

**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAŞI
FACULTATEA DE INFORMATICĂ**



LUCRARE DE LICENȚĂ

LungVision

propusă de:

Pasat Tudor Cosmin

Sesiunea: iulie, 2023

Coordonator științific
Drd. Olariu Florin

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAŞI
FACULTATEA DE INFORMATICĂ

LungVision

Pasat Tudor Cosmin

Sesiunea: iulie, 2023

Coordonator științific
Drd. Olariu Florin

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele Drd. Olariu Florin

Data 20.06.2023 Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnata/ul Pasat Tudor Cosmin cu domiciliul în strada 14 Iulie, bloc L3, scara A, apartament 10, născută la data de 24.07.2001, identificată prin CNP 5010724375201, absolventă a Universității ”Alexandru Ioan Cuza” din Iași, Facultatea de Informatică, specializarea Română, promoția 2020, 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: *LungVision* elaborată sub îndrumarea dl. Drd. Olariu Florin, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul sau în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consumând 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 diplomă 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, 20.06.2023

Semnătură student,

DECLARAȚIE PRIVIND ORIGINALITATEA ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul *LungVision* este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau din străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursa, imaginile etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, 20.06.2023

Absolvent Pasat Tudor Cosmin

DECLARAȚIE DE CONSUMĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul *LungVision*, codul sursa al programelor și celealte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea "Alexandru Ioan Cuza" din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursa, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 20.06.2023

Absolvent Pasat Tudor Cosmin

Abstract

Lucrarea de licență se concentrează pe proiectarea și implementarea unui sistem de predicție medicală, folosind învățarea automată, pentru asistarea personalului medical în procesul de diagnosticare a uneia dintre bolile pulmonare: pneumonie, tuberculoză, covid-19. Sunt prezentate atât detalii ale procesului de construire ale unui model de rețea neuronală convoluțională potrivit, cât și arhitectura și implementarea unei aplicații web, ce servește ca interfață. Pentru a realiza acest deziderat am folosit o arhitectură de forma client-server. un tech stack diversificat, incluzând Angular pentru frontend, Java pentru backend și Python pentru dezvoltarea și interacțiunea cu modelele de învățare automată.

Cuprins

Abstract	5
Motivație	8
Introducere	9
Capitolul 1: Aplicații Similară	10
1.1 Aidoc	10
1.2 Nanox Vision	11
1.3 Google Health	12
Capitolul 2: Noțiuni Utilizate	13
2.1 Boli pulmonare	13
2.2 Inteligența Artificială	14
2.2.1 Învățare Automată	15
2.2.2 Rețele Neuronale Artificiale	16
2.2.3 Rețele Neuronale Convoluționale	17
2.2.3.1 Stratul de intrare	17
2.2.3.2 Stratul de convoluție	17
2.2.3.3 Stratul de reducere a dimensiunii	18
2.2.3.4 Stratul complet conectat	18
Capitolul 3: Tehnologii Utilizate	20
3.1 Python	20
3.1.2 TensorFlow	20
3.1.2 Albumentations	21
3.1.2 Flask	21
3.2 Angular	21
3.2.1 PrimeNG	22
3.2.2 Express	22
3.3 Java	22
3.2.1 Spring Boot	22
3.2.2 JPA	23
3.2.3 Hibernate	23
3.4 PostgreSQL	24
3.4 Cloudinary	24
3.4 Heroku	25
Capitolul 4: Dezvoltarea Modelului de CNN	26
4.1 Setul de date	26
4.1.1 Introducere	26
4.1.2 Preprocesarea datelor	26
4.2 Descrierea modelelor	28
4.2.1 Modelul 1	28
4.2.1 Modelul 2	30
4.2.1 Modelul 3	31
4.2.1 Modelul 4	33
4.3 Concluzii	34
Capitolul 5: Arhitectură și implementare	35

5.1 Arhitectura sistemului	35
5.2 Arhitectura Serverului Principal	36
5.3 Arhitectura Bazei de Date	37
5.4 Detalii de Implementare	38
5.4.1 Pagina Principală	38
5.4.2 Pagina de Analiză și Predicție	39
5.4.3 Istoric și Vizualizarea Datelor	41
5.4.5 Gestionarea Erorii 404	42
5.4.6 Internaționalizare	42
5.4.7 Deploy	43
Capitolul 6: Scenarii de utilizare	44
6.1 Ajutor în Procesul de Diagnosticare	44
6.2 Confirmarea Diagnosticului	44
6.3 A Doua Opinie	44
6.4 Urmărirea Progresului	44
6.5 Instrument pentru Învățare	44
Contribuții	46
Concluzii	47
Bibliografie	48

Motivație

Milioane de oameni din întreaga lume suferă de boli pulmonare, ceea ce reprezintă o problemă serioasă de sănătate publică [41]. Detectarea unor astfel de boli pulmonare, precum Pneumonie, Tuberculoză și Covid-19, într-un timp cât mai scurt este esențială pentru formarea unui plan de tratament cât mai eficient posibil.

Conform World Health Organization(WHO), la nivel mondial, unul din cinci decese ale copiilor cu vîrste între 1 și 5 ani au drept cauză pneumonia [39]. De asemenea, în cei peste trei ani de cand a apărut, Covid-19 a reprezentat o afecțiune deosebit de gravă pentru milioane de oameni la nivel global, fiind și cauza decesului a aproape 7 milioane de oameni [40]. Tuberculoza este la fel de periculoasă, afectând doar în 2021 aproximativ 10.6 milioane de oameni, din care au rezultat 1.6 milioane de decese [42].

În ultimii ani, tehnologia radiologică a evoluat rapid, oferind o modalitate precisă și non-invazivă de a identifica diferite boli pulmonare prin intermediul scanărilor cu raze X. Pentru evaluarea acestor este nevoie de expertiza unui specialist radiolog, fiind nevoie de cunoștințe avansate în anatomie, fiziologie și patologie. Problema cel mai des întâmpinată este disponibilitatea unor astfel de specialiști, în special în țările mai puțin dezvoltate.

Scopul acestei lucrări constă în proiectarea și implementarea unui model de Rețea Neuronală Convoluțională care primește o scanare cu raze X a pieptului ca intrare și returnează una dintre cele patru clase: Sănătos, Pneumonie, Tuberculoză și Covid-19. Acest model este integrat într-o aplicație web, pentru a fi folosită împreună cu expertiza medicală a unui doctor, în procesul de diagnosticare și formare a unui plan de tratament.

Introducere

Lucrarea propusă aduce în prim plan analiza modului în care aplicația LungVision încearcă să abordeze și să rezolve problemele menționate mai sus. **Capitolul 1** prezintă câteva aplicații similare și valoarea pe care o oferă fiecare.

Capitolul 2 propune o expunere detaliată a noțiunilor fundamentale utilizate în capitoalele următoare, furnizând astfel un context teoretic necesar pentru înțelegerea în profunzime a subiectului abordat. **Capitolul 3** se concentrează pe prezentarea tehnologiilor utilizate în cadrul aplicației, ilustrând astfel nivelul de expertiză și conștientizare tehnică a echipei de dezvoltare.

Capitolul 4 oferă o analiză a procesului de construire a modelelor de rețele neuronale, printre care o descriere detaliată a setului de date utilizat, metodele aplicate pentru augmentarea datelor, precum și structura și arhitectura modelelor în sine. Acest capitol prezintă și o comparație între diversele modele propuse, evidențiind astfel avantajele și dezavantajele fiecărui.

În continuare, **Capitolul 5** furnizează informații relevante cu privire la arhitectura generală a sistemului implementat, ilustrând în detaliu modul în care interfața web și serverele au fost dezvoltate și integrate.

Capitolul 6 prezintă o serie de scenarii de utilizare, conturand astfel beneficiile pe care aplicația LungVision le aduce în contextul utilizării reale.

În final, lucrarea se încheie cu o secțiune de **concluzii**, care prezintă principalele rezultate obținute pe parcursul întregului proces de dezvoltare și analiza. Aceasta prezinta, pe scurt, contribuțiile aduse de aplicația LungVision și importanța acesteia în domeniul medical.

Capitolul 1: Aplicații Similare

În secțiunile următoare sunt prezentate câteva aplicații similare cu LungVision, folosite deja în domeniul medical. Toate imaginile sunt preluate din aplicațiile prezentate.

1.1 Aidoc¹

Aidoc este o platformă medicală de imagistică ce folosește modele inteligență artificială, inclusiv rețele neuronale convolutionale pentru a ajuta personalul medical în procesul de diagnosticare a diferite condiții medicale. Aidoc are în prezent cele mai multe aprobări primite din partea FDA pentru algoritmi de învățare automată în domeniul medical, printre care se enumeră modele ce pot detecta embolisme pulmonare, fracturi cervicale, hemoragii intracraiene.

Algoritmii produși de Aidoc ajută în fiecare luna peste 1 milion de pacienți din toată lumea, aplicația fiind integrată în peste 1000 de spitale la nivel global și conform statisticilor, salvează în medie 59 de minute per pacient.

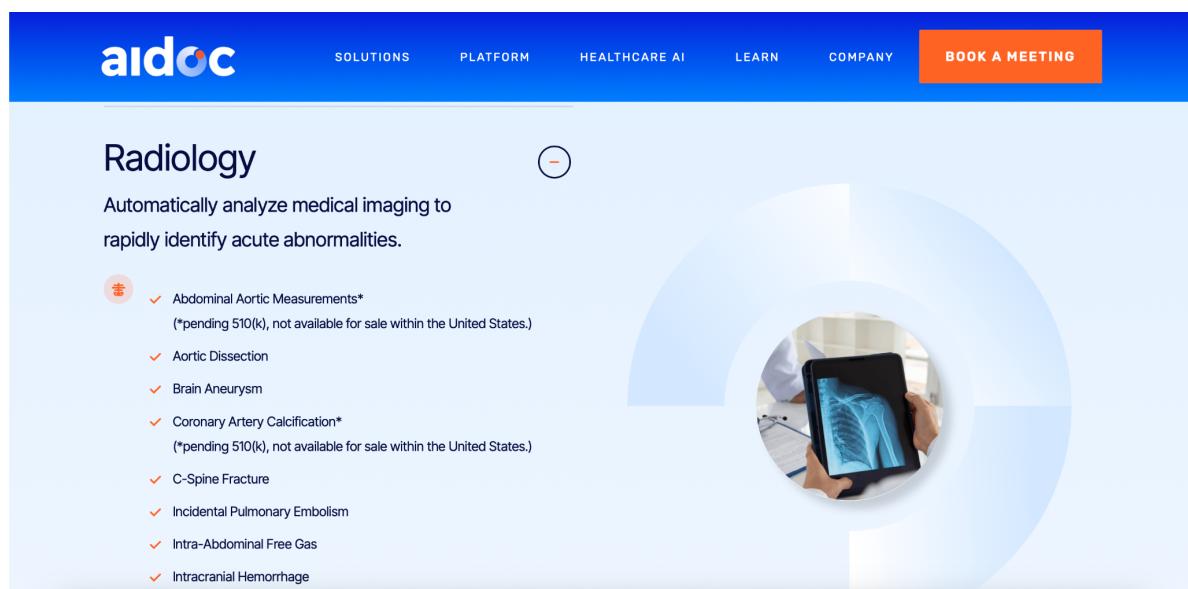


Figura 1: Câteva dintre funcționalitățile platformei Aidoc

¹ [2]

1.2 Nanox Vision²

Nanox are ca scop construirea unei infrastructuri medicale de imagistică la nivel global. Urmăresc formarea unui ecosistem complet, pornind de la producția dispozitivelor medicale de imagistica, până la formarea unui sistem în cloud bazat pe inteligență artificială pentru analiza imaginilor.

În **Figurile 2 și 3**, putem vedea că Nanox are deja în producție algoritmi aprobați de FDA, ce pot detecta afectiuni precum calcifierea arterelor coronare și a osteoporozei. De asemenea, în **Figura 4** se evidențiază că au în procesul de dezvoltare și un algoritm pentru identificarea pacienților cu ficat gras.



**HealthCCSng
cardiac solution**

FDA CLEARED | CE

TECHNOLOGY

AI algorithm that automatically detects coronary artery calcification (CAC) to help identify patients at risk of coronary artery disease without a need for discrete imaging.

DISEASE STATE OF FOCUS

Coronary artery disease (CAD)

**HealthOST
Bone solution**

FDA CLEARED *

TECHNOLOGY

Algorithm that automatically detects vertebral compression fractures and low bone mineral density to help identify patients at risk of musculoskeletal disease without a need for discrete imaging.

DISEASE STATE OF FOCUS

Osteoporosis



Figura 2: Detectarea Calcifierii
Arterelor Coronare

Figura 3: Detectarea Osteoporozei



Fatty Liver solution

IN DEVELOPMENT

TECHNOLOGY

AI algorithm is being developed to identify patient's with fatty liver without a need for discrete imaging

DISEASE STATES OF FOCUS

Hepatic Steatosis, NASH

PRODUCT DESCRIPTION

Figura 4: Identificarea Pacienților cu
Ficat Gras

² [26]

1.3 Google Health³

Cei de la Google au construit o echipă de cercetare formată din experți din diferite domenii pentru a construi și testa modele de inteligență artificială. Au ca scop dezvoltarea pe termen lung a potențialului tehnologiei în domeniul medical, astfel încât să diminueze impactul lipsei de doctori la nivel mondial și a accesului limitat la unelte de imagistică medicală și de diagnosticare în anumite parti ale lumii.

Printre modelele deja dezvoltate, putem vedea în **Figura 5** că au dezvoltat deja modele ce pot detecta afecțiuni acute la nivelul rinichilor cu până la 48 de ore înaintea medicilor. Un alt proiect de menționat, este sevențierea ADN-ului pentru detectarea pacienților cu afecțiuni genetice, ce au un risc ridicat de dezvoltare a cancerului la sân.

Using AI to give doctors a 48-hour head start on life-threatening illness

In this research in [Nature](#), we demonstrated how artificial intelligence could accurately predict acute kidney injuries (AKI) in patients up to 48 hours earlier than it is currently diagnosed. Notoriously difficult to spot, AKI affects up to one in five hospitalized patients in the US and UK, and deterioration can happen quickly.
[Read the article](#)

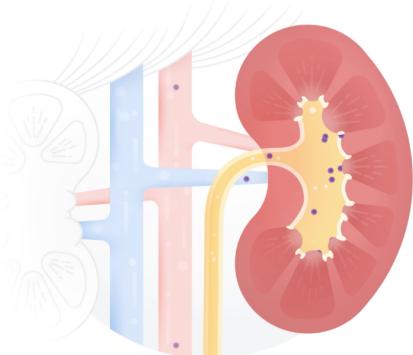


Figura 5: Detectarea afecțiunilor cronice ale rinichilor

³ [1]

Capitolul 2: Notiuni Utilizate

2.1 Boli pulmonare

Aplicația web LungVision urmărește realizarea unor scanari medicale cu raze X pentru a determina dacă pacientul este sau nu bolnav. Bolile pulmonare analizate sunt:

- Pneumonie
- Covid-19
- Tuberculoza



Figura 6: Plămâni sănătoși

[7]



Figura 7: Plămâni afectați de pneumonie

[7]

În **Figura 6**, este prezentată o scanare cu raze X a unor plămâni sănătoși. Se poate observa o culoare uniformă pe toată suprafața lor, în timp ce în **Figura 7**, sunt o pereche de plămâni afectați de pneumonie bacteriană. Aceasta se caracterizează de obicei printr-o decolorare parțială a unuia dintre lobi, în acest caz în plamanul drept, partea superioară [7].

În **Figura 8**, putem vedea o imagine ce prezintă o consolidare în zona superioară a plămânlui drept cu mărire ipsilaterală a hilului datorită limfadenopatiei. Consolidarea se referă la o opacitate sau o zonă de densitate crescută în plămân, care indică prezența unei afecțiuni sau a unei infecții și limfadenopatia înseamnă mărirea sau inflamația ganglionilor limfatici. Aceste sunt trăsăturile specifice tuberculozei [22].



Figura 8: Plămâni afectați de tuberculoză
[22]

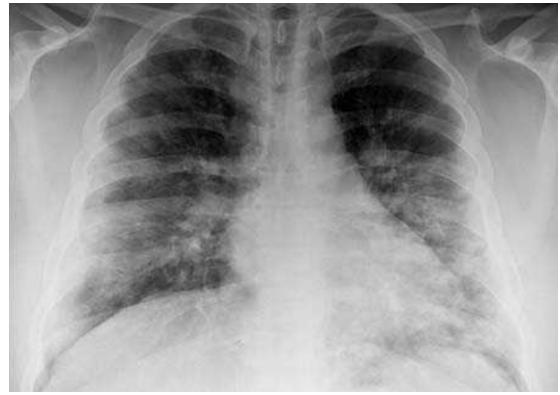


Figura 9: Plămâni afectați de covid-19
[23]

În **Figura 9**, sunt prezente opacități pulmonare bilaterale și patchy în periferia bazelor pulmonare. Aceste condiții pot duce la diagnosticarea pacientului cu pneumonie covid-19. Patchy se referă la prezența unor zone discontinue în imaginile pulmonare. Opacitățile pulmonare sunt zone în care transparența plămânlui este redusă sau în care apar consolidări sau alte modificări [23].

2.2 Inteligența Artificială

Inteligența umană este o trăsătură mentală care include capacitați de învăță noi capacitați prin experiență, de a se adapta la circumstanțe noi, de a înțelege și gestiona idei abstracte, de a înțelege și gestiona idei abstracte și de a aplica cunoștințele pentru a influența mediul înconjurător. [35]

Inteligența artificială urmărește replicarea inteligenței umane, astfel încât un computer sau un robot controlat de un computer să aibă abilitatea de a îndeplini sarcini asociate cu ființele inteligente. De la apariția computerului digital în anii 1940, au apărut numeroase situații în care astfel de computere pot fi programate pentru a rezolva sarcini complexe, precum descoperirea unor teoreme matematice. Momentan, suntem încă limitați de tehnologie, chiar dacă puterea computațională a crescut considerabil în ultimii ani, și nu există programe care să fie la nivelul inteligenței umane. Pe de altă parte, au fost dezvoltate unele modele de inteligență artificială care pot depăși performanța unor experți în anumite sarcini specifice, precum găsirea unor diagnostice medicale. [10]

În 1950, matematicianul Alan Turing a propus Testul Turing, ce urmărește măsurarea capacitații unui computer de a manifesta o inteligență care nu poate fi diferențiată de cea umană. Testul presupune un evaluator uman, și 2 candidați, un computer și un om. Scopul

testului este de a determina dacă evaluatorul poate diferenția conversația purtată cu o mașină, de cea purtată cu o persoană.

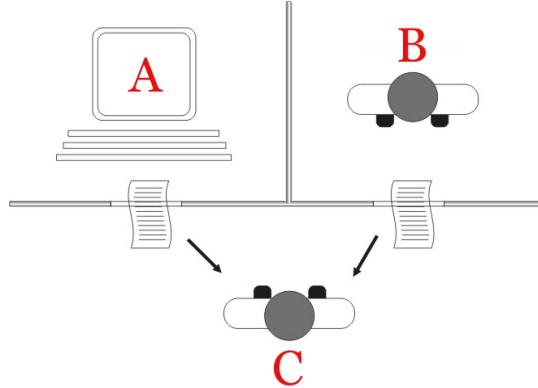


Figura 10: Testul Turing⁴

2.2.1 Învățare Automată

“Un program de calculator este considerat a învăța din experiența E cu privire la o anumită clasă de sarcini T și măsurarea performanței P, dacă performanța sa la sarcinile din T, măsurată prin P, se îmbunătățește cu experiența E.”, definiție preluat din [25].

Învățarea automată este folosită în scopul de a putea analiza un set de date mai eficient. În ultimii ani tehnologia a avansat la nivel exponențial, fiind folosită în diverse domenii, și odată cu ea a crescut și volumul de date complexe ce trebuie analizate. Având un flux constant de date noi, în unele cazuri ar fi imposibil ca o echipă de oameni să țină pasul în interpretarea acestora. Din acest motiv a apărut învățarea automată, care poate analiza un volum mare de date într-un timp considerabil mai scurt. De asemenea, în unele cazuri o persoană nu poate extrage informații din date la fel de bine ca un computer.

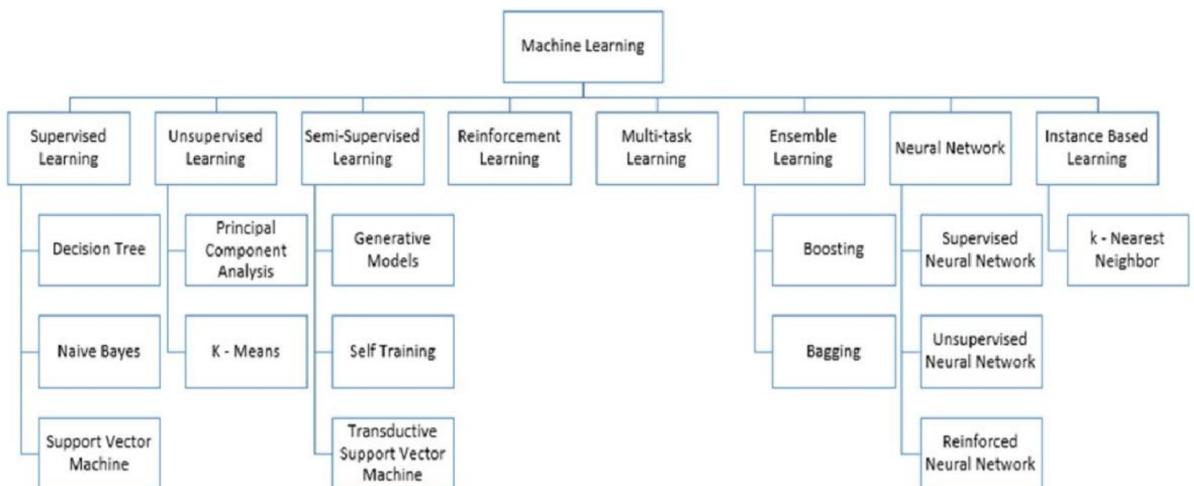


Figura 11: Clasificarea algoritmilor de învățare automată⁵

⁴ [38]

⁵ [24]

După cum putem vedea în **Figura 11**, există diverși algoritmi de învățare automată, dar în această lucrare am folosit doar algoritmi de Rețele Neuronale Supervizate.

2.2.2 Rețele Neuronale Artificiale

Modul de funcționare al unui sistem nervos organic, precum creierul uman, reprezintă principala sursă de inspirație pentru rețelele neuronale artificiale(ANNs). Un proces distribuit de învățare este folosit the ANNs, ce sunt formate dintr-un număr mare de noduri computaționale interconectate cunoscute drept neuroni, pentru a putea invata din datele de intrare și pentru a produce un rezultat cât mai precis. [29]

Structura cea mai simplă a unei ANN este reprezentată de un strat de input, un strat ascuns și un strat de ieșire. Datele de intrare, reprezentate de cele mai multe ori printr-un vector multidimensional, sunt introduse pe rând în stratul de intrare, care le distribuie stratului ascuns. Stratul ascuns folosește datele primite de la stratul anterior decide ce schimbare internă poate face pentru a obține un rezultat cat mai apropiat de cel dorit, astfel formandu-se procesul de învățare automată. Înlănțuirea mai multor astfel de straturi ascunse este cunoscută ca învățare în profunzime. [29]

Învățarea automată supervizată implică un proces de învățare folosit un set de date de intrare format din valori pre etichetate, care servesc drept ţinte. Fiecare punct de date este format dintr-un set de valori de intrare și una sau mai multe valori de ieșire specificate. Scopul acestui antrenament este de a reduce eroarea generată de clasificarea modelului prin determinarea exactă a valorii de ieșire pentru fiecare exemplu, astfel având o precizie cat mai mare la clarificarea unor date cu etichetă necunoscută.

