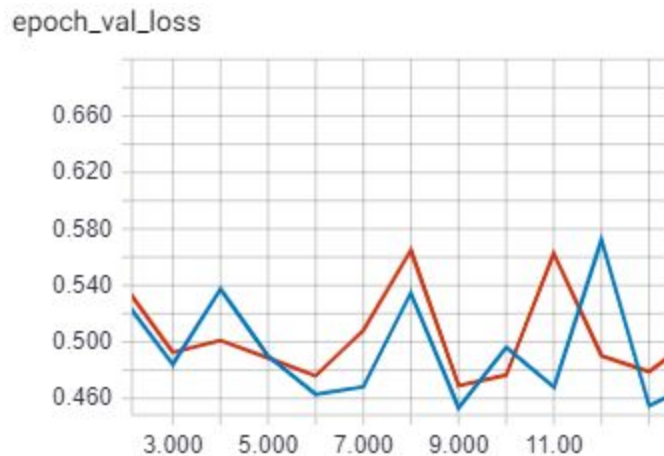
Arhitectura celor două părți ale rețelei este identică, doar o mică diferență poate fi identificată și anume, numărul de canale de culoare.

Astfel la finalul celor două rețele, luăm matricile rezultate și le unim pentru a fi procesate în continuare de un strat complet conectat. În așa fel încât să extragă atât informații despre muchii cât și despre culoare. Faptul că ar trebui să mărească șansele de clasificare.

După cum s-a dovedit, această abordare a dus la procesul de învățare în cadrul rețelei neuronale, indicată de o acuratețe la validare de 79.23% și o pierdere la validare de numai 0.46.

Totuși, rezultatele comparativ cu prima încercare nu s-au îmbunătățit, astfel structura minimală folosită prima a obținut rezultate mai bune decât cea actuală.

De asemenea, constituie un mic avantaj această abordare, întrucât doar după 7 repetări de antrenament s-a ajuns la acest rezultat.



4.4 Rețea neuronală proprie grayscale - filtre diferite

De departe, cel mai de succes model testat este bazat pe arhitectura anterioară, cu mici modificări făcute în numărul blocurilor de straturi ascunse și mărimea filtrelor.

Urmând modelul AlexNet, am făcut în așa fel încât să variez mărimea filtrelor inițiale din primele două blocuri de neuroni, și am micșorat mărimea în ultimele două, aceasta a adus de departe cea mai mare îmbunătățire a preciziei și o pierdere pe setul de date de validare, la o valoare de 0.43 crossentropie binară la doar 15 repetări.

Având în vedere că mărimea rețelei este destul de mare cu o probabilitate de dropout ridicată, 0.5 pentru straturile conexe și 0.3 pentru cele convolutive, plus tendința micșorării constante a pierderii după a 12-a repetare, acest model ar putea trece prin 20+ repetări fără manifestarea unui overfitment agresiv.

Pentru vizualizare avem aici evoluția pierderii la validare:



O mică modificare care mai poate fi adusă rețelei obținute ar fi mutarea straturilor complet conectate înainte de a uni cele două rețele, care în practică a dovedit că reduce pierderea la validare cu încă 0.02, rezultând la o acuratețe de 81.27% și o pierdere de numai 0.42.



Ca extensie, atunci când introducem imaginile în rețea, am putea face și o rotire aleatorie la stânga sau dreapta, fapt care nu ne va afecta rețeaua, având în vedere că imaginile sunt reprezentate prin intermediul unor matrici pătratice de 32x32.

Acest fapt ar putea limita și mai mult overfitting-ul prin variația datelor și pozițiile spațiale ale celulelor, fapt care este foarte benefic.

În acest fel, am ales un factor de rotație aleator spre stânga pentru pozele care vor fi consumate de rețeaua neuronală între 0 și 3, factorul va rămâne comun în rotația imaginilor atât peste grayscale cât și peste canalele de culoare Roșu, Verde, Albastru.

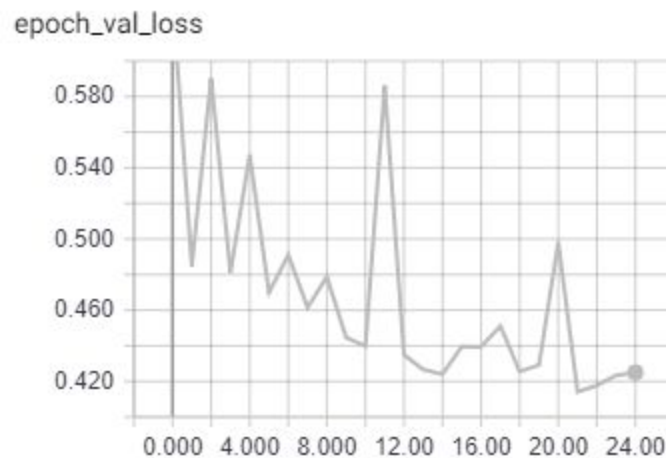
Deoarece dorim să păstrăm atât trăsăturile spațiale extrase cât și cele de culoare, această procedură este eșantionarea datelor și este posibilă, întrucât celulele maligne nu sunt dependent spațiale, contează cum a fost așezat țesutul sub microscop.

