



Departamentul Automatică și Informatică Industrială
Facultatea Automatică și Calculatoare
Universitatea POLITEHNICA din București



LUCRARE DE DIPLOMĂ

Detectarea melanoamelor

MelanoMaven – utilizarea rețelelor neuronale convoluționale în imagistica medicală

Coordonatori

Prof. Dr. Ing. Mihnea-Alexandru Moisescu

As. Drd. Ing. Miruna-Elena Iliuță

Absolvent

Tănăsescu Dimitrie

2023

Cuprins

Cuprins.....	2
Abstract.....	4
Glosar de termeni și abrevieri.....	5
1. Introducere.....	7
1.1 Obiectivele lucrării.....	7
1.2 Descrierea domeniului.....	7
1.2.1 Utilizarea Rețelelor Neuronale Convoluționale pentru clasificarea imaginilor...	7
1.2.2 Despre cancerul de piele și melanoame.....	8
1.3 Prezentarea capitolelor.....	9
2. Descrierea problemei abordate.....	10
2.1 Formularea problemei.....	10
2.2 Soluții similare.....	11
2.2.1 SolarScan.....	11
2.2.2 MelaFind.....	14
2.3 Cerințe funcționale și nonfuncționale ale aplicației.....	16
2.3.1 Funcționalități ale aplicației.....	16
2.3.2 Cerințe nefuncționale.....	18
3. Stadiul actual în domeniu și selectarea soluției tehnice.....	19
3.1 Stadiul actual al tehnologiilor utilizate pentru dezvoltarea soluției.....	19
3.1.1 Despre Deep Learning și imagistica medicală.....	19
3.1.2 Despre Rețelele Neuronale Artificiale.....	19
3.1.3 Regularizare și Dropout.....	22
3.2 Tehnologii utilizate în dezvoltarea aplicației.....	23
3.2.1 Arhitectura Inception V3.....	23
3.2.2 Arhitectura VGG16.....	25
3.2.3 Telegram chatbots. PTB.....	26
4. Dezvoltarea aplicației.....	28
4.1 Arhitectură.....	28
4.1.1 MelanoMaven v1.....	28
4.1.2 MelanoMaven v2.....	29
4.2 Implementarea aplicației.....	30
4.2.1 MelanoMaven v1.....	30

4.2.2	MelanoMaven v2	32
4.3	Rezultate.....	34
5.	Studiu de caz.....	37
5.1	Utilizare.....	37
5.2	Lansare proiect	39
6.	Concluzii și contribuții	41
6.1	Concluzii	41
6.2	Contribuții personale.....	42
6.3	Dezvoltări ulterioare.....	42
7.	Bibliografie.....	43

Abstract

Proiectul de diplomă își propune dezvoltarea unei soluții proprii pentru detectarea de melanoame folosind rețele neuronale. Pentru elaborarea acesteia au fost efectuate studii și sinteze asupra importanței identificării din timp a cancerelor de piele, a metodelor de diagnosticare, precum și asupra tehnologiilor capabile să faciliteze acest proces de recunoaștere automată a leziunilor maligne. Au fost identificate, analizate și comparate 2 sisteme deja existente, SolarScan și MelanoMaven. Aplicația, pentru care au fost dezvoltate 2 modele de predicție, utilizează arhitectura a 2 rețele neuronale convoluționale a căror performanță este bine documentată: Inception V3 și VGG16. Antrenarea modelelor a fost făcută folosind seturi de date puse la dispoziție de către ISIC. Utilizarea aplicației dezvoltate este facilitată de o interfață implementată folosind un Telegram chatbot. Sunt documentate procesul de dezvoltare a aplicației, modul de utilizare și mentenanță al acesteia, precum și rezultatele obținute împreună cu idei de dezvoltare ulterioară menite să îmbunătățească performanța modelelor.