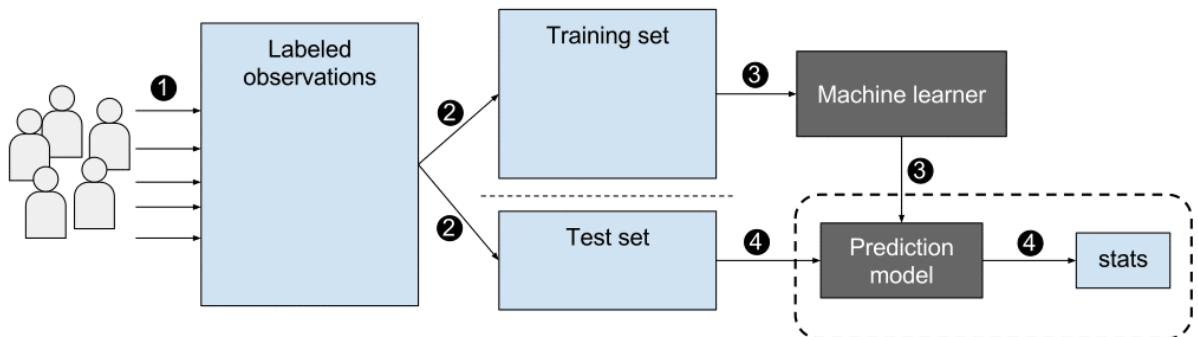


Figura 12: Diagrama procesului de construire al unei rețele neuronale supervizate⁶

⁶ [33]

2.2.3 Rețele Neuronale Convoluționale

O rețea neuronală conoluțională(CNN) este un tip de rețea neuronală supervizată care se ocupă în special de prelucrarea imaginilor, dar și a altor date cu structură similară, precum sunetele audio.

Introduse în [21], CNN-urile sunt folosite pentru rezolvarea unei ecuații prin introducerea succesivă a unui element X printr-o serie de filtre conoluționale și neliniarități simple. Ele au o structură ierarhică, astfel încât având ca intrare x_{j-1} , îl putem calcula pe x_j folosind formula [19]:

$$x_j = \rho * W_j * x_{j-1}$$

În formula prezentată, W_j este un operator liniar, o conoluție în cazul CNN, și ρ este o neliniaritate, cunoscută și ca o funcție de activare. În cadrul proiectului am folosit doar funcția de activare ReLu(rectified linear unit), definită: $f(x) = \max(x, 0)$.

Funcția de activare ReLu este una dintre cele mai populare funcții folosite în algoritmi de clasificare a imaginilor, fiind folosită în numeroase modele de învățare automată. Este cunoscută datorită eficienței ei în cadrul clasificării unor date neliniare, precum imaginile. De asemenea, oferă și o eficiență sporită, din punctul de vedere al timpului de calcul, comparativ cu alte funcții de activare, cum ar fi funcția sigmoidă.

O rețea neuronală conoluțională este formată din trei tipuri de straturi: straturi de conoluție, straturi de reducere a dimensiunii(pooling) și straturi complet conectate. Aceste vor fi prezentate în detaliu mai jos [29]:

2.2.3.1 Stratul de intrare

La fel ca la orice rețea neuronală artificială, și aici regăsim un strat de intrare. În cazul CNN, acesta conține un vector multidimensional ce conține valorile culorii fiecărui pixel al imaginii.

2.2.3.2 Stratul de conoluție

Stratul de conoluție, a cărui prezență o regăsim și în denumirea algoritmului, este cel mai important strat pentru o rețea neuronală conoluțională. Este folosit pentru extragerea unor caracteristici ale imaginii, folosind un set de filtre, numite kernels. Aceste filtre sunt inițial generate aleator și ajustate pe parcursul procesului de antrenare. Prin aplicarea lor asupra imaginilor, se pot detecta diferite tipuri de caracteristici, cum ar fi margini sau texturi.

Un kernel are de obicei dimensiuni reduse, mult mai mici decat cea a imaginii. Este folosit prin secționarea imaginii în segmente de dimensiunea sa și aplicarea filtrului pe fiecare segment. Prin filtrarea datelor folosind un astfel de filtru, se produce o harta de activare 2D, aşa cum putem vedea în **Figura 13**.

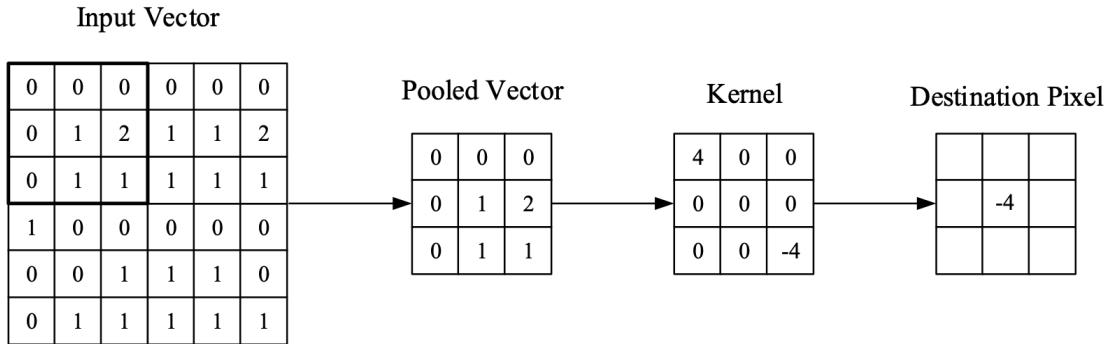


Figura 13: Aplicarea unui kernel asupra unei imagini⁷

Elementul central al unui kernel este plasat pe centrul vectorului de intrare și se realizează înmulțirea valorilor de pe fiecare poziție a filtrului cu valorile segmentului ales. În continuare, aceste produse sunt adunate pentru a obține o valoare de ieșire pentru un pixel specific.

2.2.3.3 Stratul de reducere a dimensiunii

După cum sugerează și numele, acest strat are ca scop reducerea dimensiunii reprezentării datelor, astfel încât se reduce numărul de parametri utilizati de algoritm și scade complexitatea computațională a modelului.

Stratul de reducere a dimensiunii se aplică asupra hărților de activare a unei intrări și îi reduce dimensiunile folosind o funcție specifică. În acest proiect am folosit doar funcția **MAX**, pe o dimensiune a segmentului de **2x2**. La fel ca un filtru menționat în secțiunea următoare, imaginile sunt împărțite în segmente de 2x2, cei patru pixeli fiind reduși la unul singur având valoarea egală cu maximul dintre valorile celor patru.

2.2.3.4 Stratul complet conectate

Arhitectura rețelelor neuronale convolutionale se bazează pe straturile complet conectate, cunoscute și ca straturi dense. Ele sunt responsabile pentru clasificarea finală a

⁷ [29]

datelor, fiind aşezate la sfârşitul reţelei. Astfel de straturi sunt comune cu cele regăsite într-o reţea neuronală artificială obişnuită.

Fiecare neuron din stratul anterior este conectat la fiecare neuron dintr-un strat dens. În consecinţă, fiecare valoare de ieşire din stratul anterior este luată în considerare atunci când se calculează valoarea de ieşire a neuronilor din stratul dens. Straturile dense pot capta şi combina informaţii complexe şi interacţiuni între caracteristicile extrase în straturile anterioare datorită acestei conexiuni complete.

În CNN-uri, straturile dense sunt adesea utilizate pentru a modifica caracteristicile extrase prin straturile conoluţionale şi pentru a reduce dimensiunea pentru a oferi o reprezentare finală care este potrivită pentru clasificare. Acestea sunt de obicei urmate de un strat de activare şi un strat de ieşire care generează probabilităţi pentru fiecare clasă posibilă.

Capitolul 3: Tehnologii Utilizate

3.1 Python

Python este unul dintre cele mai populare limbaje de programare folosite pentru dezvoltarea algoritmilor de învățare automată. Este cunoscut datorită multitudinii de biblioteci pe care le oferă, suportul comunității, capacitatele de integrare, flexibilitatea și opțiunile de implementare, permitând dezvoltatorilor să construiască și să implementeze modele în mod eficient.

3.1.2 TensorFlow⁸

TensorFlow a reprezentat o componentă esențială în elaborarea acestui proiect, fiind framework-ul de învățare automată folosit. Creat de Google, este un ecosistem robust de unele și module ce permite dezvoltarea, testarea și punerea în producție a modelelor de învățare automată. E opțiunea cea mai bună pentru algoritmi ce au o dificultate ridicată, de exemplu procesarea imaginilor, datorită implementărilor eficiente ale rețelelor neuronale. Straturile modelelor ce vor fi prezentate într-un capitol următor sunt următoarele:

- Input: oferă implementarea stratului de intrare al rețelei
- Rescaling: un strat de procesare pentru normalizarea valorilor imaginii
- Conv2D: pune în practică stratul de conoluție
- MaxPooling2D: concretizează stratul de reducere a dimensiunii
- Flatten: transformă imaginea reprezentată ca un vector multidimensional într-un vector unidimensional
- Dense: implementează stratul conectat complet

Procesul de antrenare ale unui CNN necesită o rată de învățare. Pentru aproximarea acesteia am folosit implementarea din TensorFlow a algoritmului Adam(Adaptive Moment Estimation). Aceasta menține rata de învățare adaptativă pentru fiecare parametru din model, permitând ajustarea dinamica în timpul procesului de învățare. În plus, acest proces are nevoie și de o funcție de pierdere. În acest proiect am folosit sparse categorical cross entropy, ce compara la fiecare pas eticheta reală a unei instanțe cu eticheta estimată de către algoritm și face schimbările necesare.

⁸ [36]

3.1.2 Albumentations⁹

Albumentations este o unealtă de computer vision folosită pentru sporirea performanței unui algoritm de învățare automată. Este un modul din Python ce permite o augmentare rapidă a datelor, oferind o gama largă de operații de transformare a imaginilor, ce sunt optimizate pentru performanță. Pentru augmentarea datelor în acest proiect am folosit următoarele operații de transformare:

- ShiftScaleRotate: aplică transformări de rotație, translație, scalare cu valori aleatoare din intervale date ca parametre
- HorizontalFlip: oglindește imaginea pe orizontală
- RandomBrightnessContrast: aplică perturbări ale luminozității și contrastului imaginii
- ColorJitter: modifică culorile imaginii prin ajustarea contrastului, luminozității, saturăției și nuanței
- GaussNoise: adaugă zgomot gaussian
- Normalize: normalizează valorile pixelilor din imagine aducându-le într-un interval specific

3.1.2 Flask¹⁰

Flask este unul dintre cele mai utilizate framework-uri pentru dezvoltarea aplicațiilor web bazate pe limbajul de programare Python. Este opțiunea ideală pentru dezvoltarea atât a aplicațiilor de dimensiuni mari, cât și a celor de dimensiuni mici datorită simplității și flexibilității pe care le oferă.

3.2 Angular¹¹

Angular este o platformă de dezvoltare web populară și puternică dezvoltată de Google. O abordare modulară și componentizată este oferită pentru crearea de aplicații web complexe și dinamice. Limbajul TypeScript este folosit de Angular pentru a oferi tipizarea statică și un sistem de gestionare al erorilor în procesul de dezvoltare. Deși oferă o gama largă de funcționalități, cum ar fi legarea bidirectională a datelor, gestionarea

⁹ [3]

¹⁰ [14]

¹¹ [4]

rutelor, serviciile și validarea formularelor, framework-ul pune accent pe separarea clară a responsabilităților.

3.2.1 PrimeNG¹²

PrimeNG este o biblioteca de componente Angular care ajuta la crearea de interfețe web rapide și reutilizabile. Este folosită pentru a adăuga componente de înaltă calitate la aplicațiile Angular, cum ar fi butoane, tabele, formulare, meniuri, diagrame și multe altele. Aceste componente sunt bine documentate și oferă o varietate de funcționalități și opțiuni de configurare și alături de un set puternic de componente predefinite, permite crearea unor interfețe personalizate. PrimeNG oferă o experiență de dezvoltare fluidă și eficientă care funcționează bine cu Angular și TypeScript.

3.2.2 Express

Express este un framework ce permite dezvoltarea aplicațiilor web în Node.js. În acest proiect am folosit Express pentru a face deploy într-un environment de producție a componentei de front end, fiind responsabil pentru administrarea fișierelor compilate ale aplicației Angular, inclusiv HTML, CSS și TypeScript.

3.3 Java¹³

Java este un limbaj de programare versatil și popular, fiind recunoscut pentru fiabilitate și portabilitatea sa. S-a stabilit în timp drept cea mai bună alegere pentru o varietate de aplicații, de la dezvoltare online și mobila, la soluții software pentru afaceri, datorită ecosistemului vast și a setului de capabilități pe care le prezintă.