Datorată acestui fapt, dataset-ul nostru poate fi mărit de 4 ori, depinzând de rotațiile aleatoare.

Este preferabil ca numărul de rotație să fie aleator deoarece rețeaua neuronală ar putea să înceapă să intuiască anumite pattern-uri care nu sunt reale dacă rotațiile se fac crescător, fapt care nu este de dorit.

Pentru un rezultat final, în cazul în care introducem și rotații aleatoare ale imaginilor cu un număr de 25 de repetări, am ajuns la o acuratețe la validare de 81.31% și o pierdere de numai 0.41.

Deși acuratețea la validare este puțin mai mică față de prima încercare, ne dorim mai mult un grad de generalizare mai ridicat, indicat de pierderi.



5. Contribuții:

Spre deosebire de folosirea unei arhitecturi clasice, am ales să adaptez arhitectura AlexNet, deja demonstrată ca fiind superioară altor arhitecturi pentru clasificarea imaginilor 2D.

Împărțirea inputului în 2 părți și antrenarea a două rețele neuronale în paralel una fiind responsabilă de detectarea muchiilor, în cazul de față pereții celulari, iar cealaltă rețea să fie responsabilă doar de tonurile de culoare observate în imagine.

Un avantaj major al acestei structuri este faptul că neuronii din stratul complet conectat se balansează tot timpul între culori și forme, care duc la un model foarte robust din acest punct de vedere.

După cum se poate observa în antrenarea variantei finale de rețea neuronală, se dovedește a fi îndeajuns de lipsită de overfitting, având în vedere ca după repetarea a 10-a acuratețea la validare este mai mare decât cea la antrenare și pierderea este invers proporțională, lucru de dorit în elaborarea unui model.

6. Îmbunătățiri:

Deși imaginea celulelor oferă o cantitate destul de mare de informație pentru diagnosticarea lor, anumite date mai pot fi extrase înainte de a fi analizate, cum ar fi proveniența lor, tipul de țesut din care sunt extrase precum și un tabel de adevăr pentru pacient cu anumite informații, predispoziția la cancer ș.a.m.d.

Dacă data set-ul ar conține și alte metadate, algoritmul de clasificare ar putea fi modificat foarte ușor pentru a lua în considerare și aceste trăsături, fapt care ar conduce la un clasificator mult mai puternic.

O analiză pe alte trăsături ale imaginilor ar putea aduce rezultate și mai bune, poate în primă fază o normalizare a pixelilor, după calcularea zero a centering-ului.

Deoarece pentru cazul de față nu s-a făcut normalizare, din cauza efortului computațional mare, am ales doar să centram pixeli în jurul valorii de 0. Astfel, ar putea fi observat și

comportamentul rețelei neuronale în cazul în care normalizăm după ce calculăm valorile pixelilor, am putea reuși să extragem niște trăsături mult mai puternice în acest caz.

7. Bibliografie

<https://www.tensorflow.org/tutorials>

<https://keras.io/>

“Basic Operations on Images”, *OpenCV-Python Tutorials*, platformă online cu diferite tutoriale,

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_core/py_basic_ops/py_basic_ops.html#additional-resources , s.d., 3.02.2019

“Arithmetic Operations on Images”, *OpenCV-Python Tutorials*, platformă online cu diferite tutoriale,

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_core/py_image_arithmetics/py_image_arithmetics.html#image-arithmetics, s.d., 3.02.2019

<https://docs.anaconda.com/anaconda/user-guide/>

“The Jupyter Notebook”, *Jupyter Notebook*, platformă online,

<https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
<http://cs231n.github.io/neural-networks-2/>, s.d., 2.02.2019

KRIZHEVSKY Alex et al., “ImageNet Classification with Deep Convolutional Neural Networks”, *Advances in Neural Information Processing Systems 25*,

<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>, 2012, 2.02.2019, pp. 1-9

BHARATH Raj, “A Simple Guide to the Versions of the Inception Network”, *Towards Data Science*,

<https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202>, 29.05.2018, 2.02.2019

<https://docs.scipy.org/doc/numpy-1.13.0/reference/> -> ndarray, ndtyp, rot90, linalg

“API overview”, Matplotlib, https://matplotlib.org/api/api_overview.html, 2.02.2019

REZAUL Karim et al., *“Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python”*, Birmingham, Packt Publishing Ltd., 2018

SUMIT Saha, “A comprehensive guide to convolutional neural network - the ELI5 way”,
”, *Towards Data Science*, <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> 15.12.2018, 20.01.2019

“Machine Learning Yearning” Andrew Ng

“Machine Learning” Tom M. Mitchell

“AlexNet” Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton