MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL INFORMATICĂ APLICATĂ

APLICAŢIE PENTRU DETECŢIA BOLILOR PULMONARE

LUCRARE DE LICENŢĂ

Absolvent: **Mihai COZMUȚA**
Conducător ştiinţific: **Asist. drd. ing. Vlad MIHALY**

2021

MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL INFORMATICĂ APLICATĂ

DECAN,
Prof. dr. ing. Liviu MICLEA

DIRECTOR DEPARTAMENT,
Prof. dr. ing. Honoriu VĂLEAN

Absolvent: **Mihai COZMUȚA**

APLICAŢIE PENTRU DETECŢIA BOLILOR PULMONARE

1. **Enunţul temei:** *Lucrarea de faţă propune o aplicaţie pentru clasificarea imaginilor radiografice în două clase distincte: sănătos sau bolnav. În realizarea soluţiei s-au folosit diverşi algoritmi pentru procesarea imaginilor, extragerea trăsăturilor şi clasificarea sau diagnosticului final.*
2. **Conţinutul lucrării:** *Introducere, Obiectivele proiectului, Studiu Bibliografic, Analiză şi Fundamentare Teoretică, Proiectare de detaliu şi Implementare, Concluzii, Bibliografie*
3. **Locul documentării:** Universitatea Tehnică din Cluj-Napoca, Departamentul Automatică
4. **Data emiterii temei:** 1 Noiembrie 2020
5. **Data predării:** 8 Iulie 2021

Absolvent: Mihai Cozmuţa

Coordonator ştiinţific: Asist. drd. ing. Vlad Mihaly

MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL INFORMATICĂ APLICATĂ

Declaraţie pe proprie răspundere privind autenticitatea lucrării de licenţă

Subsemnatul(a) Cozmuța Mihai, legitimat(ă) cu C.I. MM 780330, CNP 1970116245026, autorul lucrării APLICAŢIE PENTRU DETECŢIA BOLILOR PULMONARE, elaborată în vederea susţinerii examenului de finalizare a studiilor de licenţă la Facultatea de Automatică şi Calculatoare, Specializarea Automatică şi Informatică Aplicată din cadrul Universităţii Tehnice din Cluj-Napoca, sesiunea Iulie a anului universitar 2020-2021, declar pe proprie răspundere, că această lucrare este rezultatul propriei activităţi intelectuale, pe baza cercetărilor mele şi pe baza informaţiilor obţinute din surse care au fost citate, în textul lucrării şi în bibliografie.

Declar, că această lucrare nu conţine porţiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislaţiei române şi a convenţiilor internaţionale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în faţa unei alte comisii de examen de licenţă.

În cazul constatării ulterioare a unor declaraţii false, voi suporta sancţiunile administrative, respectiv, *anularea examenului de licenţă*.

Data	Nume, Prenume
08.07.2021	Cozmuţa, Mihai

Semnătura

Cuprins

Capitolul 1

Introducere - Contextul proiectului

1.1 Contextul proiectului

Pneumonia este una dintre cauzele principale de deces ale copiilor şi unul dintre cei mai mari factori ai creşterii mortalităţii mondiale. Această boală este responsabilă pentru 16 procente din decesele copiilor sub vărsta de 5 ani, fiind astfel cauza principală a mortalităţii în cazul copiilor. De exemplu, doar în Statele Unite, un milion de adulţi sunt spitalizaţi în fiecare an din cauza pneumoniei, dintre care peste 50 000 se sting ca urmare a bolii [1]. Recenta criză cauzată de COVID-19 este o ameninţare cu viaţa pentru mii de oameni in anul 2020. Pneumonia secondată de virusul SARS-CoV-2 este o ingrijorare globală, având confirmate cazuri în peste 185 de ţări la momentul redactării acestei lucrări [2].

Totuşi o diagnosticare precoce, urmată de un tratament de specialitate determină o şansă mare de vindecare in rândul bolnavilor. Diagnosticul precoce a pneumoniei distructive permite iniţierea tratamentului şi reducerea evoluţiei nefavorabile a proceselor supurative, care pot duce la invaliditate şi deces.

Diagnosticarea

Radiografiile ($X\text{-}RAY$) sunt metoda cea mai bună pentru a incepe consultarea unui pacient suspect de pneumonie. Diagnosticarea pneumoniei se face prin analiza radiografiilor făcute la nivelul pieptului de către medici specializţi în acest domeniu. Aceasta se manifestă prin prezenţa unor zone opace in radiografie, diagnoza fiind confirmată prin investigarea istoricului medical al pacientului, urmată de analize mai amănunţite de laborator. Diagnosticarea doar prin inspectarea radiografiilor este dificilă din cazua unor alte boli prezente la nivelul plămănilor (pierderea in volum, săngerari, schimbari cauzate de radiografii precedente sau intervenţii chirurgicale). Când se poate, compararea unor radiografii realizate la momente diferite în timp si corelarea acestora cu simptomele suferite este un ajutor pentru medici în luarea deciziei. Un alt factor care influenţează semnificativ diagnosticul specialiştilor este şi modul în care pacientul este poziţionat în timpul radio-

grafiei, precum şi felul acestuia de a inspira aerul. Aceste lucruri pot complica sarcina medicilor de a oferi un diagnostic corect.

Factori de risc

Persoanele cu cel mai mare factor de risc sunt copiii care au vârsta de 2 ani sau mai mici si adulţii care au varsta de 65 de ani sau peste. Printre cei mai recunoscuţi factori de risc se număra:

- spitalizarea (folosirea ventilatoarelor mecanice din camerele de terapie intensivă);
- bolile cronice cum ar fi astmul sau alte probleme cardiace;
- un sistem imunitar slăbit (persoanele care au SIDA sau persoanele care au avut un transplant de organ).

1.2 Motivația proiectului

Există o semnificativă divseritate a interpretărilor radiografice în randul medicilor. Pentru a imbunătăţii eficienţa şi acurateţea diagnosticului, sistemele asistate de calculator pentru ajutorul diagnosticării au fost exploatate intens în ultimul deceniu. Detecţia de către medici necesită timp îndelugant şi este nevoie de multă experienţă şi pregatire specializată. De preferat ar fi ca acest proces sa fie unul cât mai rapid şi sigur, identificând doua grupuri: cei fară probleme si cei la care exista şansa de a avea pneumonie, urmând ca cea de-a doua categorie să treacă mai departe la analize mai amanunţite pentru a stabili cu exactitate prezenţa sau absenţa pneumoniei. Acest lucru ar minimiza semnificativ resursele medicale, timpul petrecut de medici analizând radiografiile şi, fiind un proces automat, identificarea s-ar putea face chiar de către persoana care face radiografiile.

Proiectul de faţă propune o soluţie automată pentru detecţia şi clasificarea radiografiilor pulmonare. Metodele folosite pentru identificarea zonelor problematice precum şi extragerea trăsăturilor sunt metode proprii care vin ca o îmbunătăţire la soluţiile deja existente din acest domeniu. Algoritmii şi metodele propuse se bazează pe tehnici noi de procesare a imaginilor precum reţelele neuronale convoluţionale.

Capitolul 2

Obiectivele Proiectului

2.1 Obiective

Ţinta acesteri lucrări se doreşte a fi realizarea unui sistem automat de clasificare a radiografiilor pulmonare. În procesul realizării acestui lucru, se definesc urmatoarele obiective:

1. Importarea şi pre-procesarea datelor. Aplicarea unor corecţii asupra radiografiilor pentru a avea un standard în ceea ce le priveşte (aceeaşi dimensiune, luminozitate etc.). Imaginile furnizate sunt de asemenea insuficiente ca şi număr şi nebalansate din punct de vedere al categoriilor, lucru care trebuie avut in vedere înaintea începerii construirii reţelei neuronale.

2. Extragerea caracterisiticilor (*Feature extraction*). Caracteristicile vor fi extrase cu ajutorul unei reţele neuronale convoluţionale pe care o vom folosi si pentru clasificare.

3. Clasificarea propriu-zisă bazată pe rezultatele de la punctul *2*. Clasificarea folosită va fi una binară: pacient sănatos sau pacient bolnav. In cel din urma caz, vor fi necesare investigaţii ulterioare.

2.2 Specificaţii

2.2.1 Procesarea datelor

Primul pas în procesarea datelor (n.r. radiografii pulmonare) este aplicarea unei corecţii Gamma care constă în reglarea luminozităţii unei imagini, mai precis a intensităţii luminoase [2.1].

Pixelii care formează o imagine iau valori intre 0 si 255. Urmatoarea operaţiune aplicată a fost normalizarea valorilor pixelilor din intervalul [0-255] în intervalul [0-1] pentru a imbunatăţii performantele reţelei.

Figura 2.1: Rezultatele aplicării unei corecţii Gamma. Sursa: [3]

Ultimul pas este reprezentat de augmentarea datelor. Acest proces este necesar deoarece setul de date poate fi nebalansat (mai multe radiografii ale pacienţilor nesănatoşi decât ale pacientilor sănatoşi), lucru care afectează negativ procesul de învatare al reţelei neuronale. În cadrul acestei etape se aplică alterări ale imaginilor (decupări, rotiri, *zoom*) pentru a obţine mai multe date şi a avea un set cât mai echilibrat.

2.2.2 Extragerea Caracteristicilor

Spre îndeplinirea acestui obiectiv se folosesc Reţelele Neuronale Covoluţionale (*CNN*) sunt o sub clasă a reţelelor neuronale. Acestea sunt formate din câteva straturi (*layers*): *input, output* şi straturile ascunse dintre ele. Acestea din urma fac cea mai multă treaba din punctul de vedere al calculelor. În extragerea caracteristicilor, stratul convoluţional(*convolutional layer*) este cel mai important: se axează pe caracteristicile mărunte (cu ajutorul unor filtre) şi le agrega în caracteristici de nivel înalt in următorul layer [4].

Un alt element care are un rol important este aşa-numitul *pooling layer*. Scopul principal al acestora (vor fi mai multe în cadrul unui CNN) este reducerea mărimii imaginii fără a pierde informaţii relevante, cu aceasta reducand numarul de operaţii şi memoria utilizată. Acest proces scade şi numărul parametrilor, eliminand astfel riscul de învăţare greşită al algoritmului.

2.2.3 Clasificarea Radiografiilor

Pentru clasificare, ultimul pas din această problemă, Reţeaua Neuronală Convoluţională vine în ajutor printr-un parametru important şi anume funcţia de activare (*activation function*). Funcţia de activare primeşte ca şi intrare ieşirile neuronilor layer-ului anterior, rezultatul ei fiind primit ca şi input de către neuronii următorului strat. Prin urmare, vom avea mai multe funcţii de activare în reţeaua noastră.

Clasificarea propriu zisă o vom face folosind funcţia de activare Sigmoid [2.2], fiind cea mai buna în metode de clasificare. Indiferent de valorile pe care funcţia le primeşte de la reţeaua neuronală, ieşirea acesteia (şi implicit a algoritmului nostru) va fi o valoare subunitară ([0-1]), reprezentand probabilitatea unui anumit rezultat.

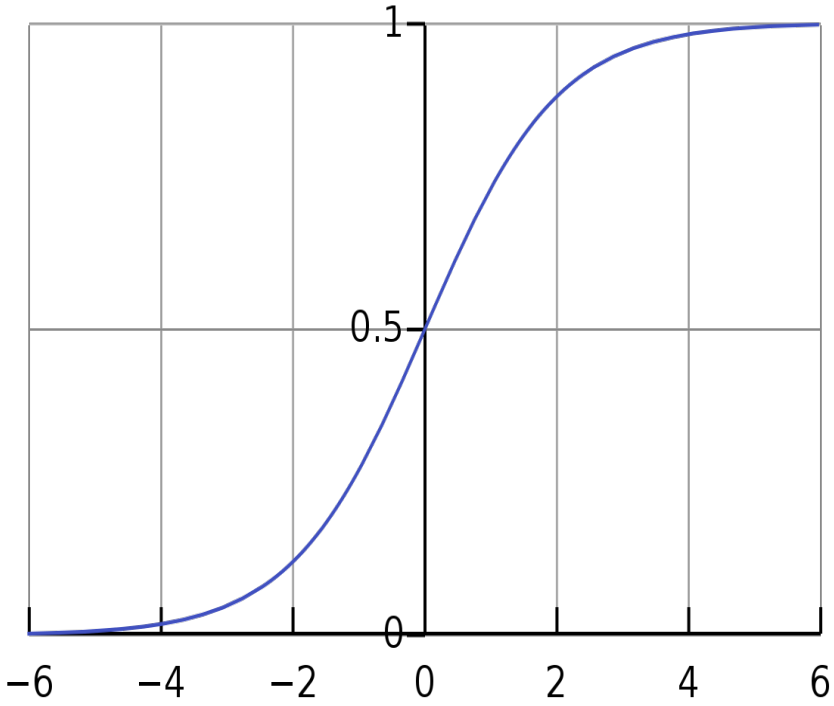

Figura 2.2: Funcţia Sigmoid

În figura 2.3 am realizat arhitectura completă a aplicaţiei (de la intrare la ieşire) conform celor enumerate şi parţial descrise în această secţiune. Se poate observa că principala componentă în alcătuirea soluţiei este blocul CNN (*Convolutional Neural Network*), bloc ce conţine mai multe sub-module care sunt de fapt layerele care compun reţeaua neuronală.