Java oferă o baza puternica pentru crearea unor aplicații scalabile, sigure și cu performanță ridicată, indiferent de nivelul de experientă al dezvoltatorului. Principalele componente folosite în acest proiect sunt prezentate în secțiunile următoare.

3.2.1 Spring Boot¹⁴

Spring Boot este un framework, construit peste framework-ul Spring. Pentru a simplifica crearea unei aplicații Java, Spring Boot oferă o metodologie și o selecție de

¹² [31]

¹³ [27]

¹⁴ [34]

funcționalități preconfigurate și de dependențe. Astfel se minimizează nevoia de configurare manuală și urmărește accelerarea procesului de dezvoltare al aplicațiilor.

Adițional, o varietate de capabilități și de unelte sunt disponibile pentru crearea diferitor tipuri de aplicații, precum aplicații web, microservicii și API-uri RESTful.

3.2.2 JPA¹⁵

Prin oferirea unui model de persistență **POJO**(Plain Old Java Object), JPA(Java Persistence API) pune la dispoziție o metodă simplificată pentru maparea obiect relațională. JPA a fost creat inițial de grupul de experți software EJB 3.0 ca parte a JSR 220, deși acum are aplicații în afara componentelor software EJB. Acum, aplicațiile online, clienții aplicațiilor și chiar programele JAVA SE care rulează în afara platformei JAVA EE pot utiliza direct acest API. Datorită adaptabilității sale, JPA poate fi integrat cu ușurință în diverse medii, facilitând persistența eficientă și uniformă a datelor din mai multe aplicații bazate pe Java.

3.2.3 Hibernate¹⁶

Hibernate ORM simplifică dezvoltarea aplicațiilor ce necesită stocarea datelor pe termen lung. Fiind un framework de mapare obiect relațională, Hibernate asigura persistența datelor în baza de date utilizând JDBC. Pe lângă dezvoltarea propriului API, Hibernate reprezintă și o implementare a specificațiilor JPA.

Principala funcționalitate implementată de Hibernate este maparea claselor din Java și a tipurilor de date în tabele în baza de date și tipuri de date SQL. În plus, se ocupă de generarea apelurilor SQL și procesarea răspunsului, astfel simplificând procesul și scurtează timpul de dezvoltare al aplicației.

Maparea claselor din Java în tabele în baza de date este realizată prin intermediul unor adnotări(Java Annotations). Acestea funcționalități permit formarea unor relații între entități, precum one-to-one, one-to-many, many-to-one, many-to-many, utilizând chei strainer în baza de date. Hibernate facilitează gestionarea și sincronizarea acestor relații permitând dezvoltatorilor să modeleze și să utilizeze structuri complexe de date în aplicații.

În limbajul Java, Hibernate permite și generarea automată a operațiilor de tip join între tabele. Acest lucru face procesul de interogare a datelor din baza de date mai simplu

¹⁵ [17]

¹⁶ [16]

și garantează că tabelele interconectate primesc informațiile corecte. Această metodă face crearea aplicațiilor care necesită interogari complicate sau manipularea cu relații dintre entități mai ușoară.

De asemenea, Hibernate suportă și maparea unor tipuri de date custom, tot prin intermediul adnotarilor. Astfel se pot forma proprietăți de tipul Java Enum sau asocierea unei singure proprietăți mai multor coloane din baza de date.

3.4 PostgreSQL¹⁷

PostgreSQL este un sistem de baze de date de tipul obiect-relațional cunoscută pentru gama largă de funcționalități. Folosește limbajul SQL împreună cu un număr mare de funcționalități de ultimă generație pentru a gestiona un număr mare de date. A fost dezvoltat pentru prima dată în 1986 în cadrul proiectului **POSTGRES** la Universitatea din California, Berkeley.

PostgreSQL și-a construit o reputație solidă bazată pe asigurarea integrității datelor, gama largă de funcționalități, flexibilitate, dar și de comunitatea open source dedicată ce aduce constant îmbunătățiri. Din 2001, a urmărit principiile ACID(Atomicitate, Coerență, Izolare, Durabilitate) și este compatibilă cu toate sistemele de operare importante.

PostgreSQL pune la dispoziție o serie de capabilități pentru a-i ajuta pe programatori să creeze aplicații și pe administratori să mențină integritatea datelor. Pe lângă avantajul de a fi un program gratuit și open source, oferă și posibilitatea scalabilității. Se pot construi tipuri de date unice sau să se combine secvențe de cod din diferite limbaje de programare, totul fără a fie nevoie de o recompilare a bazei de date.

3.4 Cloudinary¹⁸

Cloudinary este o platformă cloud pentru administrarea fisierelor media. Are o gama largă de funcționalități pentru stocarea, optimizarea, transformarea folosind inteligență artificială, și distribuirea imaginilor și a fișierelor video pe diferite platforme. Cloudinary pune la dispoziție atât un API robust, cât și un SDK ce simplifică integrarea acestuia cu diferite framework-uri de dezvoltare, fiind soluția ideală pentru administrarea conținutului media într-o aplicație web sau una mobilă.

¹⁷ [30]

¹⁸ [8]

3.4 Heroku

Heroku este o platformă cloud ce permite dezvoltatorilor programelor software să facă deploy, să administreze și să își scaleze aplicațiile. Pună la dispoziție un mod eficient de a găzdui aplicațiile web, dezvoltatorul nefiind nevoie să se ocupe de administrarea infrastructurii. Folosind Heroku, se poate face deploy aplicațiilor scrise în diferite limbaje de programare, precum Java, Python și Node.js. În plus oferă și posibilitatea de a adăuga diferite resurse unei aplicații prin intermediul unui marketplace, printre care și baze de date în cloud.

Capitolul 4: Dezvoltarea Modelului de CNN

4.1 Setul de date

4.1.1 Introducere

Setul de date a fost construit din 3 seturi de date mai mici, fiecare având 2 tipuri de scanări cu raze X, un set de scanări cu plămâni sănătoși și un set ce prezintă una dintre bolile prezentate în acest proiect:

- [18]: conține imagini cu plămâni afectați de pneumonie. Scanările au fost adunate pe parcursul verificărilor de rutină ale unor pacienți pediatrici cu vîrstă între 0 și 5 ani. Fiecare imagine a fost clasificată manual de către 2 specialiști, care au și eliminat toate scanările cu o calitate scăzută¹⁹.
- [32]: prezintă scanări ale unor pacienți diagnosticați cu tuberculoză. A fost format de o echipă de cercetători din cadrul Universității Qatar, din Doha, în parteneriat cu Universitatea din Dhaka, Bangladesh, colaboratori din Malaysia și doctori de la Hamad Medical Corporation și din Bangladesh²⁰.
- [9]: cuprinde un set de imagini ale unor pacienți având Covid-19. Au fost adunate de autori în scop de cercetare, atât pentru dezvoltarea propriilor lucrări, cât și pentru folosirea de alții dezvoltatori.

4.1.2 Preprocesarea datelor

Fiecare set de date era deja împărțit fie în 2 seturi: de antrenare și de testare, fie în 3 seturi: de antrenare, de testare și de validare. Pentru a avea o uniformitate, am combinat pentru fiecare caz toate seturile de date, obținând 4 foldere, cate unul pentru fiecare clasa. Am împărțit noul set de date obținut în 3 secțiuni: 80% pentru antrenare, 10% pentru validare și 10% pentru testare.

Pentru a îmbunătăți performanța modelelor, am ales folosirea unor tehnici de augmentare a datelor doar pe secțiunea selectată pentru antrenare. Acest proces ajuta la:

- Gestionarea dezechilibrului dintre clase: clasa de clasificare Normal s-a regasit în toate cele 3 seturi de date originale, fiind principalul motiv pentru care noul set de date nu era echilibrat

¹⁹ [7]

²⁰ [22]

- Evitarea ovefit-ului și generalizarea modelului: prin aplicarea diferitor transformări asupra datelor se poate obține un model mai robust, capabil să generalizeze informațiile obținute din setul de date pe exemple noi.
- Creșterea numărului de date de antrenament

În procesul de augmentare, am cercetat diferite surse ce prezintă transformări potrivite pentru domeniul medical, precum [11], [12], și am ales câteva potrivite pentru o scanare a plamanilor, fiecare având o probabilitate de aplicare de 50%:

- ShiftScaleRotate: am folosit limita de 0.05 pentru scalare și translație și 15 grade pentru rotație. Prin aceasta transformare luăm în considerare cazurile în care pacientul nu este într-o poziție complet verticală pe centru imaginii și nu ținem cont de dimensiunea pacientului.



Figura 14:
Plamani afectati
de tuberculoza -
Original



Figura 15:
Transformarea
ShiftScaleRotate



Figura 16:
Transformarea
HorizontalFlip

- HorizontalFlip: bolile prezентate pot afecta ambii plamani în același mod
- RandomBrightnessContrast: scanările pot avea o variație în luminozitate în funcție modul în care au fost digitalizate: dacă au fost produse direct de aparatul de radiografie în format digital, sau dacă au fost ulterior fotografiate sau scanate
- ColorJitter: în funcție de aparatul de radiografie, scanările pot avea diverse nuanțe de culori
- GaussNoise: pentru a lua în calcul diferențele zgomote ce pot apărea într-o imagine, am adăugat zgomot gaussian cu o medie de 10



Figura 17:
Transformarea
RandomBrightness
Contrast

Figura 18:
Transformarea
ColorJitter

Figura 19:
Transformarea
GaussNoise

- Normalize: asigură că intensitatea pixelilor din imagine se încadrează într-un nivel comun, cu medii apropiate de zero și o dispersie redusă. Am folosit media celor 3 canale de culoare egală cu 0.485, 0.456, 0.406 și deviația standard 0.229, 0.224, 0.225

4.2 Descrierea modelelor

4.2.1 Modelul 1

Tipul Stratului	Forma de ieșire	Numărul de parametrii	Funcție de activare
Input	(256, 256, 3)	0	-
Rescaling	(256, 256, 3)	0	-
Conv2D	(254, 254, 32)	896	ReLU
MaxPooling2D	(84, 84, 32)	0	-
Dense	(84, 84, 64)	2112	-
Conv2D	(82, 82, 64)	36928	ReLU
MaxPooling2D	(27, 27, 64)	0	-
Flatten	(46656)	0	-
Dense	(4)	186628	SoftMax

Numărul total de parametrii este 226564, toți fiind antrenabili.

Tabelul 1: Structura Modelului 1

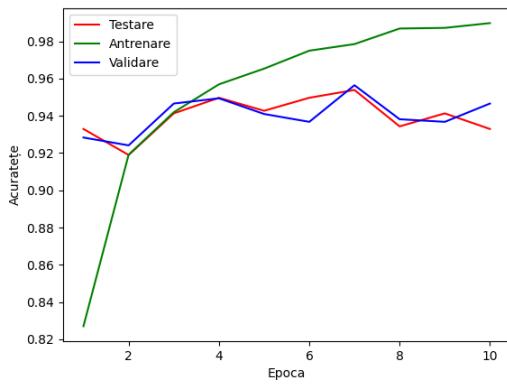


Figura 20: Acuratețe

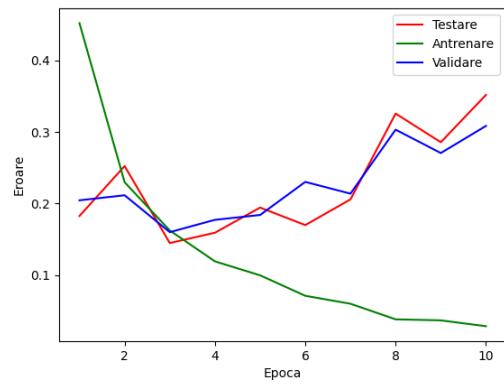


Figura 21: Eroare

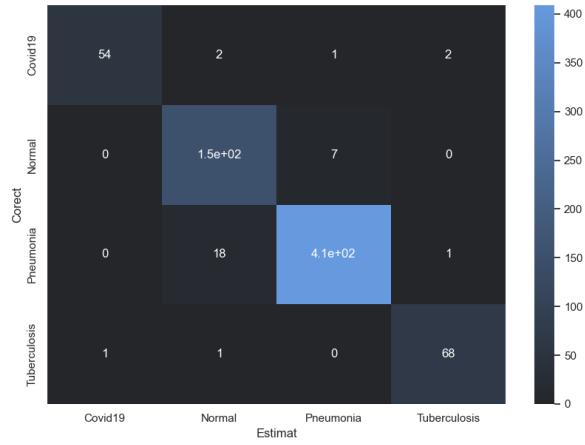


Figura 22: Matricea de confuzie

Modelul prezentat are ca intrare o imagine în format RGB de 256x256 de pixeli și un total de 226564 de parametrii. Din **Figurile 20 și 21** putem vedea că de la epoca 7 avem acuratețea maximă, de 95.39%. De la epoca 8, modelul învăță trăsăturile setului de antrenare și își pierde acuratețea la validare și testare.