Glosar de termeni și abrevieri

ANN = Artificial Neural Network; model matematic construit structural și arhitectural prin imitarea sistemului nervos biologic. Mai multe celule de calcul, numite neuroni, sunt grupate în straturi funcționale

CNN = Convolutional Neural Network; ANN care folosește drept componentă de calcul principală filtre de convoluție

CUDA = Compute Unified Device Architecture; utilitar care reduce timpul de calcul prin paralelizarea proceselor. Suportat de anumite procesoare GPU produse de Nvidia

DCCNN = Densely Connected Convolutional Neural Network; CNN în care fiecare neuron este conectat la toți ceilalți neuroni

DL = Deep Learning; ramură a ML care se ocupă cu studiul de ANNs

DNN = Deep Neural Network; ANN cu o arhitectură complexă și foarte multe straturi interne

DOS = Denial of Service; în context de securitate cibernetică. Atac cibernetic prin care se dorește suprasolicitarea unui server prin trimiterea repetată de request-uri, astfel încât alt utilizator nu poate stabili o conexiune cu serviciul țintă

GPU = Graphics Processing Unit; circuit electronic fabricat pentru a eficientiza consumul de putere de calcul în generarea și procesarea de informații video. Unitățile GPU dedicate se ocupă de administrarea display-ului într-un calculator obișnuit, însă pot fi folosite și pe dispozitive fără interfață grafică. Prin accesarea puterii de calcul paralelă a acestora, GPU-rile sunt utilizate într-o multitudine de domenii

ILSVRC = ImageNet Large Scale Visual Recognition Challenge; competiție care are scopul de a evalua și monitoriza progresul sistemelor de vedere artificială. ImageNet 1K este setul de date pus la dispoziție pentru această competiție

ISIC = The International Skin Imaging Collaboration; organizație care promovează dezvoltarea de soluții pentru diagnosticarea melanoamelor folosind tehnici de DL

ML = Machine Learning; ramură a inteligenței artificiale prin care se urmărește dezvoltarea de algoritmi și soluții ce conduc la construirea unor modele matematice capabile să generalizeze anumite informații despre date, folosind exemple

PTB = Python-Telegram-Bot; utilitar pentru limbajul Python care permite interacțiunea cu Bot API-ul pus la dispoziție de către Telegram

ReLU = Rectified Linear Unit; funcție de activare utilizată în construirea de ANNs

RNN = Recurrent Neural Network; ANN în care conexiunile dintre neuroni sunt structurate ciclic

RPN = Region Proposal Network

VGG = Visual Geometry Group; organizație din cadrul Departamentului de Științe Inginerești al Universității din Oxford. Departamentul se ocupă cu cercetare în domeniul inteligenței artificiale și al vederii artificiale

VGGx = Arhitectură de CNN dezvoltată de către VGG; x reprezintă numărul de straturi de convoluție prezente în arhitectură

1. Introducere

1.1 Obiectivele lucrării

Melanomul este cea mai agresivă formă de cancer de piele, având cea mai mare rată a mortalității și asociind o rată de supraviețuire la 5 ani de 30% pentru pacienții aflați în stadiul 4 [36]. Cu toate acestea, dacă este depistat precoce și diagnosticat în primele stadii, poate fi tratat doar prin eliminare chirurgicală, fără a fi nevoie de radioterapie sau de chimioterapie.

Prin această lucrare este propusă realizarea unor studii asupra metodelor de diagnosticare și recunoaștere a melanoamelor, a tehnologiilor de ML și DL cu ajutorul cărora se poate modela un sistem de diagnostic automat, precum și a soluțiilor deja existente la această problemă.

În urma acestor studii, se dorește realizarea unei aplicații care pune la dispoziție, într-un stadiu incipient, posibilitatea de evaluare preliminară a leziunilor cutanate, efectuată automat, minimizând necesitatea expertizei unui medic dermatolog. Aplicația va fi distribuită publicului prin intermediul unui chatbot pe platforma Telegram, căruia utilizatorii vor putea trimite o fotografie cu leziunea de interes, apoi vor primi un răspuns cu rezultatul evaluării. Aceste predicții nu vor substitui expertiza medicală a unui specialist, ci vor reprezenta o primă evaluare pentru un utilizator care are suspiciuni cu privire la natura leziunilor sale.