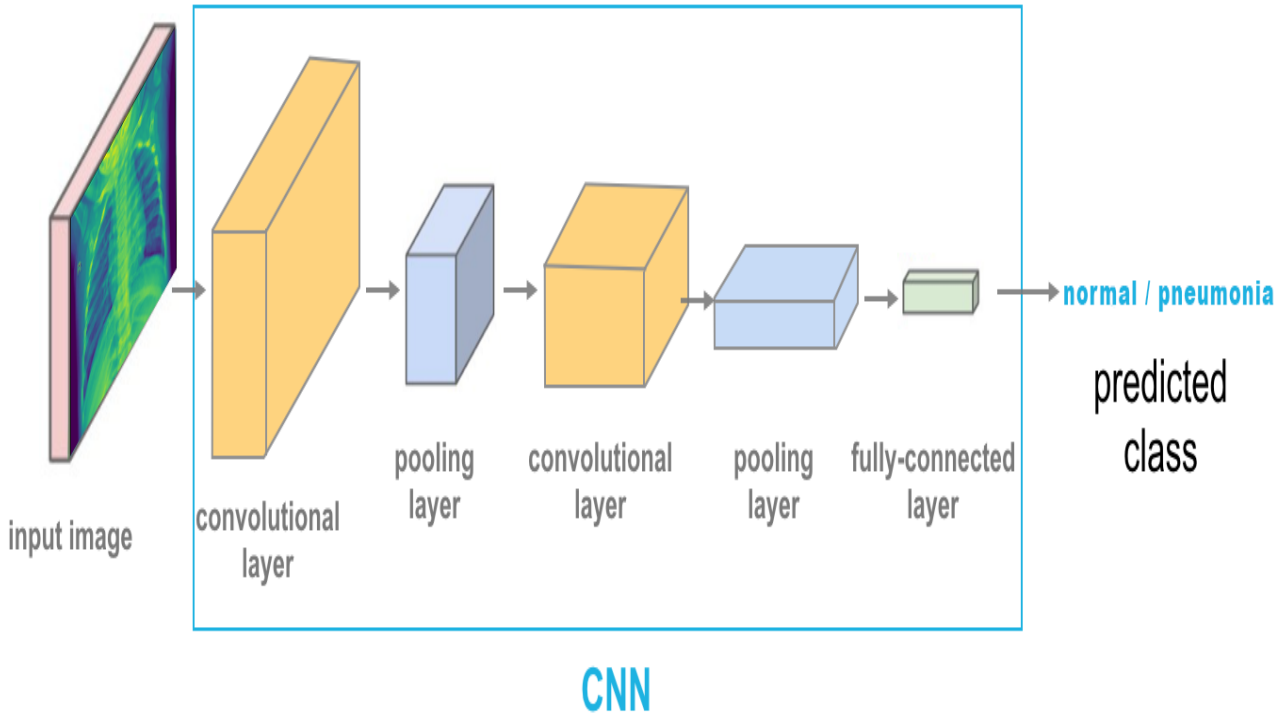

Figura 2.3: Arhitectura aplicaţiei

Se pot observa, în figura de mai sus (2.3), blocurile principale care compun soluţia prezentată în acest proiect. Fiecare dintre aceste componente, începand de la imaginea plânilor şi terminand cu componenta de prezicere finală, vor fi analizate în amănunt în cadrul capitolelor următoare. Acest model al arhitecturii unei soluţii pentru prezicerea clasei unei imagini este printre cele preferate de autorii studiilor în domeniu [5], [6].

Pentru o completare a figurii 2.3, trebuie menţionat faptul ca imaginea de input care ajunge sa fie transmisă reţelei neuronale va suferi diferite transformări (ajustări ale mărimii, culorii etc.) care vor facilita o prezicere cu un grad cât mai ridicat de acurateţe, care este şi scopul unei lucrări din domeniul medical.

Obiectivul final (și cel mai important) din cadrul proiectului este o predicţie cât mai aproape de realitate care să reprezinte un punct de pornire pentru dezvoltarea ulterioară a soluţiei de faţă, cum ar fi intregrarea într-un sistem medical mai amplu care să conţină informaţii relevante despre pacienţi.

Capitolul 3

Studiu Bibliografic

Clasificarea radiografiilor şi detecţia anormalităţilor regăsite în acestea este o problema comună în zilele noastre. Se află o mulţime de algoritmi şi soluţii care incearcă sa rezolve aceasta problemă. Mai jos în acest capitol voi prezenta cele mai folosite şi studiate abordări care există pe această nişă a procesărilor de imagini în scop medical. Aceste abordări au ajutat ca şi puncte de pornie pentru soluţiile propuse in acest proiect.

3.1 Pre-procesarea imaginilor

Unul dintre cei mai importanţi paşi, cand vine vorba de seturi de date care conţin imagini, este pre-procesarea acestora. Un set de date care nu este potrivit duce la rezultate incoerente, deci inutilizabile. În [7], autorul prezintă mai multe metode de augmentare şi standardizare a imaginilor avand ca şi scop îmbunătăţirea performanţelor Reţelelor Neuronale:

- Corecţia Gamma
- Normalizarea şi redmiensionarea după acelaşi standard al imaginilor
- Îmbunătăţirea setului de date

3.1.1 Corecţia Gamma

Prin Corecţia Gamma (3.1) se înţelege modificarea intensităţii luminoase a unei imagini şi reprezintă una dintre cele consacrate tehnici din domeniul procesărilor de imagini [8]. Transformarea Gamma este definită prin urmatoarea relaţie:

$$s = c * r^{\gamma} \tag{3.1}$$

unde c şi r sunte nişte constante pozitive.

Pentru un nivel de gri normal (valoarea pixelilor în intervalul [0-255]):

- dacă $\gamma > 1$, atunci se obţin imagini cu un nivel al luminozităţii ridicat ($r > r^{\gamma}$);
- dacă $\gamma < 1$, atunci imaginile sunt mai întunecate ($r < r^{\gamma}$).

Rolul constantei c este de a rafina luminozitatea obţinută prin Corecţia Gamma definită mai sus.

În unele cazuri este considerată varianta normalizată a imaginilor (valoarea pixelilor aparţinand intervalului [0-1]) despre care vom discuta in secţiunea următoare. In situaţiile sus-numite, efectul transformării este invers:

- $\gamma < 1$ imaginea corectată este mai luminoasă decât cea originala;
- $\gamma > 1$ imaginea corectată este mai întunecată decât cea originala.

Un exemplu al aplicării acestei Transformate Gamma se regăseşte în capitolul anterior, figura 2.1.

3.1.2 Normalizarea si redmiensionarea după acelaşi standard al imaginilor

Problema normalizării setului de date de intrare se ridică atunci cand datele (în cazul nostru, imaginilie) au un mare domeniu de valori. Descris foarte bine de R.C. *Gonzalez* în [7], aceasta este o etapă importantă şi obligatorie aproape de fiecare data cand lucrăm cu seturi mari de date.

Fiind nişte radiografii făcute prin intermediul unor instrumente medicale folosite de tehnicieni radiologi (influenţa factorului uman), acesta din urmă corelată cu faptul că nu toate radiografiile au fost realizate folosind acelaşi aparat, duce la o neregularitate în dimensiunea imaginilor. Acest aspect este evidenţiat în figura [], unde putem regăsi un exemplu din setul de date folosit în realizarea proiectului împreună cu eticheta aferentă fiecărei radiografii, observându-se discrepanţa şi lipsa unui standard în ceea ce priveşte dimensiunea imaginilor.

În [9], autorii afirmă că una dintre cele mai comune metode de normalizare este o scalare liniară a datelor de intrare. În cazul procesării imaginilor, acest lucru se referă la remaparea valorilor pixelilor din intervalul lor obişnuit, [0-255], în intervalul [0-1].

Tot în scopul îmbunătăţirii performanţei se propune standardizarea imaginilor, acestea trebuind redimensionate şi preferabil trecute printr-un filtru *gray-scale* (vezi figura 3.1) [10], [7].

Soluţia de proecasre a imaginilor propusă de autori în [6] pune în evidenţă mai multe operaţii care se pot realiza în scopul obţinerii unor rezultate care să satisfacă obiectivele propuse, însă argumentul adus de Rafael Gonzalez şi Richard Woods în lucrarea [7] este în favoarea procesărilor simple (redimensionare, aplicarea gray-scale etc.) dar cu un impact important asupra rezultatului final.

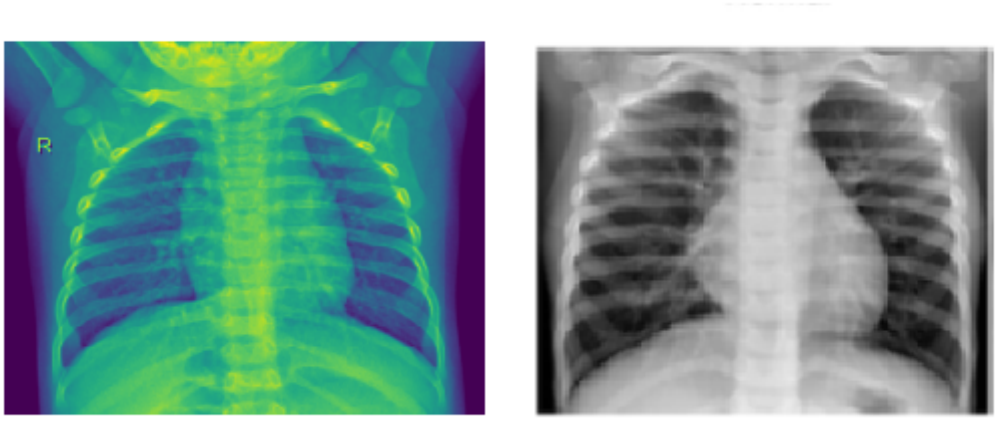

Figura 3.1: Aplicarea filtrului *gray-scale*

3.2 Îmbunătăţirea setului de date existent

Un alt aspect important, care de asemenea îmbunătăţeste semnificativ rezultatele şi elimină riscul incapacităţii Reţelei Neuronale Convoluţionale de a generaliza şi a nu învata pe derost, este lipsa echilibrului în setul de input [7]. Mai exact, imaginile etichetate ca aparţinand clasei A sunt în numar mult mai mic decât imaginile clasei B, 3.2.

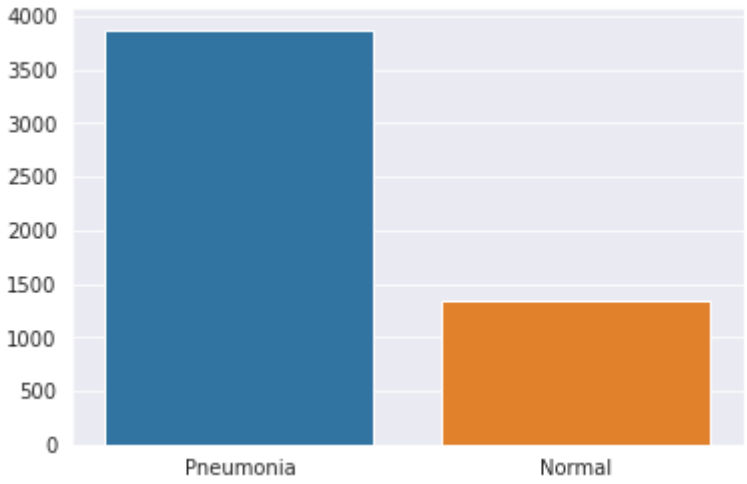

Figura 3.2: Exemplu de data-set nebalansat

Aceasta este o problema generala cand se aduce in discutie orice problema care necesita invatarea din date pentru a produce o clasificare, dar aceasta este mult mai serioasa

cand vine vorba de domeniul medical. În [11], autorii descriu pe larg probleme majore care apar din cauza unui set de date care favorizeaza una dintre clase in domeniul bancar sau medical, acestea fiind cele mai predispuse la erori semnificative.

A.Geron [12] şi Gonzalez et al. [7] propun mai multe metode pentru augmentarea datelor, fiind în majoritatea circumstanţelor o muncă laborioasă pentru a obţine mai multe date aparţinand unei anumite clase. Dintre acestea enumerăm:

- decupare;
- mărire (efectul de zoom-in);
- rotire.

Toate operaţiile enumerate mai sus se aplică imaginilor deja existene. Acestea se realizează folosind metodele puse la dispoziţie in librăria Keras [12], care oferă algoritmii potriviti pentru aplicarea operaţiilor menţionate.

3.3 Extragerea caracteristicilor

Retelele Neuronale Convoluţionale (*Convolutional Neural Networks*) sunt folosite la scară largă în ceea ce priveşte problemele de recunoaştere a imaginii (*image recognition*) deoarece prezintă un important set de avantaje în comparatie cu alte tehnici.

3.3.1 Reţeaua Neuronală Convoluţională — Aspecte generale

O Reţea Neuronală Convoluţională (*Convolutional Neural Network*) este un sistem de *neuroni* artificiali care fac schimb de informaţii între ei. Conexiunile dintre aceştia au ponderi numerice care sunt transformate în timpul procesului de antrenare, astfel încât o reţea antrenată corespunzător să raspundă corect (sa returneze o valoarea corectă) cand la intrare îi este aplicată o imagine pe care trebuie să o recunoască.

Straturile (*layers*) unei astfel de reţele sunt construite astfel încât primul strat detectează nişte tipare (*patterns*) primitive din imaginea de input, al doilea detectează tipare ale tiparelor, al treilea tipare ale tiparelor precedente şi aşa mai departe. Un CNN are, în cele mai multe cazuri, intre 5 şi 25 de astfel de straturi pentru recunoaşterea tiparelor.

Reţelele neuronale sunt inspirate din sistemele neuronale biologice. Unitatea de bază (structurală şi functională) a sistemului nervos este neuronul. Neuronii sunt conectaţi între ei cu ajutorul sinapselor (legături neuronale). Figura 3.3 propune o comparaţie între un neruon biologic şi un model matematic al unui neuron.

Deşi în lucrările lor [5], [6], autorii propun metode diferite de a aborda clasficarea imaginilor din diferite puncte de vedere, un lucru pe care îl au în comun şi asupra căruia aproape că nu există contestaţie în acest domeniu este folosirea reţelelor neuronale de tip convoluţional oricând vine vorba de a lucra (indiferent de scopul final: clasificare, clusterizare etc.) cu un set de date compus din imagini.