Matricea de confuzie din **Figura 22** a fost construită folosind varianta de la epoca 7 a modelului și setul de date de test.

4.2.1 Modelul 2

Tipul Stratului	Forma de ieșire	Numărul de parametrii	Funcție de activare
Input	(256, 256, 3)	0	-
Rescaling	(256, 256, 3)	0	-
Conv2D	(254, 254, 64)	1792	ReLU
MaxPooling2D	(84, 84, 64)	0	-
Dense	(84, 84, 64)	4160	-
Conv2D	(82, 82, 128)	73856	ReLU
MaxPooling2D	(27, 27, 128)	0	-
Conv2D	(25, 25, 256)	295168	ReLU
MaxPooling2D	(8, 8, 256)	0	-
Flatten	(16384)	0	-
Dense	(4)	65540	SoftMax
Numărul total de parametrii este 440516, toți fiind antrenabili.			

Tabelul 2: Structura **Modelului 2**

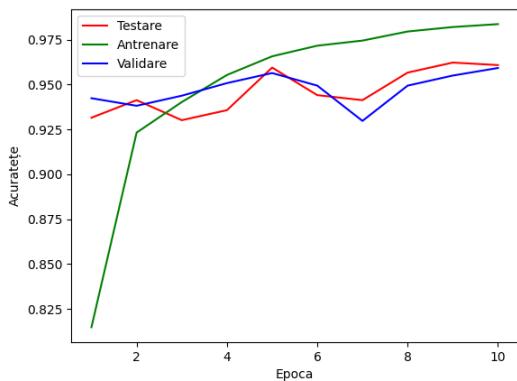


Figura 23: Acurătete

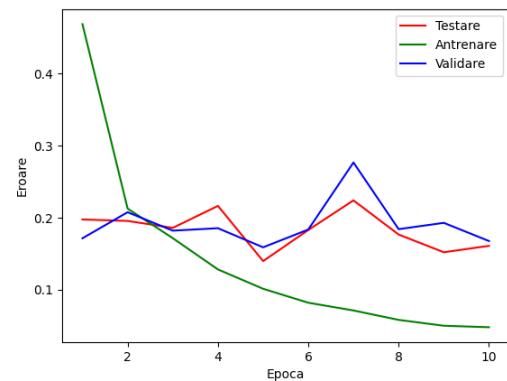


Figura 24: Eroare

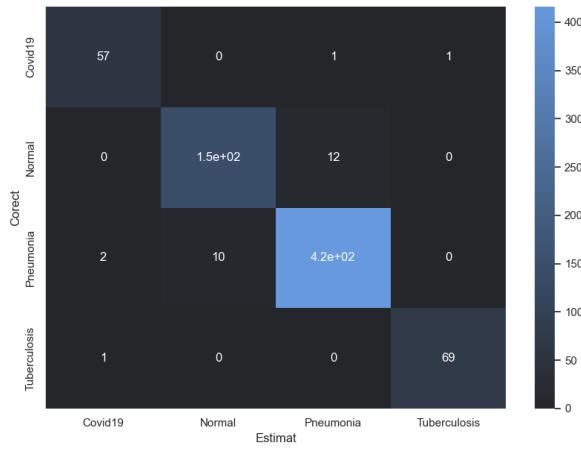


Figura 25: Matrice de confuzie

Modelul 2 extinde modelul prezentat în secțiunea anterioară, diferențindu-se cu o parte din parametrii straturilor și în plus are o perche de straturi(unul de convoluție și unul de reducere al dimensiunii). Astfel cu dezavantajul unui timp de procesare mai ridicat din cauza unui număr mai mare de parametrii, obținem o acuratețe puțin mai mare, de 96.23%. Din **Figurile 23 și 24** putem vedea că epoca 9 este cea mai bună, matricea de confuzie pentru ea fiind reprezentată în **Figura 25**.

4.2.1 Modelul 3

Tipul Stratului	Forma de ieșire	Nr de parametrii	Funcția de activare
Input	(256, 256, 3)	0	-
Rescaling	(256, 256, 3)	0	-
Conv2D	(254, 254, 64)	1792	ReLU
MaxPooling2D	(84, 84, 64)	0	-
Conv2D	(82, 82, 128)	73856	ReLU
MaxPooling2D	(27, 27, 128)	0	-
Conv2D	(25, 25, 256)	295168	ReLU
MaxPooling2D	(8, 8, 256)	0	-
Dense	(8, 8, 64)	16448	-
Flatten	(3094)	0	-
Dense	(4)	16388	SoftMax
Numărul total de parametrii este 403652, toți fiind antrenabili.			

Tabelul 3: Structura **Modelului 3**

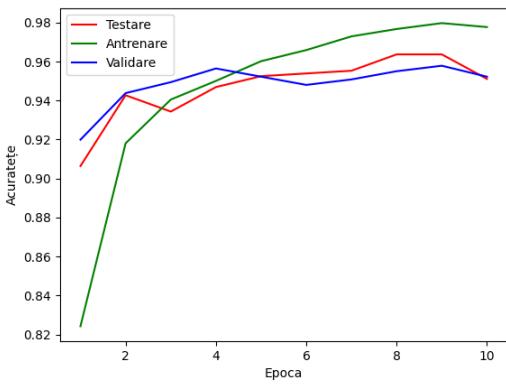


Figura 26: Acuratețe

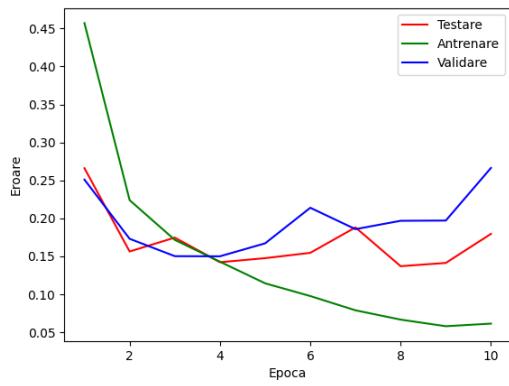


Figura 27: Eroare

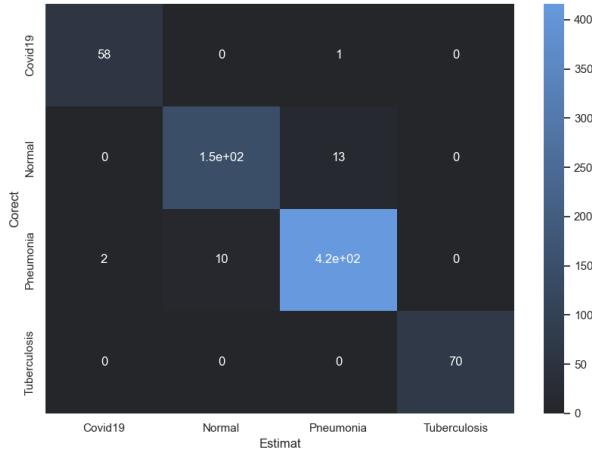


Figura 28: Matrice de confuzie

Similar cu modelele prezentate anterior, **Modelul 3**, primește la intrare o imagine RGB de 256x256 de pixeli. Din punctul de vedere al numărului de parametrii, acest model se situează între **Modelul 1** și **Modelul 2**, cu un total de aproape 404000 de parametrii, dar are o acuratețe mai bună decât ambele, de 96.37%. Din matricea de confuzie putem vedea erorile apărute în procesul de estimare al setului de date folosind modelul în varianta sa de la epoca 9.

4.2.1 Modelul 4

Tipul Stratului	Forma de ieșire	Numărul de parametrii	Funcție de activare
Input	(32, 32, 1)	0	-
Rescaling	(32, 32, 1)	0	-
Conv2D	(30, 30, 32)	320	ReLU
MaxPooling2D	(15, 15, 32)	0	-
Conv2D	(13, 13, 64)	18496	ReLU
MaxPooling2D	(6, 6, 64)	0	-
Flatten	(2304)	0	-
Dense	(1024)	2360320	-
Dense	(200)	205000	-
Dense	(4)	804	SoftMax
Numărul total de parametrii este 2594940, toți fiind antrenabili.			

Tabelul 4: Structura Modelului 4

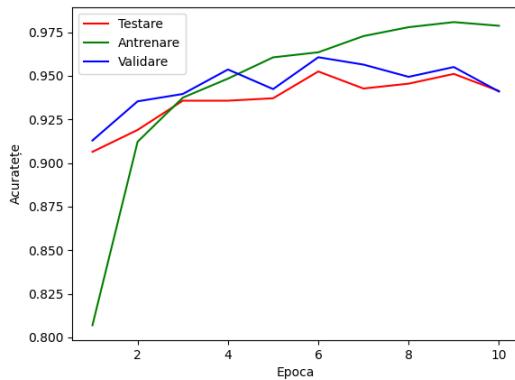


Figura 29: Acuratețe

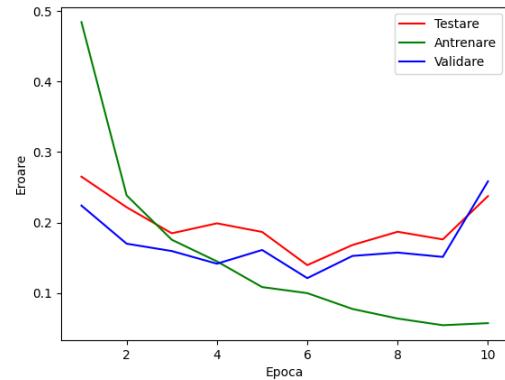


Figura 30: Eroare

Comparativ cu **Modelele 1-3**, **Modelul 4** pornește de la o imagine de 32x32 de pixeli în format alb-negru. Chiar dacă numărul de parametri, aproape 2.6 milioane este foarte mare comparativ cu celelalte modele, timpul de procesare al unei imagini este de 5 ori mai mic decât **Modelul 3** și reușește să obțină în epoca 6 o acuratețe de 95.25%.

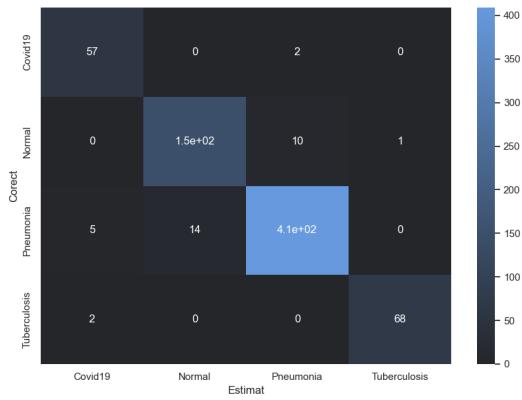


Figura 31: Matrice de confuzie

4.3 Concluzii

După cum am precizat în fiecare secțiune de mai sus, fiecare model are avantajele și dezavantajele lui. Din matricile de confuzie putem vedea că toate modelele au erori majoritare pe aceleași categorii, pe perechile de forma Corect-Estimat: Pneumonie-Normal și Normal-Pneumonie.

Dintre primele 3 modele, **Modelul 3** prezintă raportul cel mai bun dintre acuratețe și timpul de procesare. **Modelul 4** este cel mai rapid din punctul de vedere al timpului de procesare datorită dimensiunii mici a datelor, dar nu este o diferență destul de mare cât să compenseze acuratețea scăzută față de **Modelul 3**.

Alegerea numărului de epoci este esențială în dezvoltarea unui CNN, deoarece acesta poate duce la o performanță mai bună a modelului. Alegerea epocii ideale pentru a folosi modelul în practică după antrenare, este la fel de importantă. Din analiza graficelor de performanță, putem concluziona că epoca 9 a **Modelului 3** este cea mai bună, fiind punctul de convergență al modelului.