1.2 Descrierea domeniului

1.2.1 Utilizarea Rețelelor Neuronale Convoluționale pentru clasificarea imaginilor

CNN sunt folosite tot mai des în aplicații din diverse domenii, cu precădere în vederea artificială. Primele modele – familia LeNet – au fost simple înșiruri de filtre convoluționale ce aveau drept scop extragerea de trăsături [8]. AlexNet, rețea care a câștigat provocarea anuală ILSVRC în 2012 [13], a introdus publicului utilizarea de acceleratoare hardware GPU pentru îmbunătățirea semnificativă a vitezei de calcul.

Utilizarea de DNN facilitează dezvoltarea de soluții pentru clasificarea și segmentarea obiectelor din imagini prin extragerea de trăsături și memorarea acestora. Mai multe filtre de convoluție, de dimensiuni diferite, dispuse pe mai multe niveluri de procesare și conectate între ele, extrag caracteristicile generale ale claselor de imagini utilizate în dezvoltarea modelelor. În cazul abordării de tip Supervised Learning – modelul primește drept informații de intrare imagini cu diferite obiecte de interes și rezultatul așteptat pe care ar trebui să-l returneze la ieșire – în timpul antrenării modelului, aceste trăsături extrase sunt utilizate împreună cu informația de ieșire asociată intrării pentru a calcula eroare de predicție (folosind o anumită funcție de cost) [25].

1.2.2 Despre cancerul de piele și melanoame

Primul factor de risc al apariției și dezvoltării cancerului de piele este nivelul de melanină din piele. Astfel, oamenii cu pielea de culoare mai deschisă (deci cu un nivel de melanină scăzut) – sau cu afecțiuni de piele pre-existente sau congenitale – sunt mult mai susceptibili la apariția leziunilor canceroase. De exemplu, o persoană care suferă de albinism – nivelul de melanină din piele scăzut drastic – prezintă un risc mult mai ridicat la apariția unui cancer cutanat, cu atât mai mult într-o regiune cu expunere largă la radiații solare. [34].



Figura 1: Africa Centrală, copil mult mai sensibil decât alți membri ai comunității

Radiațiile ultraviolete (UV) au cea mai semnificativă prezență în ceea ce privește agenți cancerigeni ai pielii. Nivelul de melanină din piele determină susceptibilitatea indivizilor la diverse afecțiuni deoarece aceasta este utilizată în acțiuni de curățare și reînnoire a țesuturilor pielii și totodată funcționează ca un ecran protector împotriva razelor UV. Dacă pielea nu este protejată adecvat împotriva radiațiilor UV, aceasta devine foarte sensibilă la apariția cancerelor la nivelul celulelor bazale (Stratum basale) și la nivelul celulelor scuamoase. Acestea, împreună cu melanoamele, reprezintă cele mai comune formațiuni canceroase de la nivelul pielii [17].

1.3 Prezentarea capitolelor

Lucrarea este structurată pe 6 capitole principale, după cum urmează.

În primul capitol, *Introducere*, sunt prezentate obiectivele lucrării, domeniul lucrării și este realizată o descriere succintă întregului proiect de licență. A fost acordată o atenție deosebită asupra prezentării domeniului din punct de vedere medical, cu precădere importanța diagnosticării la timp a cancerelor de piele, precum și tehnicilor de diagnostic utilizate curent în dermatologie.

În capitolul doi, *Descrierea problemei abordate* sunt prezentate motivația din spatele dezvoltării soluției precum și 2 sisteme similare – *SolarScan* și *MelaFind*. Totodată sunt enumerate funcționalitățile și cerințele aplicației.

Capitolul trei prezintă studiu asupra tehnologiilor utilizate (secțiunea 3.2), precum și asupra teoriei din spatele acestora (secțiunea 3.1).

Capitolul patru detaliază procesul de dezvoltare al aplicației, etapele și modul de abordare al acestora, totodată și cu prezentarea rezultatelor obținute.