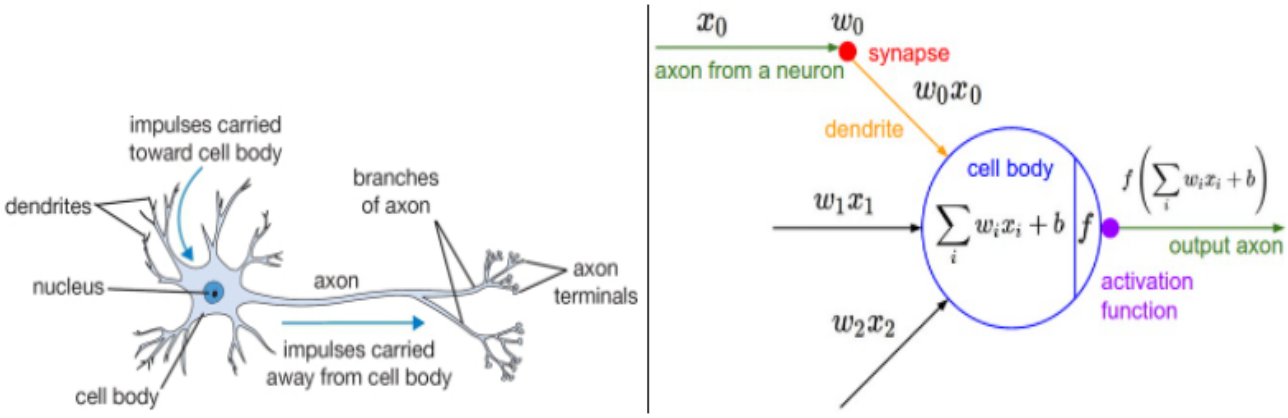

Figura 3.3: Neuron biologic (stanga) şi reprezentarea sa matematică (dreapta) [13]

În domeniul clasificării de imagini (scopul acestei lucrări), identificarea se face folosind setul de date de antrenare: imagini ale radiografiilor pulmonare, fiecare impreună cu categoria aferentă acesteia (pentru a putea determina acurateţea rezultatelor). Procesul de antrenare seamănă cu o buclă de reglare clasia care îşi ajustează ponderile într-un mod iterativ (calculând eroarea la fiecare pas) până se ajunge la o valoare acceptabilă a erorii. Schema bloc a procesului descris anterior este reprezentată în Figura 3.4, unde se pot vedea elementele şi cum acestea interacţionează între ele, precum şi influenţa lor asupra reţelei neuronale: ajustarea ponderilor în funcţie de eroarea calculată.

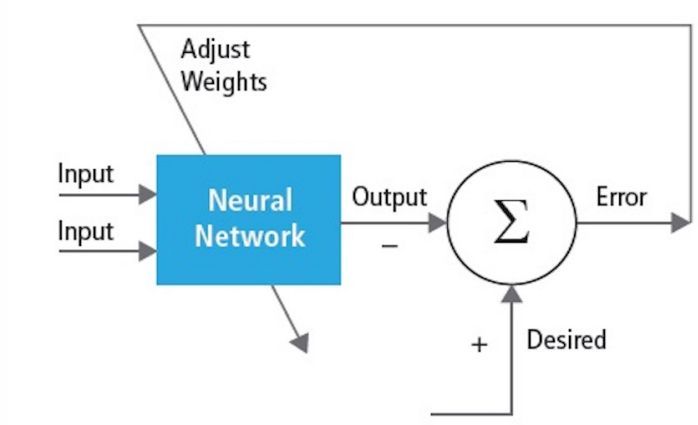

Figura 3.4: Schema procesului de învaţare a unei Reţele Neuronale

Reţelele Neuronale Convoluţionale se prezintă ca fiind cea mai bună şi corectă metodă de identificare şi clasificare a imaginilor. Nu doar că oferă cea mai bună soluţie din punct de vedere al performanţelor (comparativ cu alţi algoritmi de identificare), ci chiar

depăşesc în performanţe factorul uman în probleme cum ar fi recunoaşterea unor specii de animale (de exemplu specii de câini sau păsări) [14].

Layerele unei reţele neuronale pot fi împaţite în 3 categorii: layerul de intrare (*input layer*), layerele ascunse (*hidden layers* sunt în număr mai mare şi pot fi la rândul lor clasificate în mai multe grupuri) şi layerul final de ieşire (*output layer*). Figura 3.5 pune în evidenţă şi modul cum acestea comunică între ele la nivel înalt.

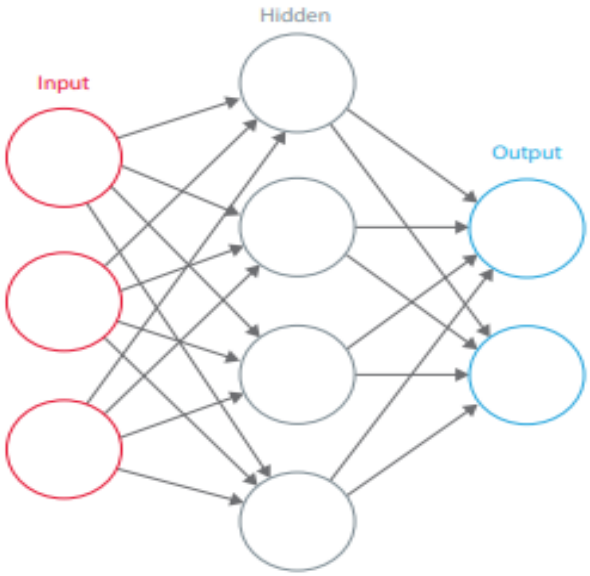

Figura 3.5: Distribuţia layerelor într-o reţea neuronală. Sursa: [15]

3.3.2 Modul de extragere al caractersiticilor

Caractersiticile (*features*) au fost extrase cu ajutorul reţelelor neuronale. Propuse de Baltruschat, I.M., Nickisch, H., Grass, M. *et al.* în [16], acestea sunt considerate cel mai înalt standard în domeniul de clasificare a imaginilor. Layerele convoluţionale din reţea extrag trăsăturile dintr-o anumită imagine, învăţand pe perioada antrenării care caracteristică este specifică fiecărei clase.

Un alt mod in care poate fi facuta o clasificare este fara a antrena deloc o retea neuronala cu imaginile din setul de date, ci folosirea unor modele antrenare anterior si consacrate in acest domeniu. Acest tip de abordare se numeşte *transfer learning* şi este des întlânit în cazul clasificărilor de imagini medicale [13], [4]. Figura prezintă o comparaţie între abordarea clasică: un model pentru fiecare problemă şi transfer learning: un model deja antrenat care este adaptat la o problemă nouă.

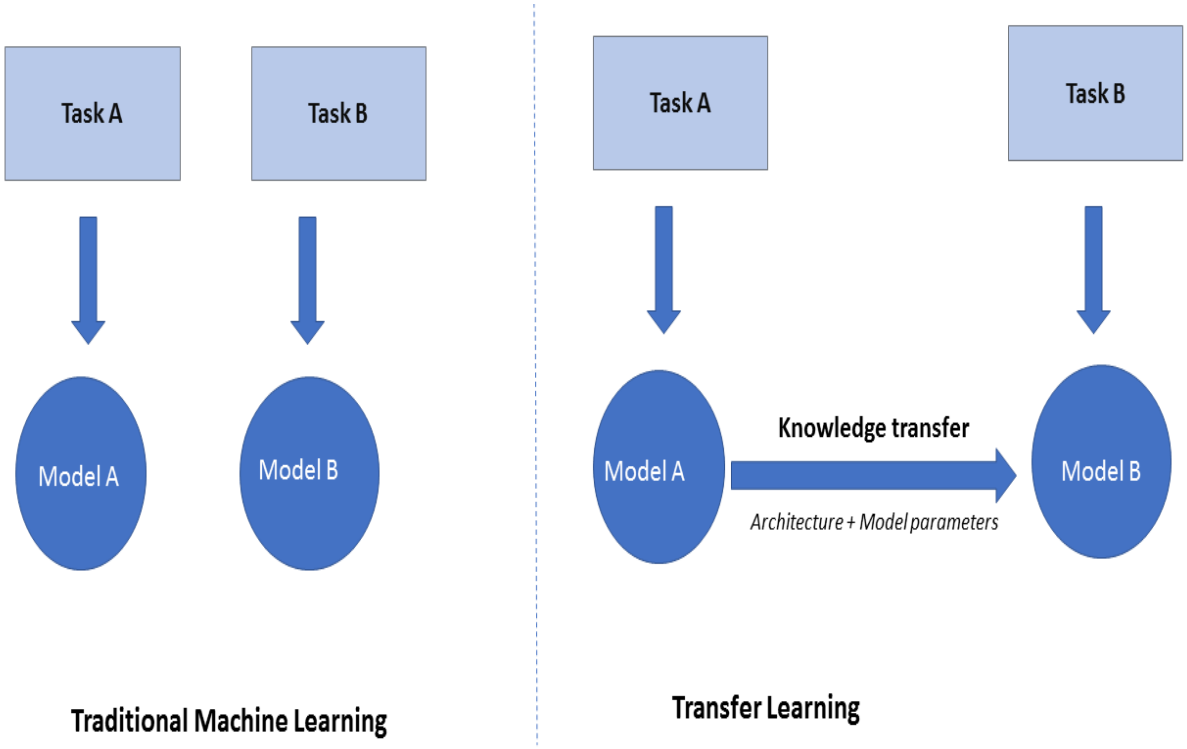

Figura 3.6: Metoda clasică de Machine Learning şi metoda Transfer Learning [17]

3.3.3 Clasificarea finală

Ultima parte a reţelei neuronale este alcătuită din partea de clasificare. În cadrul acestei părţi, părerile sunt împărţite. Unii autori susţin în lucrările lor [4],[5] că antrenarea unei reţele neuronale urmată de folosirea unei funcţii de activare (Sigmoid, ReLU etc.) care să primească ca input caractersiticile obţinute de reţea până în acel moment este cea mai bună practică pentru obţinerea unui rezultat. Pe de altă parte, există autori care susţin în publicaţiile lor [18] că acele caractersitici obţinute în urma antrenării reţelei neuronale pot fi folosite pentru a antrena un model mai simplu de machine learning (Regresie Logistică, Support Vector Machine, Arbori de decizie etc.) care să realizeze predicţia pe baza caracteristicilor oferite de reţeaua neuronala. În acest din urmă caz, reţeaua neuronală ar urma să se ocupe de segmentarea imaginii, după care acestea ar fi folosite drept punct de pornire pentru algoritmii sau modelele menţionate anterior.

Capitolul 4

Analiză şi fundamentare teoretică

În prima parte a acestui capitol este prezentată soluţia propusă din punct de vedere teoretic: arhitectura reţelelor neuronale convoluţionale, algoritmii pentru pre-procesarea datelor, extragerea caractersiticilor şi soluţia finală propusă.

Arhitectura modelului propus este este divizată în trei părti: partea de pre-procesare, de extragere a caractersiticilor şi ultimul pas de clasificare.

4.1 Soluţia propusă

4.1.1 Metode pentru pre-procesarea datelor

În lucrarea de faţa s-a folosit un set de date conţinand 5856 de imagini cu radiografii la nivelul toracelui. Aproxtimativ 1500 dintre acestea sunt clasificate ca fiind ca aparţinand clasei *normal*, restul de peste 4200 indicand prezenţa pneuomniei.

În figura 4.1 sunt prezente exemple de radiografii din setul de date aparţinand atat clasei normale cât şi clasei bolnave (avand pneumonie), observandu-se caracteristicile diferite ale acestora.

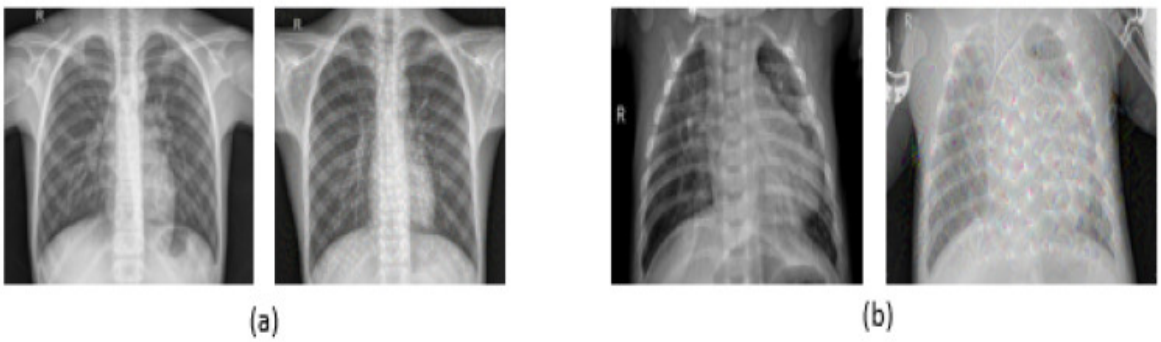

Figura 4.1: (**a**) Imagini cu plămani normali; (**b**) Imagini cu plămani bolnavi.

Imaginile din setul de date au rezoluţii de la **712** x **439** pana la **2338** x **2025** (pixeli). Înainte de a le furniza ca şi intrare pentru model, acestea trebuie să fie aduse la o dimensiune standard: **224** x **224** pixeli. Înainte de începerea antrenării reţelei neuronale, datele au fost împărţite aleatoriu în două părti: 70% destinate antrenării şi restul de 30% validarea rezultatelor.

Avand în vedere diferenţa de clase enunţată în paragrafele anterioare (diferenţa de aproape 3000 de imagini în favoarea clasei plămanilor sănatoşi), abilitatea de generalizare a unui algoritm poate fi grav afectată (acesta ar învăţa pe de rost). Pentru a îmbunătăţii performanţele s-au folosit metode de augmentare, înaintea începerii antrenării reţelei neuronale, care nu au alterat clasele cărora imaginile aparţineau.

Fiecare radiografie (n.r. imagine) a fost transformată conform paşilor descrişi ulterior:

- Rotirea: Fiecare imagine a fost rotită cu 20°;
- Transpunerea: Matrice de pixeli care formează o imagine a fost transpusă;
- Iluminarea aleatorie: S-au folosit diferite valori pentru a mări luminozitatea imaginii.

Partea de pre-procesare cuprinde de asemenea şi normalizarea imaginii. Pentru fiecare canal de culoare se împarte valoarea fiecărui pixel la 255. Aplicand un astfel de filtru *gray-scale*, valorile pixelilor nu vor mai fi în intervalul lor standard [0-255] ci vor avea doar valori in intervalul [0-1] (4.1). În urma acestui proces de normalizare se obţine o imagine gri (precum cele din Figura 4.1).

$$imagine\_normalizata = \frac{imagine_{ij}}{255} \tag{4.1}$$

Asupra imaginilor se vor mai aplica şi alte transformări, în momentul începerii procesului de antrenare al reţelei neuronale. Despre acesta vom discuta într-una dintre secţiunile următoare.