Capitolul 5: Arhitectură și implementare

5.1 Arhitectura sistemului

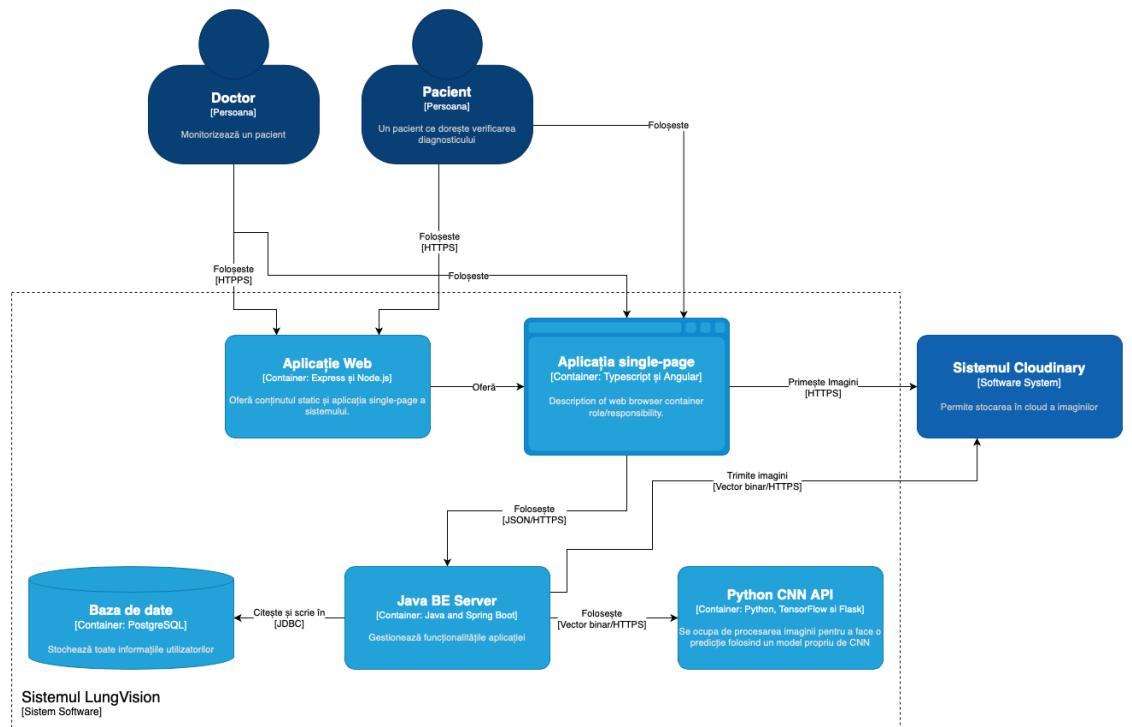


Figura 32: Nivelul 2 al diagramei C4

Diagrama containerelor

Sistemul LungVision este structurat în 3 module:

- Aplicația Web: oferă utilizatorului aplicația single page dezvoltată cu ajutorul framework-ului Angular, care servește ca interfață a aplicației.
- Serverul Java: serverul principal al aplicației. Se ocupă de înregistrarea și autentificarea utilizatorului, filtrarea cererilor în funcție de nivelul de autorizare și administrează toate acțiunile ce se pot face în aplicație, precum estimarea unui diagnostic folosind componenta responsabilă de învățare automată, efectuarea tuturor operațiilor CRUD pentru administrarea acestor predicții și a altor detalii ale pacientului. Pentru salvarea imaginilor folosește sistemul extern, Cloudinary.
- Python Server: componenta ce se ocupă de procesarea unei imagini și trecerea ei prin algoritmul de învățare automată

5.2 Arhitectura Serverului Principal

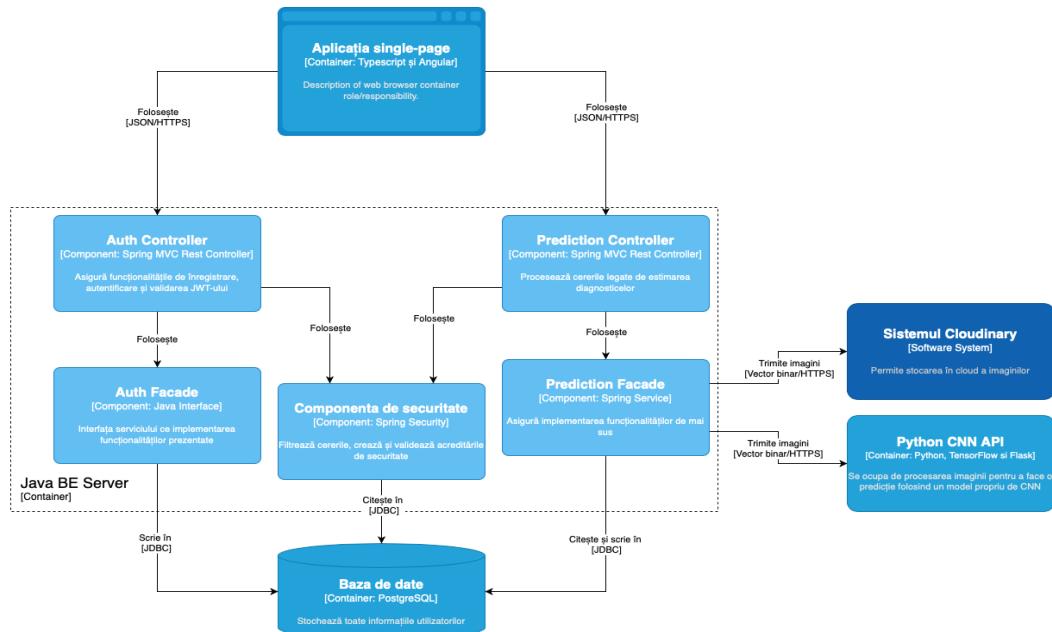


Figura 33: Nivelul 3 al diagramei C4
Diagrama Containerului Java BE Server

Serverul principal al aplicației este implementat în Java, utilizând framework-ul Spring Boot. și urmează stilul arhitectural RESTful. Acest server oferă trei funcționalități:

- Autentificarea și autorizarea utilizatorului
- Estimarea unui diagnostic pe baza unei scanări cu raze X a plămânilor, prin apelul către API-ul responsabil pentru operațiile de învățare automată
- Salvarea și administrarea estimărilor alături de alte date personale ale pentru o vizualizare ulterioară

Aplicația permite utilizatorului salvarea și vizualizarea unor date personale, motiv pentru care am ales securizarea contului folosind JWT(JSON Web Token). Serverul aplică 2 filtre de securitate:

- Primul filtru gestionează doar cererile efectuate pe calea de autentificare. După autentificare, serverul generează un token JWT și îl trimite ca răspuns. JWT-ul are o valabilitate de 5 ore și este adăugat de interceptor la orice cerere HTTPS viitoare
- Al doilea filtru administrează celelalte cereri, exceptie celor pentru înregistrare. Când utilizatorul face o cerere, filtrul verifică prezența header-ului de autorizare și validează JWT-ul. Dacă token-ul este validat, cererea este trimisă mai departe pentru generarea unui răspuns

5.3 Arhitectura Bazei de Date

În **Figura 34** este prezentată structura bazei de date prin intermediul unei diagrame de clase. Aceasta prezintă componența fiecărei entități, dar și relațiile dintre ele. Maparea din clasele Java în entități din baza de date s-a făcut cu ajutorul JPA.

Orice utilizator al aplicației are posibilitatea de a salva mai multe predicții în baza de date pentru o vizualizare ulterioară. Orice predicție salvează diagnosticul obținut pentru o scanare cu raze X, împreună cu exact o imagine, păstrată în entitatea MedicalScan, și cu alte detalii personale ale pacientului.

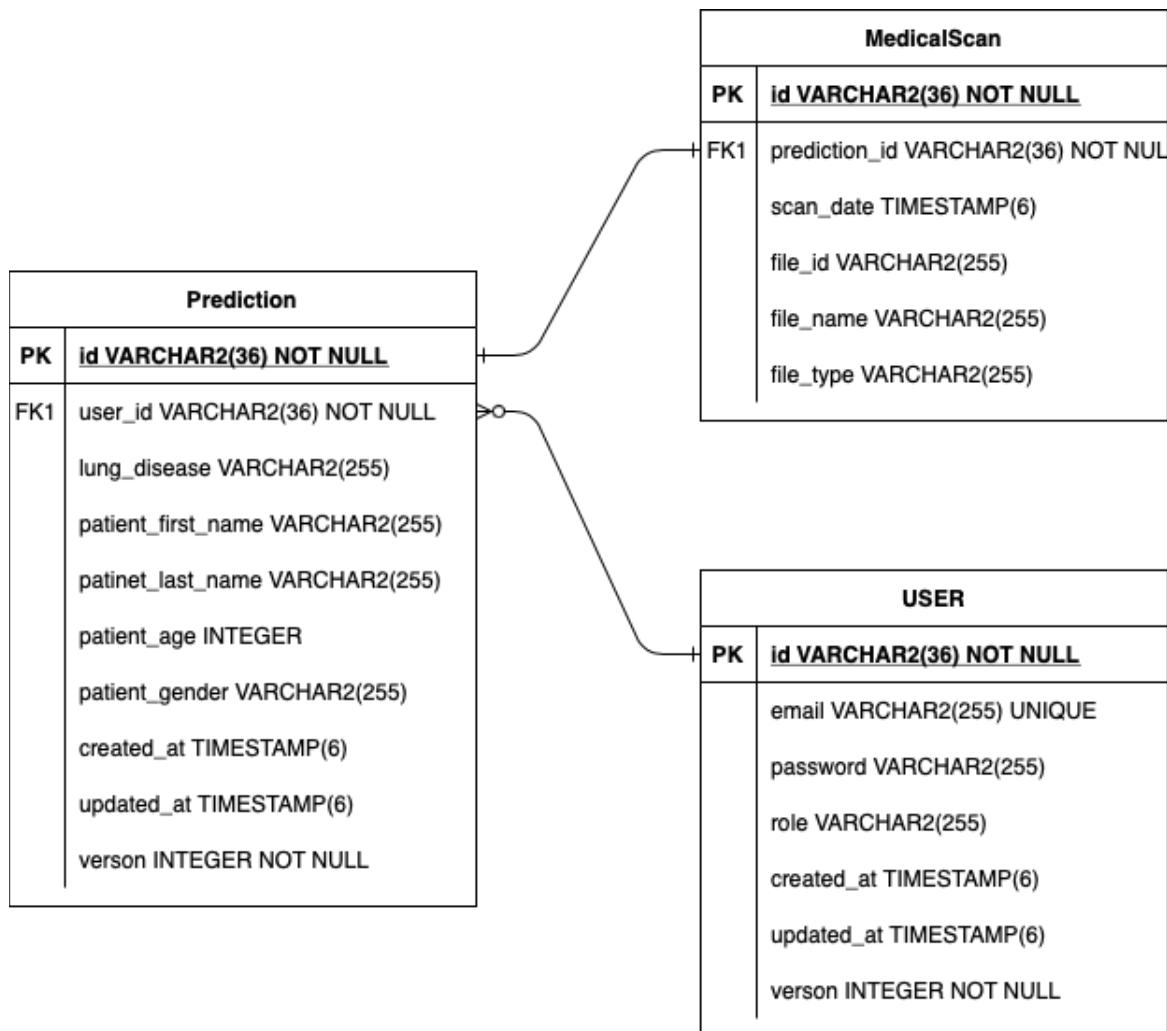


Figura 34: Diagrama de clase

5.4 Detalii de Implementare

Aplicația Web LungVision este structurată în trei pagini, fiecare oferind o funcționalitate specifică:

5.4.1 Pagina Principală



Figura 35: Pagina Principală

Aceasta este pagina de pornire a aplicației LungVision. Ea oferă o prezentare generală a aplicației și integrează componentele de autentificare și înregistrare, ce apar într-un meniu de tipul dialog prin apăsarea butonului din header.

Header-ul este o componentă comună, ce se găsește pe toate paginile aplicație, având diferite forme. Acesta folosește serviciul de securitate pentru a verifica validitatea token-ului prin apelarea unui endpoint din serverul principal. Un gardian din Angular interceptează răspunsul redirectionand utilizatorul în cazul lipsei de drepturi.