Capitolul cinci conține studiul de caz aferent aplicației. Lucrarea se încheie cu concluziile, contribuțiile personale și cu propunerile pentru dezvoltări ulterioare.

Capitolul șase oferă o prezentare generală a lucrării, cu accent deosebit pe concluzii, îmbunătățirea soluțiilor și posibilități pentru dezvoltare ulterioară.

Lucrarea este finalizată cu un capitol pentru bibliografie, în care sunt menționate toate sursele citate.

2. Descrierea problemei abordate

2.1 Formularea problemei

Cauza principală a decesului în urma dezvoltării de formațiuni maligne nu o reprezintă masa tumorală originală, ci urmările procesului de metastazare – celule canceroase din formațiunea malignă principală se desprind și circulă prin parenchimul – țesut format din celule specifice și spații intercelulare – din vecinătate și apoi ajung în sânge [31]. Acestea circulă prin sistemul cardiovascular pe distanțe largi, până în țesutul altor organe, în care se ancorează. Aceste celule vor crește în masă până la dimensiuni macroscopice, proces cunoscut în oncologie drept colonizare a organului țintă. Prin analiza sângelui unui pacient diagnosticat cu cancer, se observă că numărul de celule canceroase vehiculate prin sânge - a căror rată de supraviețuire este de 0.2% - depășește cu mult numărul de formațiuni metastatice [31].

În ceea ce privește melanoamele, forma de cancer pe care aplicația dorește să o detecteze, acestea sunt foarte periculoase cauzând peste 9000 de decese anual (2019) în Statele Unite, plasându-se pe locul 5 în cele mai comune tipuri de formațiuni maligne [31]. O provocare importantă este recunoașterea și diagnosticarea corectă a acestora: fiind vizual foarte asemănătoare cu alte leziuni de piele, benigne însă, precum nevii și cheratozele, este crucial ca pacientul să efectueze un control medical de la prima suspiciune.

O descoperire fundamentală în ceea ce privește clasificarea leziunilor benigne și canceroase, precum și diagnosticarea corectă a melanoamelor, a fost propunerea regulii de diagnostic ABCD – asimetrie, margini neregulate, culoare variată și diametru mai mare de 6 mm (în engleză Asymmetry, Border, Colour, Diameter) – descrisă pentru prima oară în 1985 [21] (a se vedea exemplul de mai jos). Aceste diferențe fundamentale între cele 2 clase de leziuni – benigne și melanoame – fac posibilă încercarea de a realiza o clasificare, cel puțin preliminară a acestora, fără a mai fi nevoie de o examinare realizată de către un medic oncolog. Trebuie menționat că aceste idei rămân la stadiul de cercetare și potențiale soluții pentru viitor. După cum a fost specificat în paragraful anterior, este puternic recomandat un control de specialitate periodic.



Figura 2: formațiune benignă (nev)

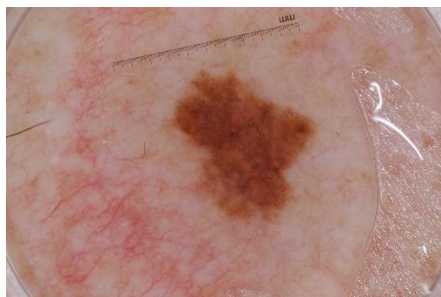


Figura 3: formațiune canceroasă (melanom)

Leziunile benigne precum nevii prezintă o structură regulată, caracterizată de un model asemănător unei grile formate din linii și spații egale, distribuite uniform și organizat pe întreaga suprafață. Liniile acestea prezintă variații neglijabile în ceea ce privește grosimea și culoarea acestora, de obicei fiind maronii [21].

Formațiunile canceroase în schimb se prezintă sub forma unei rețele neregulate, în care spațiile sunt de forme și dimensiuni variate iar liniile grilei sunt de grosimi diferite. Aceste linii neregulate se formează în urma traumatismului suferit la nivelul țesutului epitelial din cauza melanocitelor canceroase. Culoarea unei astfel de leziuni variază între nuanțe de maro, negru și gri. Aproximarea de gri și negru se datorează creșterii nivelului local de melanină, ca răspuns la apariția de țesut canceros. În cazul melanoamelor cauzate direct de radiația soarelui, leziunile prezintă de obicei linii maro sau gri, care se formează într-un model cu linii frânte și unghiuri ascuțite [21].