4.1.2 Extragera trăsăturilor

Extragerea trăsăturilor sau a caractersiticilor se realizează folosind o Reţea Neuronală Convoluţională (*Convolutional Neural Network*). Acest proces reprezintă practic procesul de antrenare al reţelei noastre neuronale, urmând ca trăsăturile obţinute să fie date ca şi intrare unei funcţii de activare (acest ultim aspect este prezentat in secţiunea următoare). Arhitectura extragerii trăsăturilor folosind o reţea neuronală este prezentată în Figura 4.2

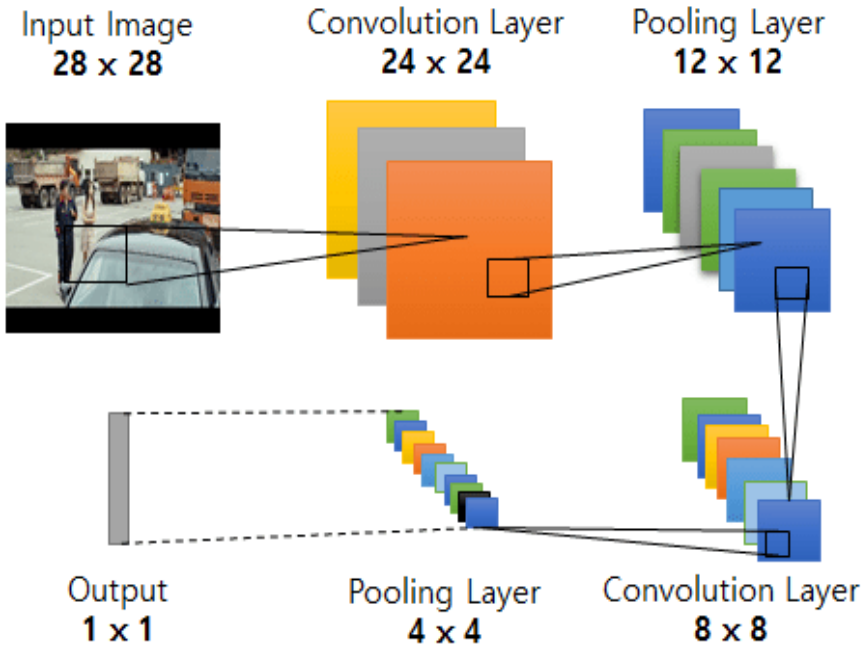

Figura 4.2: Extragerea caractersiticilor folosind CNN [19]

Convoluţia

Blocul cu cea mai mare importanţa, cel care dă şi numele acestui tip de reţea neuronală, este cel convoluţional. Convoluţia este o operaţie matematică de a combina două seturi de informaţii (semnale, matrici etc.). În cadrul acestei soluţii, convoluţia este aplicată imaginilor sub forma unui filtru de convoluţie pentru a produce o hartă a caractersiticilor (Figura 4.3).

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Figura 4.3: Exemplu de input şi filtrul de convoluţie aplicat asupra lui

În partea stangă a figurii 4.3 se află un input al blocului de convoluţie (de exemplu: imaginea unei radiografii), iar alături de acesta este filtrul asociat. Deoarece mărimea filtrului este de 3x3, aşa se numeşte şi acest tip de convoluţie.

Operaţia de convolutie se realizează prin suprapunerea filtrului peste datele de intrare. Pentru fiecare suprapunere, înmulţim matricile element cu element şi însumăm rezultatul. Suma rezultată va fi adaugată în harta caractersiticilor (*feature map*). Zona verde (Figura 4.4) unde are loc convoluţia se numeşte *receptive field*. Procesul este unul iterativ, mergand pas cu pas până la completarea hărţii caracteristicilor.

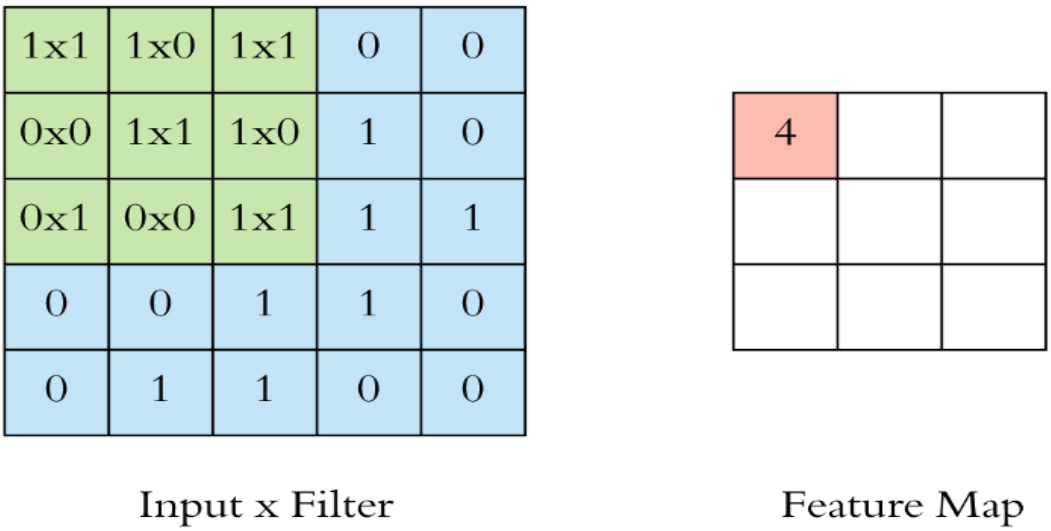

Figura 4.4: Aplicarea operaţiei de convoluţie [20]

Exemplul evidenţiat mai sus este aplicarea convoluţiei pentru o imagine bidimensională (2D) folosind un filtru 3x3. În realitate, aceste convoluţii sunt realizate în spaţiu tridimensional (3D). O imagine este reprezentată ca o matrice 3D, având dimensiuni pentru inălţime, lăţime şi adancime, prin ultima întelegând canalele de culori (RGB). Un filtru de convoluţie are dimensiuni standrad (5x5, 3x3) şi trebuie să fie şi acesta tridimensional pentru a acoperi şi adancimea imaginilor.

Un alt aspect important legat de acest proces este că, asupra unei imagini, se fac mai multe convoluţii, fiecare folosind câte un filtru diferit si rezultând într-o hartă a caracteristicilor diferită. La final se aşează toate hărţile obţinute şi acesta va fi rezultatul final al stratului de convoluţie. Acest proces va fi exmplificat în paragrafele următoare.

Se foloseşte ca exemplu o imagine cu dimensiunea 32x32x3 şi un filtru de mărimea 5x5x3 (adancimea filtrului de convoluţie trebuie să corespundă cu adancimea imaginii: 3).

Cand filtrul se află la o anumită locaţie acoperă o mică secţiune din imagine şi se realizează operaţia de convoluţie descrisă mai sus. Diferenţa este că de această dată se

face suma matricilor multiplicate in 3D in loc de 2D, dar rezultatul este tot un scalar. Se continuă procesul iterativ de convoluţie descris mai sus şi se agregă rezultatele in harta caractersiticilor. Aceasta din urmă va avea dimensiunea 32x32x1, după cum se observă in Figura 4.5.

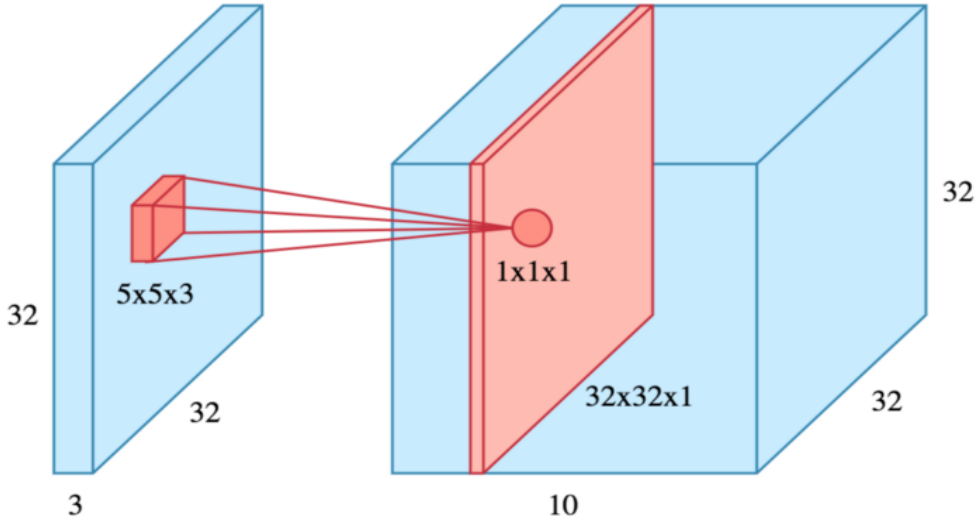

Figura 4.5: Operaţia de convoluţie în spaţiul tridimensional [20]

Dupa folosirea a 10 astfel de filtre de convoluţie cu dimensiunea 32x32x1 şi stivuirea lor pe adancime s-a obţinut ieşirea finală a stratului de convoluţie şi anume: o hartă de convoluţie cu dimesiunile 32x32x10, adancimea reprezentând numărul de convoluţii care au fost efectuate asupra imaginii iniţiale.

Convoluţia 3D nu este altceva decât o extensie a celei din spaţiul bidimensional. Mai corect, ea reprezintă o generalizare a cazului particular de convoluţie 2D. Dacă în cazul dimensiuni lăţime X înălţime, convoluţia descrie relaţia in acel spaţiu bidimensional, la fel şi o convoluţie 3D, în spaţiul lăţime X înălţime X adâncime descrie relaţia între obiectele din spaţiul tridimensional. Bineînteles, în cazul imaginilor, folosim se foloseţe exclusiv convoluția tridimensională datorită naturii inputului.

În figura 4.6 se observă un exemplu în care două convoluţii se realizează indepedent, simultan pe părţi diferite ale imaginii de intrare şi rezultă doua hărţi distincte ale trăsăturilor. Acest proces se repetă până la acoperirea completă a imaginii şi obţinerea completă a tuturor acestor hărti sau măşti ale caractersiticilor.

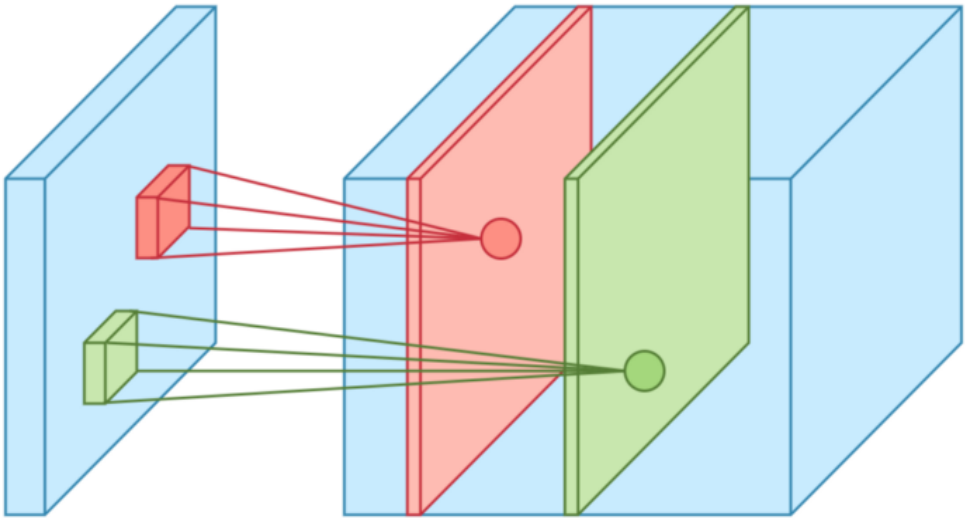

Figura 4.6: Exemplu de doua convoluţii aplicate simultan şi rezultatele acestora [20]

Pooling

După terminarea paşilor de convoluţie, urmatoarea etapă este realizarea *pooling*-ului pentru reducerea complexităţii. Această operaţie are ca scop reducerea parametrilor rezultaţi, lucru ce scurtează timpul de antrenare şi combate incapacitatea reţelei neuronale de a generaliza (reduce invăţarea pe de rost). Straturile de *pooling* se ocupă de fiecare *feature-map* (harta caracteristicilor) individual, reducând inălţimea şi lăţimea, dar păstrând adancimea intactă.

Cea mai comună formă de pooling este *max-pooling* care alege cea mai mare valoare dintr-o fereastră pre-definită. Contrar convoluţiei, aceasta nu primeşte niciun parametru. Operaţia se efectuează acoperind pe rând câte o porţiune din feature-map şi alegând valoarea maxima. Ca şi la convoluţie, trebuie să specificam dimensiunea ferestrei care acoperă inputul. Un alt tip de pooling layer este *average pooling*, care, după cum ii spune şi numele, în loc să ia valoarea maximă ca şi în cazul layerului max-pooling, calculează valoare medie a fiecărei ferestre din harta caractersiticilor.

Reţeaua neuronală va avea mai multe layere de pooling (acestă operaţie fiind aplicată de mai multe ori în timpul antrenării), după fiecare layer de convoluţie urmând să existe şi un astfel de pooling layer. Alegerea max-pooling sau average-pooling este specifică fiecărui caz şi se poate experimenta sau investiga pentru a se realiza alegerea potrivită [21].

În continuare (Figura 4.7) este prezentat un exemplu de max-pooling în cadrul căruia este folosită o fereastră având dimensiunea de 2x2 şi pasul de realizare al poolingului fiind de 2. Fiecare dintre culorile prezente reprezintă o acoperire diferită folosind fereastra cu dimensiunea specificată. Se observă că având dimensiunea de 2x2 şi pasul de 2, nu există suprapuneri la mutarea ferestrei peste matricea de input. Dimesniunea matricii de intrare este 4x4, rezultatul fiind bineînteles o injumatăţire a acestei matrici bidimensionale.