Pentru generarea imaginilor prezente în interfață, precum emblema aplicației și imaginea cu plămânnii, vizibile în **Figura 35**, am folosit platforma Midjourney. Aceasta pune la dispoziție un model de inteligență artificială care poate genera imagini pornind de la o descriere în limbaj natural. După generare, am editat rezultatele prin îndepărțarea fundalului și ajustarea culorilor, astfel încât să se potrivească temei aplicației.

Meniul de autentificare are două campuri: e-mail și parolă. Îl permite utilizatorului autentificarea pentru a iniția sau pentru a salva o estimare nouă, dar și pentru a accesa datele salvate precedent. Ambele câmpuri au aplicate validări specifice tipului lor:

- email: câmp obligatoriu și trebuie să respecte formatul regex cunoscut
- parolă: camp obligatoriu

Figura 36: Dialogul de autentificare

Figura 37: Dialogul de înregistrare

5.4.2 Pagina de Analiză și Predicție

Pe această pagină, utilizatorul poate încărca o imagine de pe dispozitivul personal, ce va apărea în panoul din stanga. Ulterior, aceasta poate fi analizată prin maximizarea ei apăsând pe componenta respectivă sau se poate estima un diagnostic prin apăsarea butonului "Predict". Prin apasarea acestuia este trimisă o cerere către serverul principal, care la rândul său apelează componenta responsabilă de algoritmul de predicție. Răspunsul este procesat

Dialogul din **Figura 37** oferă funcționalitatea de înregistrare. Similar cu meniul anterior, fiecare câmp este validat în funcție de tipul său și, în plus, câmpul de parolă prezintă un indicator de putere. Pentru a asigura validitatea parolei, aceasta trebuie să împlinească următoarele condiții:

- o literă majusculă
- o literă majusculă
- un caracter numeric
- un caracter special
- 8 caractere

de server și apoi trimis clientului pentru afișarea sub forma unei diagrame ce arată procentul pentru fiecare clasă de clasificare, iar dedesubt diagnosticul estimat.

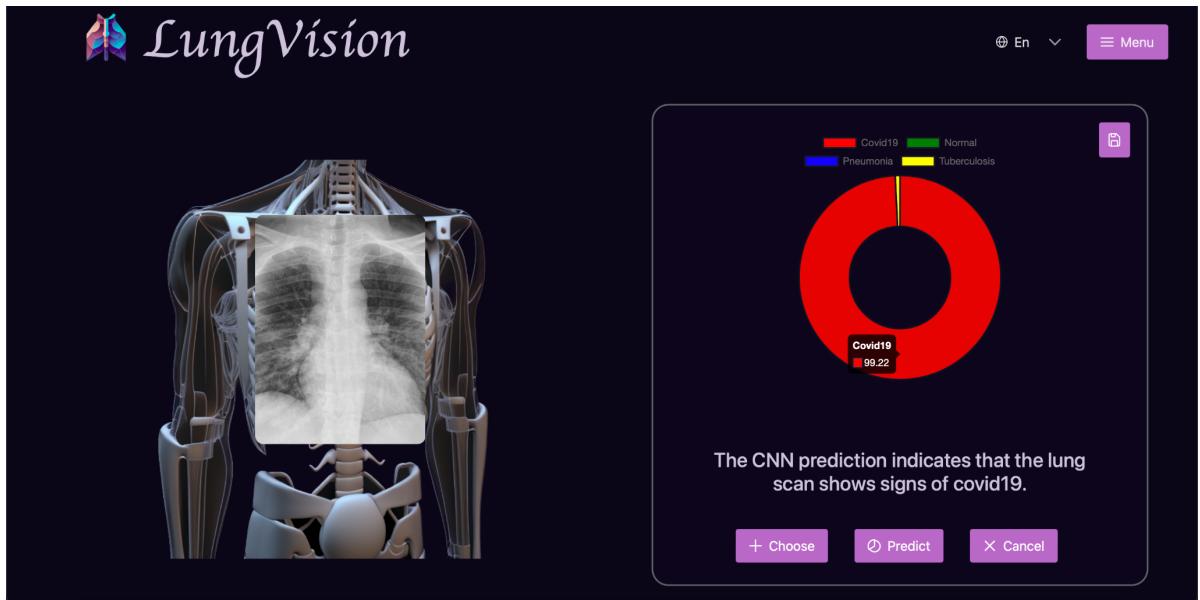


Figura 38: Pagina de Analiză și Predicție

Odată ce utilizatorul apasă butonul de salvare al predicției, vizibil în **Figura 38**, se afișează o fereastră de dialog care solicită introducerea unor date suplimentare despre pacient:

- Numele și Prenumele
- Vârstă
- Genul
- Data la care producerii scanării cu raze X

Diagnosticul estimat nu este trimis de clienți în cererea de salvare, serverul apelând din nou serviciul de clasificare

Figura 39: Dialogul pentru introducerea datelor personale ale pacientului

5.4.3 Istoric și Vizualizarea Datelor

Scan	First Name ↑	First Name ↑	Age ↑↓	Gender ↑↓	Date ↑↓	Prediction ↑↓
	Cosmin	Pasat	22	Male	10.J 23	COVID19 ✓
	Jane	Doe	22	Female	10 Jun 23	NORMAL
	John	Doe	33	Male	10 Jun 23	TUBERCULOSIS

Figura 40: Istoric și Vizualizarea Datelor

Pagina de Istoric și Vizualizare a Datelor este un spațiu în aplicație destinate administrării tuturor scanarilor salvate anterior de către utilizator. Interfața este concepută într-un mod intuitiv și accesibil, oferind funcționalități de sortare și căutare pentru a facilita găsirea rapidă a predicțiilor dorite. În plus, oferă și suport pentru editarea detaliilor introduse de utilizator, în cazul identificării unei erori de scriere.

Pentru salvarea și expunerea imaginilor am încercat inițial stocarea directă în baza de date, dar aceasta abordare încetinea întregul proces de mapare a entităților din baza de date în instanțe ale claselor din Java. Prin urmare am ales stocarea lor folosind platforma cloud Cloudinary. După încărcarea imaginii pe platformă, primim un id unic, care este stocat în baza de date pentru o utilizare ulterioară.

Această pagină este esențială pentru urmărirea progresului diagnosticelor obținute prin intermediul componentei de estimare a paginii. Ea oferă un mediu centralizat și organizat în care utilizatorii pot accesa și analiza în mod eficient informațiile relevante, contribuind astfel la îmbunătățirea procesului de diagnosticare și supraveghere a pacienților.

5.4.5 Gestionarea Erorii 404



Figura 41: Istorici și Vizualizarea Datelor

Când un utilizator încearcă să acceseze o adresă URL care nu există în cadrul aplicației LungVision, este redirecționat pe pagina din **Figura 41**. Ilustrarea prezenta în cadrul paginii este generată pe platforma Storyset și integrată în aplicație prin intermediul unui SVG. Storyset oferă o gamă largă de ilustrații ce pot fi personalizate în funcție de nevoile utilizatorului. Astfel am putut folosi culori specifice aplicației LungVision și am adaugat animații individuale pe componente alese.

5.4.6 Internaționalizare

LungVision are ca scop îmbunătățirea procesului de diagnosticare a unui pacient, motiv pentru care am considerat internaționalizarea ca fiind o necesitate. Aceasta permite adaptarea și traducerea conținutului aplicației în diferite limbi și culturi, astfel încât utilizatorii din diferite regiuni și țări să îl poată utiliza și înțelege. Datorită internaționalizării, aplicația devine accesibilă la nivel global, permitând utilizatorilor să folosească limba lor maternă, dacă este suportată, ceea ce le oferă o experiență mai confortabilă și facilitează accesarea aplicației de un număr mai mare de utilizatori.

Momentan proiectul suportă doar limba engleză, dar adaugarea unei noi limbi este automatizată. Pe partea de interfață este nevoie doar de adăugarea codului ei(ex: "en") în vectorul unidimensional cu limbi suportate, și crearea unui JSON cu o structură identică cu fișierul deja existent, care să conțină traducerile necesare. Un interceptor adaugă headerul specific cu valoarea limbii selectate și serverul răspunde corespunzător, dacă are limba selectată, sau alege limba engleză în caz contrar.

5.4.7 Deploy

Pentru a face deploy aplicației am ales folosirea platformei Heroku. Aceasta fiind structurată în trei containere distincte, ne-a permis împărțirea ei pe 3 ramuri pe github și formarea pentru fiecare a propriului pipeline. Comunicarea între ele se realizează folosind protocolul HTTPS, asigurând astfel securitatea transferului datelor.

Interfața Web a aplicației este responsabilă cu interacțiunea cu clientul și comunică cu serverul principal utilizând formatul JSON pentru transferul datelor.

Serverul principal trimite cereri atât către aplicația din Python responsabilă pentru estimarea diagnosticelor, cât și către platforma Cloudinary folosind un vector binar, care reprezintă imaginea. Răspunsul venit din partea celor două este în formatul JSON.

Utilizând formatul JSON, serverul și interfața web transmit și primesc date într-un mod organizat și simplu de procesat, ceea ce facilitează interoperabilitatea.

Capitolul 6: Scenarii de utilizare

6.1 Ajutor în Procesul de Diagnosticare

Doctorii pot integra aplicație prezentată în procesul lor de diagnosticare a unui pacient. Prin folosirea estimărilor produse de CNN ca resursă adițională, când fac decizii cu privire la planul de tratament sau determină nevoia unor teste adiționale.

6.2 Confirmarea Diagnosticului

Estimarea oferită de LungVision poate fi utilizată de către un doctor specialist pentru a valida sau a-și verifica diagnosticul original. Aceasta poate mări precizia diagnosticului și să ofere informații noi.

6.3 A Doua Opinie

Cum aplicația permite oricărui utilizator să încarce imagini cu scanări cu raze X a plămânilor, poate veni în ajutorul pacienților ce au fost diagnosticați de către un doctor, prin oferirea unei perspective adiționale. Primirea unui nou diagnostic poate asista pacientul în luarea deciziilor medicale, cum ar fi începerea tratamentului propus de medicul inițial sau apelarea la un doctor nou, în cazul în care diagnosticile diferă.

6.4 Urmărirea Progresului

LungVision poate reprezenta o platformă pentru urmărirea dezvoltării anumitor probleme în timp, datorită funcționalității de salvare a scanarilor, diagnosticelor și a datelor pacientului. Asta poate ajuta la urmărirea eficienței tratamentului, progresul de vindecare sau avansarea afecțiunii.

6.5 Instrument pentru Învățare

Este posibila folosirea aplicației ca resursă de învățare și antrenare. Prin oferirea studenților și rezidenților accesul la imaginile salvate în aplicație, instructorii le pot folosi pentru a prezenta scanări reale, după terminarea procesului de vindecare și validarea diagnosticului.