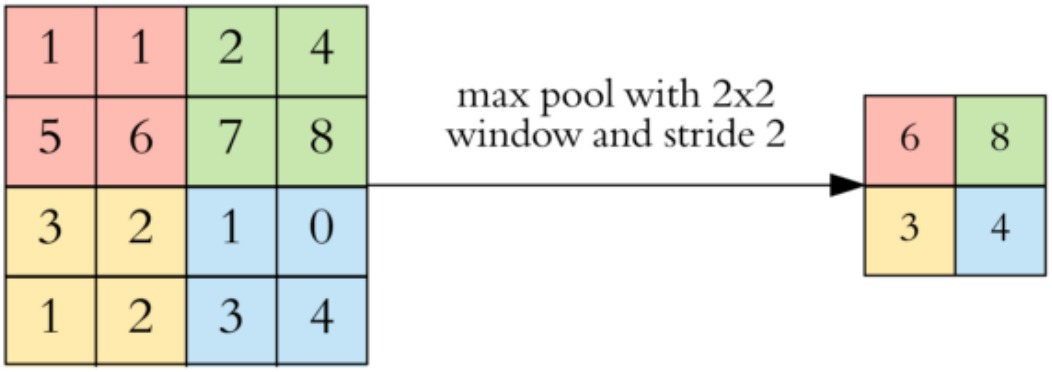

Figura 4.7: Exemplu de max-pooling cu fereastra de 2x2 şi pasul de 2 [21]

Se observă cum procesul de pooling injumătăţeşte infromaţiile primite prin intermediul matricii de intrare. Aceasta reprezintă şi motivul principal pentru folosirea lui: reducerea spaţiului de stocare al informaţiei pentru optimizarea şi creşterea performanţei, fară a se pierde insă informaţii importante pe parcurs.

Urmează o analiză a hărţii caractersiticilor înainte şi după pooling, de data aceasta in spatiul tridimensional Se va lua ca şi exemplu o harta cu dimensiunile 32x32x10 (cea care s-a analizat în paragrafele anterioare). Folosind aceeaşi parametrii pentru pooling menţionaţi anterior (fereastra cu dimensiunea de 2x2 şi pasul de pooling 2), rezultatul va avea dimensiunea finală 16x16x10, deci o înjumătăţire. Ambele, atat înălţimea cât şi lăţimea au fost înjumătăţite, însă adancimea ramane intactă deoarece pooling-ul se aplică independent fiecărui strat din harta de input (în cazul acesta: 10), nealterând astfel şi adâncimea. O schiţare grafică a procesului explicat în acest paragraf se regăseşte în Figura 4.8.

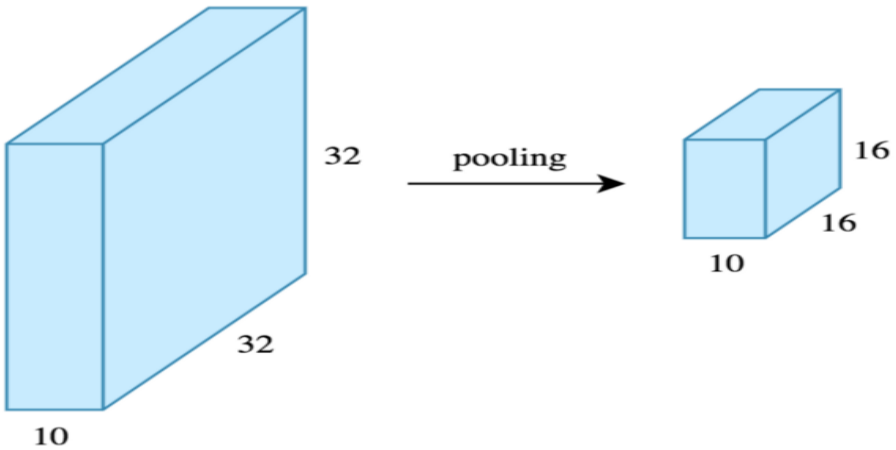

Figura 4.8: Exemplificarea utilităţii poolingului in spatiul 3D [21]

Prin înjumătăţirea înalţimii şi a lăţimii, am redus ponderile la un sfert. Având în vedere ca în cazul arhitecturii Reţelelor Neuronale Convoluţionale avem de-a face cu milioane de ponderi (*weights*), această minimzare este foarte importantă.

Hiper-parametrii

În următoarele paragrafe se vor analiza, în primul rând, hiper-parametrii din punctul de vedere al convoluţiei, analizând opţiunile disponibile. În cazul acesta sunt dipsonibili 3 parametri asupra cărora trebuie făcute decizii:

- Dimensiunea filtrului: de obicei se foloseşte un filtru 3x3, dar 5x5 sau 7x7 sunt unele variante în funcţie de aplicaţia unde sunt folosite. Filtrele au şi o a treia dimensiune, adancimea, dar aceasta fiind egală cu cea a imaginii de input, este omisă.

- Numărul filtrelor: Este parametrul cel mai variabil, luând valori puteri ale lui 2 între 32 şi 1024. Folosind mai multe filtre rezultă un model mai puternic, dar se riscă căderea în *overfitting* (învăţare pe de rost) din cauza numărului crescut de parametrii. De obicei se începe cu un numar mic de filtre la straturile iniţiale, urmând ca acest număr să crească pe măsură ce se ajunge mai adanc în reţeaua neuronală.

- Pasul: Se pastrează de obicei la valoarea implicită: 1 (*pas cu pas*).

Blocuri complet conectate

După straturile de convoluţie şi de pooling urmeaza o serie de straturi complet conectate (*fully connected layers*) pentru a îngloba arhitectura reţelei neuronale. Această arhitectură a fost prezentată în secţiunile anterioare, figura 3.5.

Un bloc complet conectat înseamna că fiecare neuron din stratul anterior este conectat la fiecare neuron din stratul următor. În paragrafele precedente s-a menţionat că ieşirile, atat a blocurilor de convoluţie cât şi a celor de pooling, sunt nişte volume 3D, insă un layer complet conectat aşteaptă un vector de numere unidimensional (1D). Din acest motiv, ieşirile primelor blocuri vor fi aplatizate (*flatten*) pentru a putea servi ca şi input pentru noul bloc. Prin procesul de aplatizare înţelegem rearanjare din 3D in 1D, fară alte complicaţii matematice.

Cea mai puternică componenta (şi principala sursă de putere) a unei CNN este blocul convoluţional. Acestea pot detecta automat caracteristici, fiind furnizate doar cu o imagine şi clasa aparţinătoare a acesteia. Motivul pentru care au această abilitate este modul cum sunt structurate: aşezate unul peste altul. Primul bloc detectează muchiile, următorul combina muchiile pentru a crea forme, urmând ca cele ce urmează să unească toate aceste informaţii pentru a detecta un obiect. Reţeaua neruonală nu ştie ce sunt acelea obiecte, dar întalnindu-le în repetate rânduri învaţă că acestea sunt caractersitici

importante. În final, straturile complet conectate folosesc aceste caractersitici pentru a putea clasifica corect o imagine.

4.1.3 Clasificarea

O Reţea Neuronală Convoluţională este o combinaţie a două mari componente: extragerea trăsăturilor şi partea de clasificare. Convoluţia şi procesul de pooling asigură partea de extragere a caractersiticilor. De exemplu, pentru un caz mai general (un animal), blocurile de convoluţie detectează caractersitici cum ar fi: doi ochi, urechi, patru picioare etc. Blocurile complet conectate acţionează ca şi un clasificator asupra acestor caractersitici, furnizând o probabilitate ca imaginea să aparţină unei anumite clase.

Inainte de a putea face o clasificare, procesele descrise anterior vor ajuta la antrenarea reţelei neuronale: se calculează eroarea dupa fiecare parcurgere, dupa care ponderile tuturor blocurilor sunt ajustate astfel încât eroarea să fie diminuată. Acest proces este unul repetitiv care se continuă până când eroarea nu se mai reduce semnificativ. În secţiunea următoare se vor regasi mai multe detalii legate de acest proces de antrenare al reţelei neuronale.

În cazul unei clasificări binare cum este cea de faţă (plămâni sănătoşi sau plămâni bolnavi). Daca un neuron al clasei 0 (plămâni sănătoşi) primeşte o valoare de *0*, aceasta inseamnă ca este 100% nesigur dacă acea caractersitică aparţine clasei sale. Dacă primeşte valoarea *1*, este 100% sigur că acea caractersitică aparţine clasei plămâni sănătoşi (clasa 0). În ultimul bloc complet conectat, neuronii primesc valori între 0 şi 1. Aceste valori reprezintă grade de certitudine. De exemplu, o valoare de 0.7 reprezintă o certitudine de 70%.

Neuronii care au o certitudine ridicată când o caracteristica este identificată ştiu că acea imagine aparţine unei anumite clase. Aceştia produc un output, un echivalent matematic, care indică că ar trebui activaţi, deoarece acele trăsături le aparţin. Daca acest lucru se întamplă de mai multe ori în procesul de antrenare, reţeaua neruonala învaţă că atunci când anumite caractersitici sunt prezente, imaginea aparţine clasei respective.

Odată ce reţeaua a fost antrenată, aceasta poate primi ca intrare o imagine şi va fi capabilă să returneze probabilitatea ca acea imagine sa aparţină unei anumite clase, această probabilitate fiind una cu un nivel ridicat de ceritutdine.

Blocul complet conectat (fully connected) conţine pe ultimul strat un clasificator. Acest clasificator este de fapt o funcţie de activare. Scopul acestui bloc este de a folosi caractersticile rezultate din blocurile diferite pentru a clasifica imaginea, băzându-se pe datele de antrenare.

Ca si funcţie de activare pentru clasificare, vom folosi funcţia Sigmoid. Aceasta produce valori în intervalul [0, 1]. Această ultimă proprietate a funcţiei ajută la vizualizarea gradului de certitudine pentru o anumită clasă (între 0 şi 100%). Valorea de prag pentru această funcţie de activare va fi 0.5, prin urmare orice output mai mare decât această valoare va fi atribuit clasei 1, iar orice valoare mai mică decât 0.5 va fi atribuită clasei 0.

4.2 Suport teoretic si algoritmi utilizaţi

În aceasta secţiune se vor discuta în detaliu conceptele prezentate pe scurt în paragrafele anterioare, precum şi introducerea altor concepte care intra în componenţa lor.

Aceasta vine ca şi o completare şi o aprofundare în algoritmii din spatele fundamentelor teoretice prezentate anterior.

4.2.1 Reţele Neuronale Artificiale

Reţelele Neuronale Artificiale (**ANN** - Artificial Neural Networks), sunt o serie de algoritmi bazaţi pe Machine Learning. ANN folosec, ca şi unitate de bază, modelul matematic al unui neuron unde impulsul nervos primit la intrare este înmulţit cu o pondere ω_i. Pe langă pondere, modelul neuronului mai învaţă şi alt parametru: factor de infulenţă sau *bias*. Semnalul de ieşire (output) este reprezentat de produsul scalar dintre ponderi şi semnalul de intarare la care se mai adaugă şi factorul de influenţă 4.2.

$$\sum_i \omega_i \times x_i + bias \tag{4.2}$$

Semnalele rezultate din produsul scalar realizat de către neuron vor fi folosite ca şi intrare pentru funcţiile de activare din cadrul reţelei neuronale:

- Funcţia Sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.3}$$

- ReLU - Rectified Linear Unit:

$$f(x) = max(0, x) \tag{4.4}$$

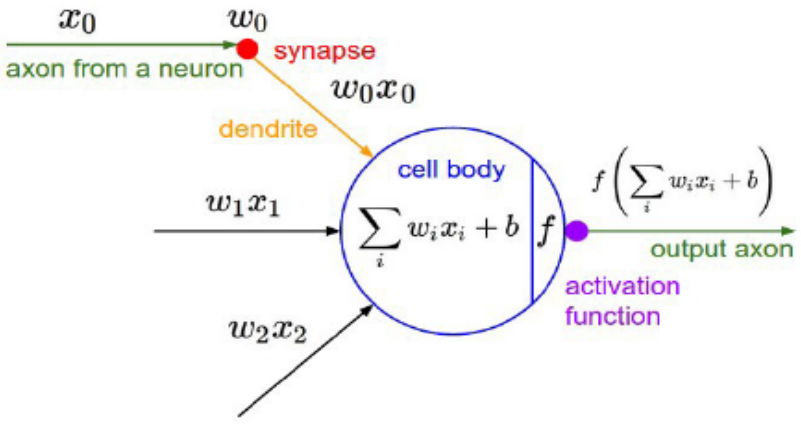

Figura 4.9: Modelul matematic al unui neuron [13]

Motivul pentru care acest tip de reţele neuronale nu se pot folosi în cazul clasificarii de imagini este datorat modului cum sunt stocate imaginile în memorie. De exemplu, având imagini de dimensiunea 500x500x3 ar rezulta neuroni cu câte 750000 de ponderi. Acestea se pot folosi doar în cazul in care imaginile au dimensiuni mici. Acesta este raţionamentul pentru care s-au introdus reţelele neuronale convoluţionale, care eleveaza clasicele ANN.

4.2.2 Reţele Neuronale Convoluţionale

Reţelele Neuronale Convoluţionale (*CNN*) sunt o sub-clasă a reţelelor neuronale care sunt performante în cazul clasificării de imagini, deoarece ţin cont de faptul ca intrările sunt imagini, deci neuronii in cadrul acesteia sunt aşezaţi după cele 3 dimensiuni ale unei imagini: inălţime, lăţime si adâncime.

CNN funcţionează ca şi reţelele neuronale clasice: fiecare neuron primeşte un input (datele de intrare), efectuează produsul scalar între ceea ce primeşte la intrare şi ponderea asociată. Se observă în Figura 4.10 cum interacţioneaza neuronii între ei: fiecare neuron dintr-un anumit layer al reţelei primeşte ca sî intrare ieşirea fiecărui neuron din stratul neuronal precedent.

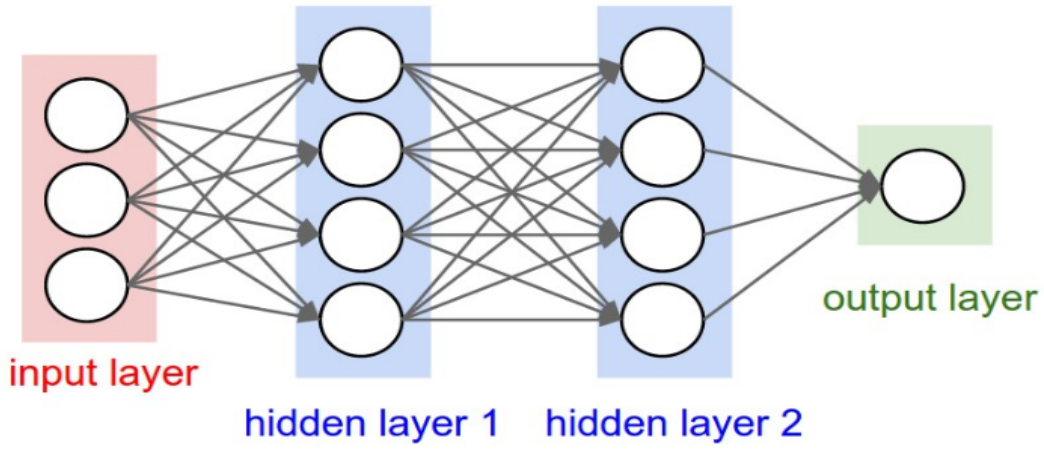

Figura 4.10: Interconectarea neuronilor într-un CNN [20]

Avantajul major în cazul procesarilor de imagini adus de către Reţelele Neuronale Convoluţionale este faptul că acestea necesită pre-procesări puţine comparat cu alte modele care se ocupă cu clasificarea imaginilor. Aceasta inseamnă că reţeaua învaţă să optimizeze filtrele pe care le va folosi prin învaţarea automată, pe când în cadrul algrotimilor clasici aceastea trebuie să fie create de către cel care dezvoltă modelul. Această independenţă

între cunoştiinţele anterioare şi intervenţia factorului uman în extragerea caractersiticilor reprezintă un avantaj major pentru reţelele neuronale convoluţionale.

Layerele princpiale din arhitectura unei reţele neuronale convoluţionale

Primul layer - este layerul (stratul) de intrare în reţeaua neuronală. Acesta este layerul care primeşte datele după ce asupra lor s-au aplicat câteva operaţii de pre-procesare. Conţine (in cazul clasificarii de imagini) culorile sau intensitatea culorilor pixelilor care compun imaginea iniţiala

Layerul convoluţional - cea mai importantă parte a reţelei neuronale convoluţionale. Scopul acestui filtru este de a învăţa o mulţime de filtre, aceştia find parametrii pe care trebuie sa îi ajusteze pe durata procesului de învăţare. Prin filtru se inţelege o fereastră sau un tablou de dimensiuni mai mici decât intrarea, excepţie făcând adâncimea care trebuie să fie identică cu cea a intrării.

Descrise şi în secţiunea anterioară, aceste filtre sau tablouri de convoluţie sunt folosite pentru a putea aplica operaţia de convoluţie asupra unor imagini (matrici tridimensionale, Figura 4.11), rezultând, pentru fiecare tip de filtru, câte o hartă caractersiticilor sau hartă de activare bidimensională care conţine răspunsul filtrului dupa convoluţie pentru fiecare poziţie a imaginii în funcţie de dimensiunea acestuia (3x3, 5x5 etc.). Reţeaua neruonala va invată filtre, acestea se vor activa în momentul apariţiei unor trăsături specifice unei anumite clase, pe care reţeaua le invaţă (în cazul pneumoniei: zone întunecoase, neregulate etc.).

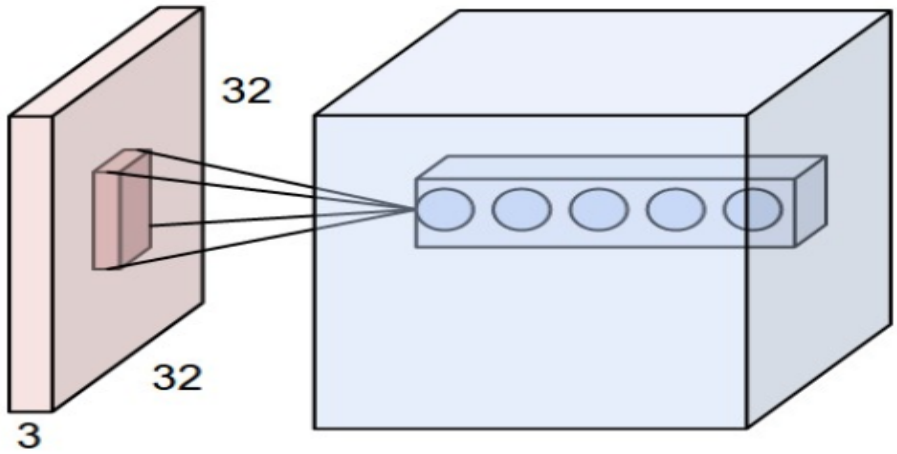

Figura 4.11: Layerul (stratul) de convoluţie [20]

Layerul Pooling - acest layer îşi aduce contribuţia în reducerea progresivă a dimensiunii spaţiale cu scopul de a reduce numărul de parametrii şi al numărului computaţiilor unei reţele neuronale pentru diminuarea învăţarii pe de de rost (*overfitting*). Sectiunea anterioară oferă mai multe figuri şi exemple despre acestă funcţionalitate a acestui layer.

Un Pooling layer sau Max-Pooling layer are următorii hipermarapetrii care pot fi ajustaţi:

- Dimensiunea ferestrei (nucleului) : N * M
- Pasul - numărul de pixeli cu care se va muta fereastra în procesul de pooling (4.7).

Layerul Dropout - această componentă a reţelei neuronale acţionează prin reducerea unui procent din neuronii implicaţi în procesul de învăţare cu scopul de a reduce şi mai mult overfittingul.

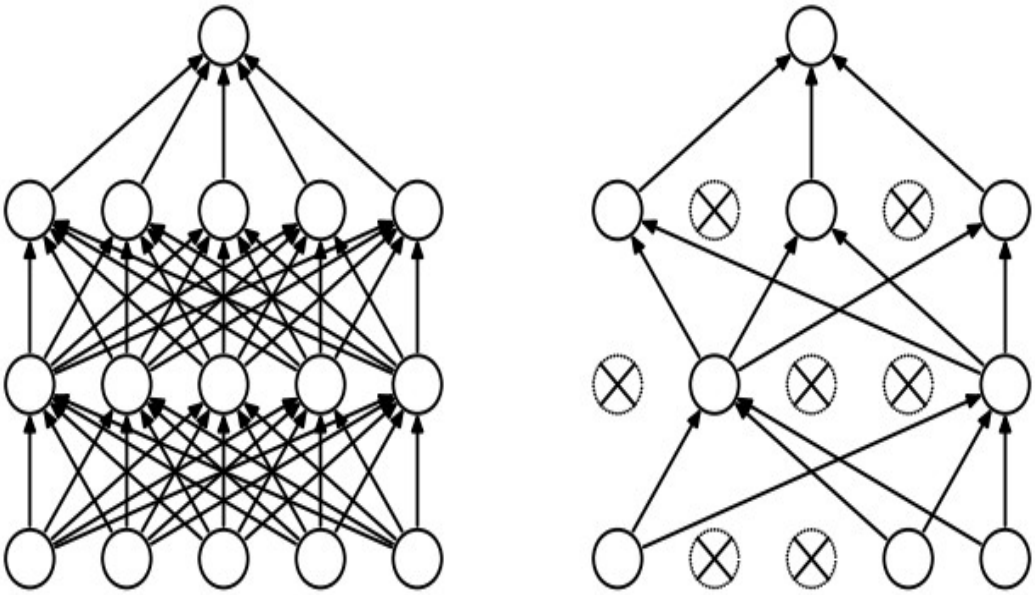

Figura 4.12: O reţea neruonală înainte şi după droput [20]

Layerul UpSampling - sau layer de deconvoluţie. Operaţia pe care acest layer o aduce în cadrul reţelei neuronale convoluţionale este repetarea valorii primite la input în interiorul unei ferestre.

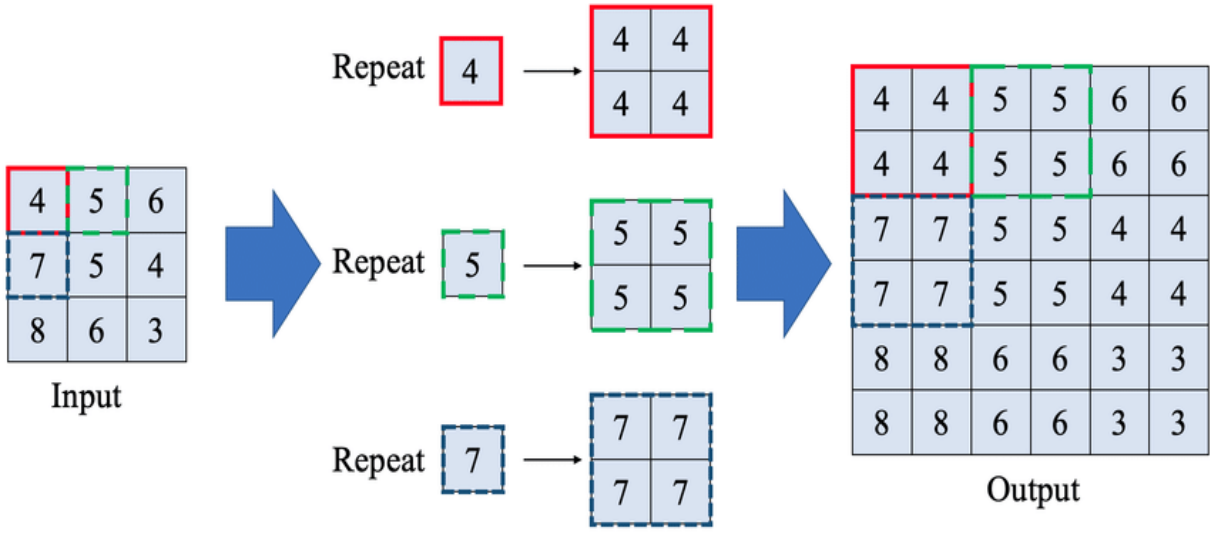

Figura 4.13: Exemplificarea procesului de upsampling [22]

Acest tip de layer se foloseşte în segmentarea imaginilor. În cazul unei operaţii de segmentare, o reţea neuronală poate fi împarţită în: reţeaua de convoluţie şi reţeaua de deconvoluţie. Prima va incerca să clasifice unele părţi ale imaginii care vor fi trimise reţelei de deconvoluţie, urmând ca cea din urmă să realizeze asamblarea imaginii segmentate.

Antrenarea unei reţele neuronale nu ar fi posibilă fară o funcţie de pierdere (*loss function*). Funcţia de pierdere nu face altceva decât calcuarea diferenţei dintre ieşirea reală a reţelei neuronale şi raspunsul corect care ar trebui rezultat (în cazul clasificării). Bazat pe acest rezultat, reţeaua işi ajutează ponderile pentru a reduce cât mai mult această funcţie.

Printre cele mai populare şi des utilizare loss-functions, folosite în antrenarea unei reţele pentru clasificare, sunt:

- Entropia Încrucişată (Binary Cross-Entropy):

$$H_p(q) = -\frac{1}{\mathrm{N}} \sum_{i=1}^{N} \mathrm{y_i} \cdot \log(\mathrm{p}(\mathrm{y_i})) + (1 - \mathrm{y_i}) \cdot \log(1 - \mathrm{p}(\mathrm{y_i})) \tag{4.5}$$

- Coeficientul Sørensen-Dice:

$$Dice = \frac{2|\mathrm{A} \bigcap \mathrm{B}|}{|\mathrm{A}| + |\mathrm{B}|} \tag{4.6}$$

4.2.3 Convoluţia

Convoluţia, descrisă pe scurt şi în secţiunea anterioară, este procesul de combinare dintre pixelii unei imagini (elementele unei matrice) şi o fereastră (nucleu), rezultând astfel un pixel din imaginea destinaţie. Elementul sau pixelul obţinut în urma operaţiei de convoluţie este o combinaţie liniară a pixelilor din imaginea sursă peste care a fost suprapusă fereastra de convoluţie [4.14].

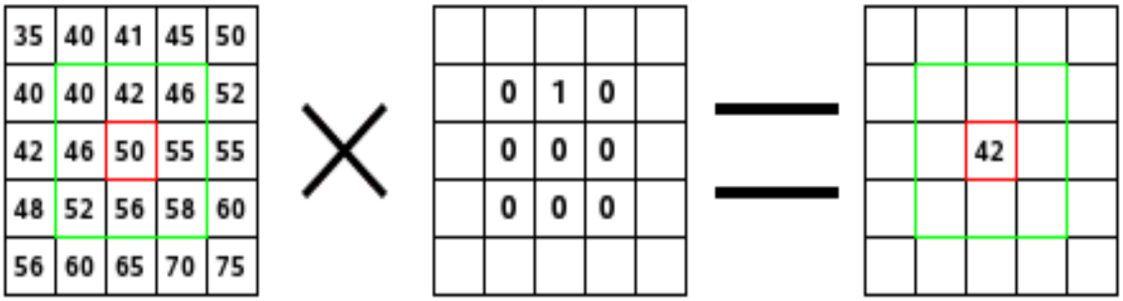

Figura 4.14: Exemplu de convoluţie cu dimensiunea ferestrei 3x3

Ecuaţia 4.7 descrie matematic operaţia de convoluţie dintre două funcţii f şi g, definită ca şi integrala produsului celor doua funcţii după ce una dintre ele este translatată.

$$\int_{-\infty}^{\infty} f(\tau) \cdot g(t-\tau)\,\mathrm{d}\tau \tag{4.7}$$

Convolutia, fiind si operatia care da numele retelei neuronale, este fundamentul pe care au fost introduse si dezvoltate aceste instrumente puternice si revolutionare in procesarea imaginilor din orice domeniu: retelele neuronale convolutionale.

Capitolul 5

Proiectare de Detaliu şi Implementare

Acest capitol este împărţit în doua secţiuni: în prima parte sunt prezentate tehnologiie utilizate în dezvoltarea aplicaţiei, iar în cea de-a doua parte este prezentată arhitectura soluţiei propuse.

5.1 Tehnologii utilizate

În implementarea şi dezvoltarea soluţiei propuse, limbajul de programare utilizat exclusiv este Python, acesta oferind stabilitate şi mijloacele necesare pentru atingerea scopului final. Este un limbaj cu o sintaxa simplă şi se preteaa foarte bine pentru utilizarea în chestiuni matematice, precum şi pentru procesarea de imagini care sunt foate uşor de manipulat în acest limbaj. În general, majoritatea aplicaţiilor care implementează algoritmi de machine learning folosesc Python ca şi limbaj de programare.