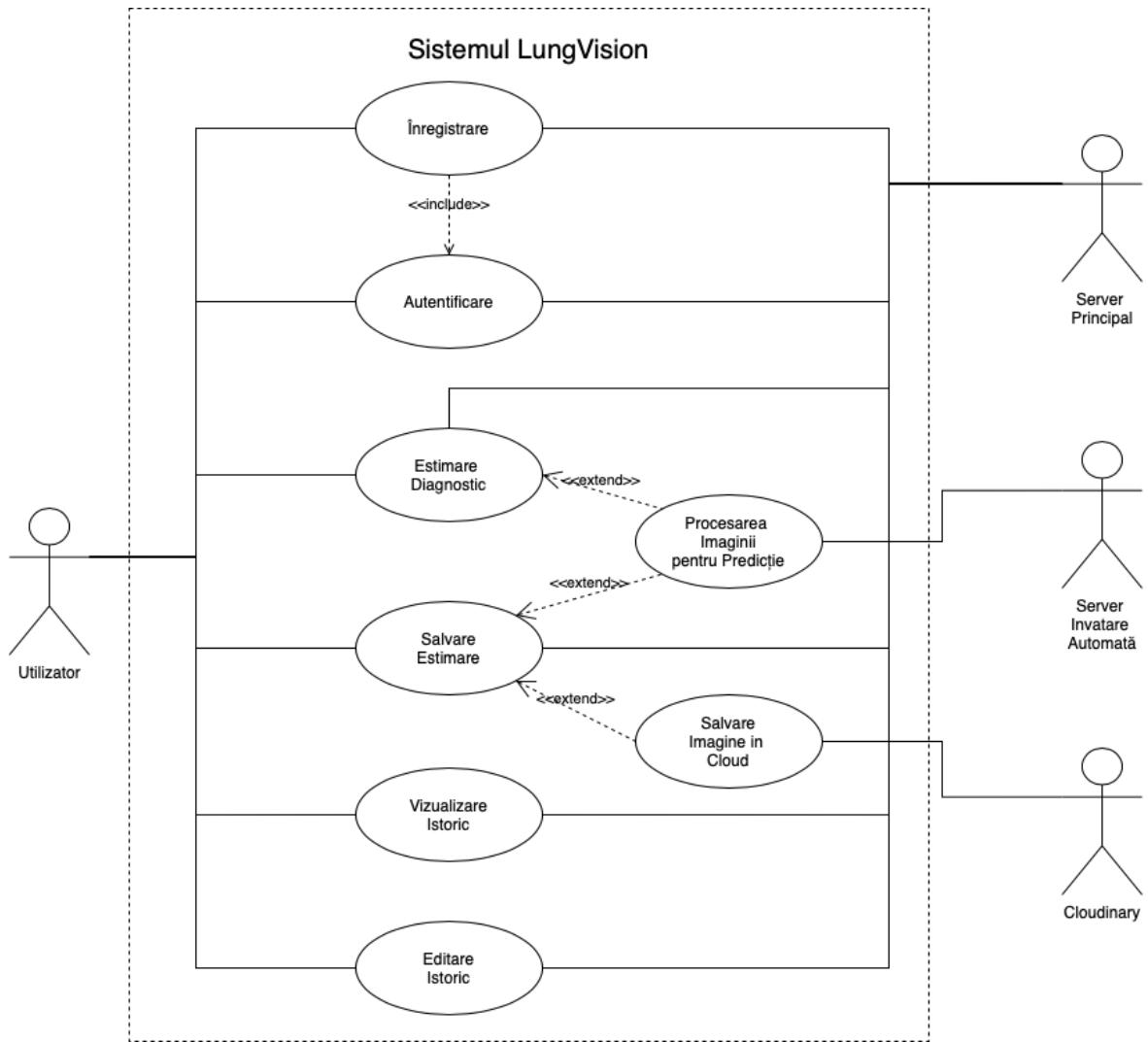


Figura 42: Diagrama Use Case

Contribuții

Contribuțiile aduse proiectului LungVision sunt:

- Am construit un set de date ce conține imagini aparținând a patru clase de clasificare: Normal, Pneumonie, Tuberculoză, Covid-19
- Am cercetat și am construit un set de transformări potrivite setului de date, pentru augmentarea datelor de antrenament
- Am cercetat și am construit câteva modele de rețele neuronale convoluționale, ce au atins o acuratețe de pana la 96% pe setul de date menționat
- Am făcut un RESTful API în Python pentru a permite utilizarea modelului de CNN
- Am creat un design intuitiv și usor de utilizate al unei aplicații Web pentru estimarea diagnosticelor folosind modelul de CNN, pe care l-am implementat cu succes, folosind framework-ul Angular
- Am creat un RESTful API în Java Spring Boot, ce poate răspunde la cereri de autentificare, înregistrare, estimarea unui diagnostic și efectuarea operațiilor CRUD pentru predicții și alte date
- Am format un sistem de securitate cu autorizare pe baza de JWT, astfel încât datele utilizatorului să rămână confidențiale
- Am luat în considerare o posibilă utilizare a aplicației de utilizatori din diferite parti ale lumii și am pregătit aplicația pentru internationalizare
- Am folosit tehnologii de actualitate, precum framework-urile Angular, Spring Boot, PostgreSQL

Idei viitoare pentru extinderea aplicației:

- Adaugarea rolului de doctor, ce ar necesita documente adiționale care să ateste acreditarea. Cu ajutorul său poate fi adăugată o componentă a aplicației unde doctorii pot comunica între ei pentru întrajutorare
- tot prin folosirea unui rol de doctor poate fi adăugată o metodă de comunicare între doctori voluntari și pacienți ce au nevoie de opinia unui specialist

Concluzii

În concluzie, studiul prezentat în **Capitolul 3** a arătat că folosirea unui model de inteligență artificială poate fi eficientă în detectarea bolilor pulmonare: Pneumonie, Tuberculoză și Covid-19. Modelele au atins un nivel mare de acuratețe, de pana la 96%, și ne arată o idee de cum ar putea deveni sistemul medical în viitor. Este de menționat, că antrenarea a fost făcută pe un număr limitat de date, modelele având nevoie de un proces de validare îndelungat, realizat de doctori specialisti prin verificarea diagnosticului oferit de aplicație după formarea propriei opinii.

De asemenea, am prezentat, prin intermediul diagramei C4, structura sistemului și a serverului principal, evidențiind relația dintre componente și protocoalele de comunicare.

Cu ajutorul scenariilor de utilizare, am arătat valoarea pe care o poate aduce aplicația în practică, fiind accesibilă prin intermediul unei interfețe sugestive și ușor de utilizat de oameni din diferite parti ale lumii.

Bibliografie

- [1] *Advancing Healthcare Research & AI in Medicine*. (n.d.). Google Health. Retrieved June 13, 2023, from <https://health.google/health-research/>
- [2] *Aidoc*. (n.d.). Aidoc Always On Healthcare AI. Retrieved June 13, 2023, from <https://www.aidoc.com/>
- [3] *Albumentations*. (n.d.). Albumentations: fast and flexible image augmentations. Retrieved June 3, 2023, from <https://albumentations.ai>
- [4] *Angular*. (n.d.). Angular.io. Retrieved June 4, 2023, from <https://angular.io>
- [5] *Augmented Alzheimer MRI Dataset*. (n.d.). Kaggle. Retrieved June 7, 2023, from <https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset>
- [6] Azhar, S., Bangyal, W. H., Ahmad, J., & Oloko, M. (2022, August 12). *Pneumonia Detection in Chest X-Ray Images Using Enhanced Restricted Boltzmann Machine*. Hindawi. Retrieved June 12, 2023, from <https://www.hindawi.com/journals/he/2022/1678000/>
- [7] *Chest X-Ray Images (Pneumonia)*. (n.d.). Kaggle. Retrieved June 7, 2023, from <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>
- [8] *Cloudinary*. (n.d.). Image and Video Upload, Storage, Optimization and CDN. Retrieved June 6, 2023, from <https://cloudinary.com/>
- [9] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q., & Ghassemi, M. (2020). COVID-19 Image Data Collection: Prospective Predictions Are the Future. *arXiv 2006.11988*. <https://github.com/ieee8023/covid-chestxray-dataset>
- [10] Copeland, B. (n.d.). *Artificial intelligence (AI) | Definition, Examples, Types, Applications, Companies, & Facts*. Encyclopedia Britannica. <https://www.britannica.com/technology/artificial-intelligence>
- [11] Desternes, J. (2021, July 13). *Deep learning in medical imaging: data augmentation*. IMAIOS. Retrieved June 7, 2023, from

<https://www.imaios.com/en/resources/blog/ai-for-medical-imaging-data-augmentation>

- [12] *The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective.* (2021, January 29). Frontiers. Retrieved June 7, 2023, from <https://www.frontiersin.org/articles/10.3389/fmed.2021.629134/full>
- [13] Facultatea de Informatică Iași. (n.d.). *Inteligentă Artificială. IA@FII.* <https://sites.google.com/view/iafii/home>
- [14] *Flask Framework.* (n.d.). Welcome to Flask — Flask Documentation (2.3.x). Retrieved June 3, 2023, from <https://flask.palletsprojects.com/en/2.3.x/>
- [15] Hashmi, M. F., Katiyar, S., Keskar, A. G., Bokde, N. D., & Geem, Z. W. (2020, June 19). *Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning.* NCBI. Retrieved June 12, 2023, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7345724/>
- [16] *Hibernate.* (n.d.). Hibernate. Everything data. Retrieved June 4, 2023, from <https://hibernate.org>
- [17] *Java Persistence API.* (n.d.). Oracle. Retrieved June 4, 2023, from <https://www.oracle.com/java/technologies/persistence-jsp.html>
- [18] Kermany, D., Zhang, K., & Goldbaum, M. (2018, January 5). *Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification.* Mendeley Data. Retrieved June 7, 2023, from <https://data.mendeley.com/datasets/rscbjbr9sj/2>
- [19] Koushik, J. (2016, 5 30). Understanding Convolutional Neural Networks. <https://arxiv.org/pdf/1605.09081.pdf>
- [20] Kundu, R., Das, R., Geem, Z. W., Han, G.-T., & Sarkar, R. (2021, 09 07). Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLoS One*, 16(9). 10.1371/journal.pone.0256630

- [21] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1990). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*.
- [22] Lloyd, G. (n.d.). *Chest X-ray - Pulmonary disease - Tuberculosis*. Radiology Masterclass -. Retrieved June 2, 2023, from https://www.radiologymasterclass.co.uk/gallery/chest/pulmonary-disease/tuberculosis_tb
- [23] Lloyd, G. (2020, November 3). *COVID-19 discussion pages: - COVID-19 Lung Disease*. Radiology Masterclass -. Retrieved June 2, 2023, from <https://www.radiologymasterclass.co.uk/tutorials/covid-19/covid-19-discussion>
- [24] Mahesh, B. (2020, 1). Machine Learning Algorithms - A Review. *International Journal of Science and Research*, 9(1). https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762_Machine_Learning_Algorithms_-A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf?eid=5082902844932096
- [25] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Education.
- [26] Nanox AI | Nanox AI's solutions help identify early signs of chronic disease within CT scans. (n.d.). Nanox Vision. Retrieved June 13, 2023, from <https://www.nanox.vision/ai>
- [27] Oracle. (n.d.). Java. Java | Oracle. Retrieved June 4, 2023, from <https://www.java.com/en/>
- [28] Ortiz-Toro, C., García-Pedrero, A., Lillo-Saavedra, M., & Gonzalo-Martín, C. (2022, 06). Automatic detection of pneumonia in chest X-ray images using textural features. *Computers in Biology and Medicine*, 145. <https://www.sciencedirect.com/science/article/pii/S001048252200258X>
- [29] O'Shea, K., & Nash, R. (2015, 12 2). *An Introduction to Convolutional Neural Networks*. <https://arxiv.org/pdf/1511.08458.pdf>

- [30] PostgreSQL. (n.d.). PostgreSQL. Retrieved June 4, 2023, from
<https://www.postgresql.org/about/>
- [31] PrimeNG. (n.d.). PrimeNG - Angular UI Component Library. Retrieved June 4, 2023, from <https://primeng.org>
- [32] Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mahbub, Z. B., Ayari, M. A., & Chowdhury, M. E.H. (2020). *Reliable Tuberculosis Detection using Chest X-ray with Deep Learning, Segmentation and Visualization* (Vol. 8). IEEE Access. 10.1109/ACCESS.2020.3031384
- [33] Salian, I. (2018, August 2). *Difference Between Supervised, Unsupervised, & Reinforcement Learning*. NVIDIA Blog. Retrieved June 13, 2023, from
<https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/>
- [34] Spring Boot. (n.d.). Spring Boot. Retrieved June 4, 2023, from
<https://spring.io/projects/spring-boot>
- [35] Sternberg, R. J. (n.d.). *Human intelligence | Definition, Types, Test, Theories, & Facts*. Encyclopedia Britannica.
<https://www.britannica.com/science/human-intelligence-psychology>
- [36] TensorFlow. (n.d.). TensorFlow.org. Retrieved June 3, 2023, from
<https://www.tensorflow.org>
- [37] *Tuberculosis (TB) Chest X-ray Database*. (n.d.). Kaggle. Retrieved June 7, 2023, from
<https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-data-set>
- [38] Turing test. (n.d.). Wikipedia. Retrieved June 13, 2023, from
https://en.wikipedia.org/wiki/Turing_test
- [39] Pneumonia in children. World Health Organization (WHO). Retrieved June 13, 2023, from <https://www.who.int/news-room/fact-sheets/detail/pneumonia>

- [40] *WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data.* (n.d.). World Health Organization (WHO). Retrieved June 13, 2023, from <https://covid19.who.int/>
- [41] *The top 10 causes of death.* (2020, December 9).. World Health Organization (WHO). Retrieved June 17, 2023, from <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [42] *Tuberculosis.* (2023, April 21). World Health Organization (WHO). Retrieved June 17, 2023, from <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>