De asemenea, ca şi adaos la caractersiticile prezentate anterior, Python pune la dispoziţie şi o multime de librării ca şi ajutor pentru dezvoltarea unei aplicaţii de acest fel:

- Pandas
- Keras
- NumPy
- Scikit learn
- Matplotlib
- OpenCV

5.1.1 Pandas

Pandas este o librarie Python care este folosită în analiza şi manipularea datelor. În cadrul proiectului de faţa, aceasta a fost folosita pentru încarcarea setului de date folosit la antrenarea reţelei neuronale.

5.1.2 Keras

Keras este un API pentru dezvoltarea şi creearea reţelelor neuronale. La baza, Keras, foloseşte unele dintre cele mai noi soluţii şi cei mai noi algoritmi din acest domeniu: Tensorflow, CNTK sau Theano. Scopul creării acestuia a fost simplificarea procesului de dezvoltare al reţelelor neuronale.

Conform documentatiei oficiale, Keras este:

- **Simplu** – reduce timpul de scris cod repetitiv, oferind şansa dezvoltatorului să se concentreze pe părţile importante ale problemei.
- **Flexibil** – pune la dispoziţie metode rapide şi usor adaptabile.
- **Puternic** – este folosit în proiecte reale şi importante, oferind suport pentru scalabilitate şi performanţă.

Printr-un model Keras se inţelege modul de organizare al straturilor (layerelor) dintr-o reţea neruonală artificială. Un exemplu de model simplu este cel secvenţial, unde layerele sunt dispuse în formă de stivă. În Keras avem o multime de layere care pot fi modificate şi ajustate dupa nevoie, oferind de asemenea şi posibilitatea construirii unor layere noi. Printre layerele existente se numară: convoluţional, pooling, drop-out etc. Metoda care configurează procesul de antrenare sau învăţare se numeşte *compile*. Această funcţie primeşte ca şi parametrii: funcţia de pierdere (loss function), metrica de optimizare etc. Utilizând Keras, dezvoltarea şi implementarea unei reţele neruonale este relativ uşoară, odata ce se cunoaşte arhitectura potrivită a acesteia.

Keras oferă posibilitatea folosirii, în procesul de antrenare, atât a procesorului (CPU) cât şi a acceleratorului grafic (placă video, GPU). De asemenea, Keras conţine şi unele seturi de date care pot fi folosite pentru antrenare.

5.1.3 NumPy

Considerat cel mai important pachet ştiinţific din Python, NumPy facilitează operaţiile matematice complexe şi se afla la baza unor librarii si pachete mai complexe (de exemplu: scikit-learn). Ariile în care NumPy oferă cel mai mult ajutor:

- Suport pentru tablouri multidimensionale.
- Operaţii cu matrici sau şiruri mari de numere.

- Sortări sau filtrări după anumite reguli.
- Operaţii puternice de algebră liniară.
- Generarea de numere sau de şiruri de numere aleatoare.

5.1.4 Scikit-learn

Scikit-learn este o librărie Python pentru machine learning. Oferă o varietate de algoritmi pentru clasificare, regresie sau clusterizare, cum ar fi:

- Support Vector Machine
- Random Forest
- Gradient Boosting
- Multe altele

Menţionat şi în secţiunea anterioară, scikit-learn este strâns legat şi are la bază librăria Numpy. Printre modulele principale pe care scikit-learn le pune la dispoziţie (şi care au fost utilizate şi în proiectul de faţa) sunt:

- Învăţare supervizată: Arbori de decizie, Naive Bayes, Stochastic Gradient Desecent etc.
- Învăţare nesupervizată: Clustering, Estimarea densitatii etc.
- Evaluarea şi selectarea modelelor: Cross-Validation, ajustarea hiper-parametrilor
- Încarcarea şi pregătirea setului de date
- Altele

5.1.5 Matplotlib

Dupa cum ii spune şi numele, Matplotlib, este o librărie Python pentru desenare şi reprezentare grafică, folosindu-se şi aceasta de NumPy. Scopul acesti librarii este vizualizarea grafica a datelor, distribuţia acestora sau vizualizarea rezultatelor anumitor algoritmi. Principalele tipuri de ploturi utilizare sunt:

- Line Plot.
- Histograme.
- Plot 3D.

- Diferite charturi reprezentative care pot accentua gravitatea unor neconformităţi (Figura 5.1)
- Multe altele

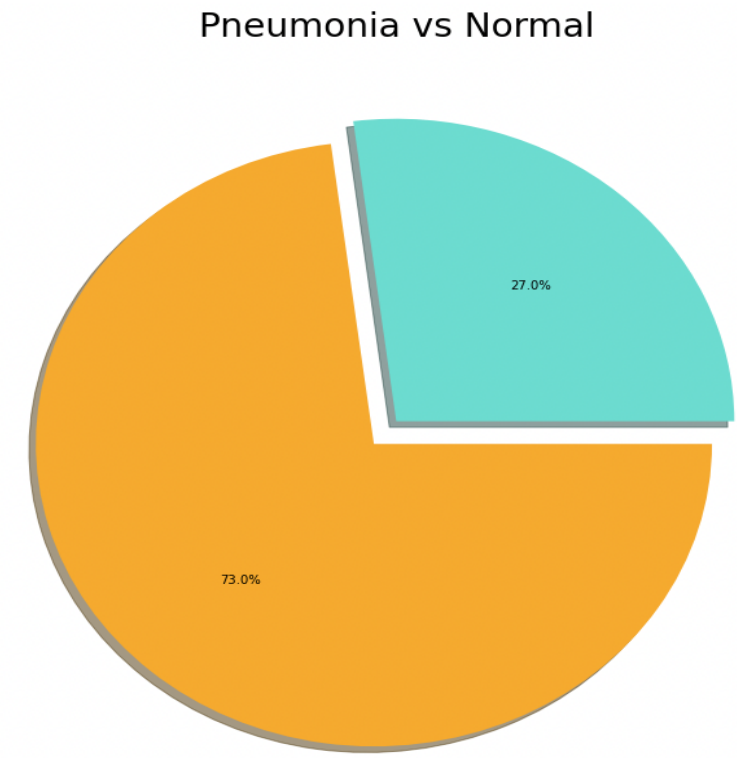

Figura 5.1: Exemplu pie-chart care reprezinta distributia datelor sub formă procentuală.

Fiecare grafic si poza care se refera la date din acest proiect au fost realizate folosind libraria matplotlib si metodele puse la dispozitie de aceasta.

5.1.6 OpenCV

Principala librărie folosită în procesare de imagini. OpenCV este disponibilă pentru limbajele Python, Java sau C++, caştigând astfel mult în popularitate. Conţine peste 2500 de algoritmi optimizaţi exclusiv pentru procesarea de imagini. Printre utilizările acestor algoritmi se numără:

- Detecţie şi recunoaştere facială.
- Identificare de obiecte.
- Clasificarea gesturilor umane din materiale video.
- Urmaiea mişcării unor obiecte (cu aplicaţii în conducerea automată a maşinilor).

5.2 Arhitectura sistemului

Sistemul este împartit în trei componente principale:

- Pregătirea şi pre-procesarea datelor
- Extragerea caractersiticitlor
- Clasificare

Aceste componente principale au la rândul lor mai multe sub-componente care au fost descrise pe larg in secţiunile şi capitolele anterioare. O reprezentare grafică a modului cum componenentele aplicaţiei interactionează unele cu altele este prezentă în figura 5.2.

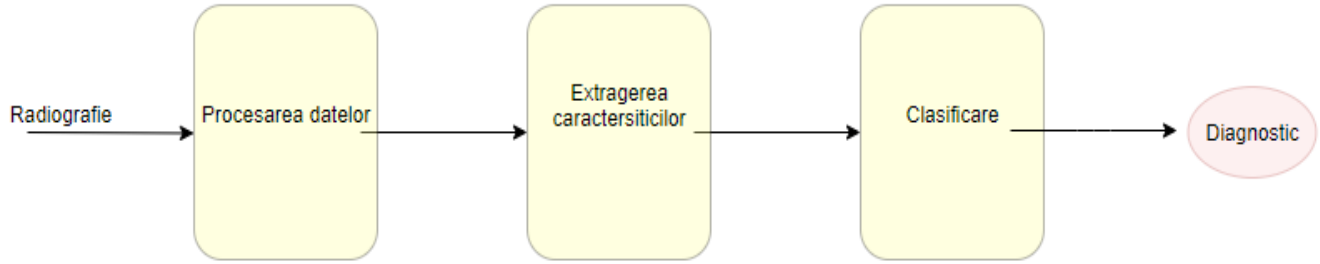

Figura 5.2: Componentele principale ale aplicaţiei

Cea mai importantă componentă prezentată în figura precedentă este cea de Clasificare, aceasta oferind practic verdictul (diagnosticul) final. Ordinea secvenţială a distribuirii blocurilor componente are ca scop evidenţierea faptului ca ultima componentă se foloseşte de rezultatele produse de cele premergătoare ei pentru a produce un rezultat final.

Sub-componentele fiecarui bloc din figura 5.2, precum şi detalii privind construirea şi customizarea acestora, au fost prezentate în secţiunile aferente fiecaruia. Componentele de Extragerea caractersiticilor si Clasificare se află în interiorul Reţelei Neuronale Convoluţionale.

Setul de date

Setul de date este împarţit în 3 directoare (antrenare, testare şi validare) şi fiecare director este la rândul său împarţit în sub-directoare pentru fiecare categorie a radiografiei (Pneumonia/Normal). Sunt disponibile în total 5863 de imagini radiografice (format JPEG), divizate în cele două clase amintite: Pneumonia şi Normal.

La centrul de medicină pediatrică din Guangzhou au fost facute radiografii copiilor cu varsta între 1 şi 5 ani. Aceste imagini reprezintă radiografii efectuate la nivelul toracelui (atât anterior cât şi posterior). Când un pacient cu varsta menţionantă anterior era primit de către spital, acestuia i se efectuau mai multe teste şi probe (incluzând bineinţeles o scanare anterioară şi posterioară cu raze X), de unde au rezultat şi aceste imagini.

Toate radiografiile au trecut printr-un proces de control al calitaţii, fiind eliminate toate acele imagini care nu erau clare sau nu satisfăceau normele de calitate (erau deteriorate). Diagnosticul (categoria fiecarei imagini) a fost verificat de către 2 experţi pentru a se asigura ca datele sunt 100% corecte şi gata pentru a putea fi folosite intr-un algoritm sau o soluţie de Inteligenţă Artificială cum este şi proiectul de faţă.

Un lucru important care trebuie avut în vedere când vine vorba de etichetarea manuală este eroarea survenită în urma participării factorului uman. Acest aspect poate duce deci la erori importante în setul de date (etichetarea greşită a unei radiografii pulmonare), lucru ce implică în mod direct o eroare care nu trebuie negljiată.

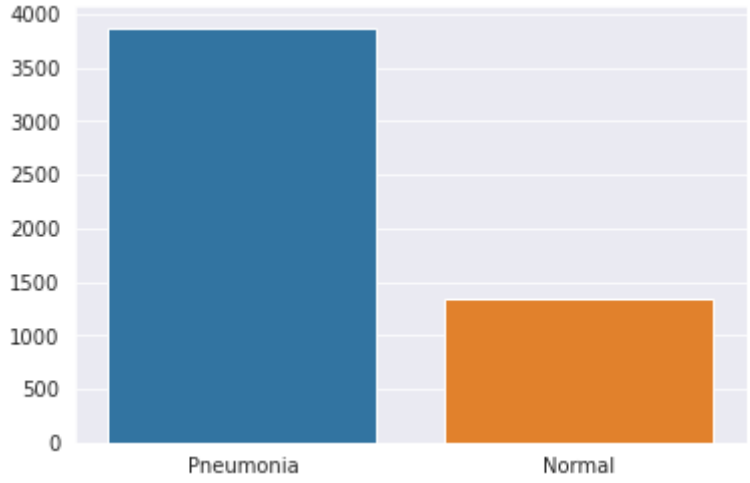

Figura 5.3: Distribuţia setului de date

Specificat şi anterior, în figura 5.3 se poate observa discrepanţa datelor între cele doua categorii de imagini, motiv pentru care au fost efectuate operaţiile de augmentare şi balansare a setului de date pentru reduce diferenţa dintre cele două categorii.

Extragerea caracteristicilor

Componenta care se ocupă cu extragerea trăsăturilor se afla în Reţeaua Neuronală Convoluţională şi reprezintă cea mai mare parte a acesteia. Privit ca un model de tip *black-box*, această componentă primeşte ca şi intrare o imagine şi harta caractersiticilor a acesteia (o mască) şi returnează numele caractersiticii urmat de valoarea 0 sau 1, în functie de cum este clasificaţa acea trăsătură (practic este un dicţionar de forma: {trăsătură: 0/1}).

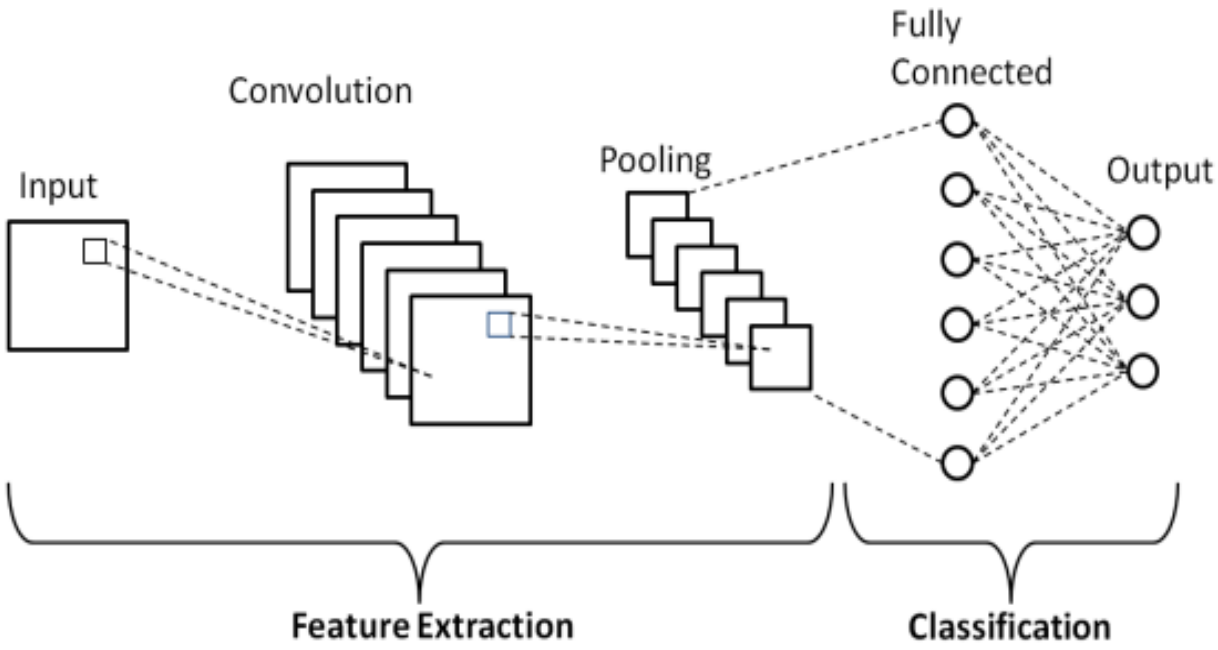

Figura 5.4: Diagrama schematică a unei Reţele Neuronale Convoluţionale [23]

Deoarece o diagrama a claselor nu se potriveşte tipului de solutie aleasa, figura 5.4 reprezintă o diagramă de analiză a proceselor din componenta de extragere a caracter-siticilor (*feature extraction*). Se observă layerele care au un aport mai mare în extragere, precum şi cele care au o contribuţie mai mare în partea de clasificare. Funcţionalitatea acestor blocuri a fost prezentata în prima parte a acestui capitol.

Clasificarea

Decizia finală sau clasificarea se face pe baza rezultatelor furnizate de procesul de extragere a trăsăturilor. Diagnosticul final va fi unul binar (plămâni sănătoşi sau plămâni bolnavi), din acest motiv se foloseşte o funcţie de activare care produce ca şi output doar doua posibile valori: 0 sau 1.

Funcţia Sigmoid este funcţia de activare folosita la ultimul pas al reţelei neuronale pentru a face clasificarea propriu-zisa, deoarece este o funcţie binară şi se preteaza pe nevoile soluţiei de fata: o clasificare binară a radiografiilor. Ieşirea ultimelor straturi ale reţelei este redimensionata într-un vector al caractersiticilor uni-dimensional (*1D feature vector*). Pe baza informaţiilor regăsite în acest vector, funcţia de activare realizează clasificarea radiografiei:

- 0 - plămâni sănătoşi
- 1 - plămâni afectaţi de pneumonie

În continuare (Figura 5.5) regăsim layerele şi parametrii reţelei neuronale alese ca şi model final în cadrul solutiei de faţă.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_35 (Conv2D)           (None, 148, 148, 32)      320
_________________________________________________________________
max_pooling2d_35 (MaxPooling (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_36 (MaxPooling (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_37 (Conv2D)           (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_37 (MaxPooling (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_38 (Conv2D)           (None, 15, 15, 256)       295168
_________________________________________________________________
max_pooling2d_38 (MaxPooling (None, 7, 7, 256)         0
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 5, 5, 512)         1180160
_________________________________________________________________
max_pooling2d_39 (MaxPooling (None, 2, 2, 512)         0
_________________________________________________________________
flatten_7 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_22 (Dense)             (None, 256)               524544
_________________________________________________________________
dense_23 (Dense)             (None, 128)               32896
_________________________________________________________________
dense_24 (Dense)             (None, 64)                8256
_________________________________________________________________
dropout_9 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_25 (Dense)             (None, 1)                 65
=================================================================
Total params: 2,133,761
Trainable params: 2,133,761
Non-trainable params: 0
```

Figura 5.5: Layerele reţelei neuronale alese

Un exemplu de predicţie se regăseşte în Figura 5.6. După antrenarea modelului şi vizualizarea performanlor acestuia, se decide daca acesta reprezintă sau nu forma finală în funcţie de performanţele obtinute. Dacă acestea din urmă satisfac obectivele propuse,

modelul poate fi salvat şi se poate oricând face o predicţie folosind metoda *predict_classes()*, care primeşte ca şi input o imagine sau un set de imagini, furnizând ca și rezultat clasa prezisă.

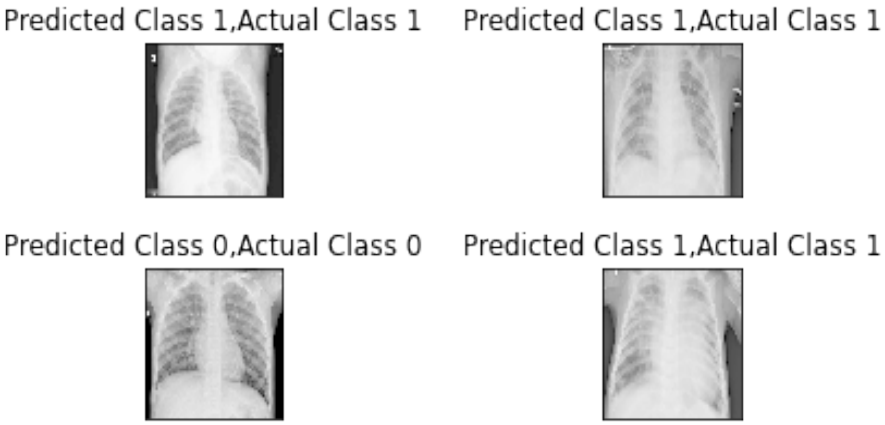

Figura 5.6: Exemplu de predicţie

Capitolul 6

Concluzii

6.1 Rezumatul contribuţiilor proprii

Scopul proiectului de faţă a fost dezvoltarea unei soluţii pentru procesarea şi clasificarea imaginilor medicale, mai precis ale unor radiografii pulomnare, într-una dintre urmatoărele clase: plămâni sănatoşi sau plâni bolnavi. Această clasificare binară determină dacă un pacient are sau nu pneumonie, bazat pe radiografia pulmonară. În atingerea acestui scop au fost definite 3 mari obiective: importarea şi pre-procesarea imaginilor, extragerea trăsăturilor folosind reţele neuronale şi clasificarae propriu-zisă bazată pe trăsăturile extrase.

- În cazul primului obiectiv am propus un algoritm de pre-procesare folosind corectîa Gamma (*Gamma Correction*) pentru reglarea intensităţii luminoase a imaginilor. Despre setul de date folosit s-a discutat în capitolul anterior, acesta fiind disponibil online şi gratuit pentru oricine doreşte să experimenteze sau să construiască o soluţie. Pentru augmentarea (îmbunătăţirea) acestui set de date am folosit *ImageDataGenerator* din librariă Keras, modificând parametrii astfel încât să funcţioneze cu imaginile radiografice.

- În ceea ce priveşte extragerea caractersiticilor, am folosit două arhitecturi de reţele neuronale pentru a putea trage o concluzie asupra celei mai performante soluţii care se preteaa pe clasificarea radiografiilor. Amândouă reţelele folosesc modelul secvenţial descris şi anterior. Aceste reţele le-am construit folosind modelul *Sequential* pus la dispoziţie în Keras, după care am adaugat layerele specifice unei reţele neuronale convoluţionale. Pentru cele doua soluţii am folosit un număr diferit de layere cu parametrii diferiţi.

- Clasificarea a fost facută folosind o functie de activare la finalul reţelei neuronale, functie care primeşte rezultatele layerelor si pe baza acestora calculează clasa din care face parte imaginea. Deoarece am avut de realizat o clasificare binara, am ales

sa folosesc functia Sigmoid ca şi funcţie de activare, pentru că are un output binar şi uşor de interpretat. Am folosit aceasta pentru ambele modele create.

6.2 Analiza rezultatelor

Prima reţea neuronală construită, unde ai fost realizate 10 epoci de antrenare, a produs o acurateţe de 85%. Acesta poate parea un număr mare la început, însa în realitate nu este un rezultat deloc satisfăcător.

Pentru cea de-a doua reţea, am reuşit să reduc numărul parametrilor neantrenabili. Ca şi rezultate, aceasta are o acurateţe de 95%. Figura 5.5 evidenţiaza layerele şi parametrii găsiţi de acestea pentru cea de-a doua reţea neuronală folosită.

Precizia pentru clasa 1 (plămâni bolnavi) este detectată cu 97% exactitate. Acesta este un aspect foarte important în cazul unei probleme din domeniul medical, deoarece se doreşte să fie identificată clasa bolnavă cu foarte mare precizie, chiar dacă unii pacienţi sănătoşi ajung să fie reexaminaţi din cauza unei greşeli, decât un pacient bolnav să fie clasificat ca şi sănătos.

Precizie şi F1-score

Precizia este raportul dintre predicţiile corecte pozitive (clasa cu plămâni sănatoşi) şi totalul predicţiilor pozitive (nu neapărat corecte).

F1-score ia în considerare atât rezultatele fals-pozitive cât şi cele fals-negative. Acurateţea singură poate fi luata în considerare dacă rezultatele fals-pozitive şi fals-negative au acelaşi cost. În cazul nostru, ne dorim să eliminăm fals-negativele, deoarece asta ar insemna că un pacient bolnav ar putea fi pus în categoria celor sănătoşi, lucru de evitat într-o situaţie medicală.

Clasa	Precizie	F1-score
0	0.89	0.90
1	0.97	0.94

Tabelul 6.1: Rezultate de Precizie si F1-score

Un F1-score foarte bun (Tabelul 6.1), inseamnă că modelul are puţine cazuri fals-pozitive şi fals-negative, un lucru esenţial pentru o solutie care ar putea avea o implicare directă în salvarea vieţilor umane.

Având în vedere rezulatele prezentate mai sus, este evident că modelul care a produs o acurateţe de 94% este cel cu care se va continua mai departe într-o eventuală dezvoltare a acestei aplicaţii, fiind mult mai performant decât cel pe care îl succede.

6.3 Direcţii de dezvoltare

Prin dezvoltări ulterioare ale proiectului de faţă se pot lua în considerare două mari aspecte: îmbunătăţirea modelului şi integrarea modelului în cadrul unei aplicaţii/unui proiect mai mare. Pentru cel dintai caz, se pot lua în vedere urmatoarele:

- Schimbarea din clasificare binară în clasificare pentru mai multe clase (mai multe tipuri de boli pulmonare).
- Antrenarea pe un set de date mai mare, lucru care ar duce la o precizie mult mai ridicată al unui eventual model
- Adaugarea unor date referitoare la pacient: boli pulmonare suferite în trecut, istoric medical relevant etc.

În ceea ce priveşte partea de integrare a modelului, dezvoltările ar putea fi următoarele:

- Integrarea într-o aplicaţie care să reţină un istoric medical al pacientului, unde se vor regăsi radiografii anterioare şi rezultatele acestora.
- Dezvoltarea unui portal web sau a unei aplicaţii desktop pentru medici specialişti. Aceasta aplicaţie ar rula in partea de back-end modelul şi ar primi o prezicere pentru radiografia încărcată.

În concluzie, având la dispoziţie un set de date relativ mic din punct de vedere al numărului imaginilor, am construit o soluţie cu peste 90% acurateţe în detecţia pneumoniei.

Soluţia propusă îmbină algoritmii clasici de procesare ai imaginilor cu algoritmii mai moderni care au la bază reţelele neuronale convoluţionale. Fiind un sistem modularizat, se poate oricând încerca alt model pentru antrenare.

Bibliografie

[1] P. Rajpurkar, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,' no. 1, pp. 10–15, 2017.

[2] "Johns hopkins coronavirus resource center. covid- 19,' 2020.

[3] D. Bull, *Communicating Pictures, A Course in Image and Video Coding.* Prentice Hall, 2014, vol. 1.

[4] M. Jogin, M. Mohana, M. Madhulika, G. Divya, R. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (cnn) and deep learning,' 05 2018, pp. 2319–2323.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions,' in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition,' in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[7] R. W. R.C. González, *Digital Image Processing.* Bellingham, Wash, 2002, vol. 1.

[8] C. Poython, *The rehabilitation of Gamma.* Academic Press, 2007, vol. 1.

[9] C. M. Bishop, *Neural Networks for Pattern Recognition.* -, 1995, vol. 1.

[10] S. M. I. S. S. Luka Račić, Tomo Popović, "Pneumonia detection using deep learning based on convolutional neural network,' *2021 25th International Conference on Information Technology*, vol. 51, no. 1, pp. 15–20, 2021.

[11] P. Shukla and K. Bhowmick, "To improve classification of imbalanced datasets,' in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, pp. 1–5.

[12] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media, Inc, 2019, vol. 1.

[13] A. Karpathy, "Neural networks part 1: Setting up the architecture. notes for cs231n convolutional neural networks for visual recognition,' *Stanford University*.

[14] O. e. a. Russakovsky, "Imagenet large scale visual recognition challenge,' *International Journal of Computer Vision*, 2014.

[15] "Artificial neural network,' available at https://en.wikipedia.org/wiki/Artificialneuralnetwork.

[16] G. Baltruschat, Nickisch, "Comparison of deep learning approaches for multi-label chest x-ray classification,' *Sci Rep*, vol. 9, 4 2019.

[17] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks,' 12 2016.

[18] T. Bao, A. Zaidi, S. Xie, and Z. Zhang, "Surface-emg based wrist kinematics estimation using convolutional neural network,' in *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2019, pp. 1–4.

[19] D. Jung, J.-w. Son, and S.-J. Kim, "Shot category detection based on object detection using convolutional neural networks,' 02 2018, pp. 36–39.

[20] A. Karpathy, "Cs231n convolutional neural networks for visual recognition,' 2018.

[21] L. A. dos Santos, "Artificial inteligence,' 2018, available at https://legacy.gitbook.com/book/leonardoaraujosantos/artifcial-inteligence.

[22] J. Shi, J. Dang, M. Cui, R. Zuo, K. Shimizu, A. Tsunoda, and Y. Suzuki, "Improvement of damage segmentation based on pixel-level data balance using vgg-unet,' *Applied Sciences*, vol. 11, pp. pp.518.1–17, 01 2021.

[23] Phung and Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,' *Applied Sciences*, vol. 9, p. 4500, 10 2019.