Este necesar de menționat faptul că există o linie critică în ceea ce privește diagnosticul și operarea tumorilor de piele, deoarece clasificarea unor leziuni benigne ca fiind cancer este de evitat. Abordarea prin care toate tumorile melanocitare sunt înlăturate chirurgical nu este fezabilă. Cu toate că această soluție ar elimina fiecare neoplasm canceros, totodată ar presupune că leziunile care nu reprezintă un pericol sunt tratate ca tumorile maligne, strategie non-practică și inefficientă.

Aplicația dezvoltată, în forma curentă, dorește să pună la dispoziție publicului o modalitate de a decide dacă leziunea de interes este de natură malignă sau nu, fără să fie necesară o examinare medicală. Aplicația poate fi îmbunătățită pentru asistarea medicilor dermatologi, făcând trecerea de la un program utilizat de publicul larg către un instrument medical în domeniul telemedicinii.

2.2 Soluții similare

2.2.1 SolarScan

Scurtă prezentare

Datorită importanței diagnosticării la timp – după cum a fost prezentat în secțiunea anterioară – mai multe instituții au investit resurse pentru a îmbunătăți procesul de diagnosticare precoce. Una dintre cele mai eficiente soluții a fost rezultatul cercetării conduse de către Australian Commonwealth Scientific and Industrial Research Organization [12].

SolarScan modelul este descris în articolul „The Performance of SolarScan” [18], a obținut rezultate foarte bune în diagnosticarea de melanoame pe bază de melanocite. Setul de date pe care a fost antrenat conținea 2430 de imagini cu leziuni – 382 fiind țesut canceros – dintre care 78 au fost imagini de înaltă calitate, preluate de la

dermoscopii ale unor pacienți cu un istoric medical detaliat. Pentru compararea rezultatelor obținute de model, a fost implicat în proiect personal medical din diverse instituții care, la analiza setului de date, au obținut diverse valori pentru sensibilitate și specificitate:

- 3 experți în dermoscopie (90% și 59%)
- 4 dermatologi (81% și 60%)
- 3 dermatologi rezidenți (85% și 36%)
- 3 practicanți în medicină generală (62% și 63%)

Rezultatele au arătat că SolarScan a produs rezultate comparabile sau mai bune decât ale medicilor, obținând o sensibilitate de 91% și o specificitate de 68%. Validitatea acestor rezultate trebuie testată în studii clinice [26].

Achiziția de date

În decursul a 5 ani (Iunie 1998 – Septembrie 2003) au fost colectate imagini cu diferite leziuni cutanate din 9 clinici, dintre care 7 de specialitate – conform articolului [18] acestea sunt:

- The Sydney Melanoma Unit (Sydney Melanoma Diagnostic Centre), Sydney, Australia
- Skin and Cancer Associates, Miami, Florida
- Department of Dermatology, University of Tübingen, Tübingen, Germany
- The Skin and Cancer Foundation, Sydney, Australia;
- KellyDerm, the private clinic of one of the authors (J.K.), Melbourne, Australia
- South East Dermatology and the Princess Alexandra Hospital, Brisbane, Australia

Pe lângă achiziția de imagini pentru formarea setului de date, a fost colectat și un istoric al pacienților pentru fiecare leziune, informații de interes fiind cu precădere evoluția în ultimii doi ani a leziunilor: dacă au sângerat fără să sufere vreun traumatism mecanic, dacă s-a modificat culoarea acestora sau dacă au crescut în suprafață sau volum [18].

Eșantioanele obținute au fost filtrate în funcție de cât de fezabile au fost pentru proiect; imaginile cu o rezoluție mult prea mică, care nu au putut fi calibrate sau care prezentau dificultăți cauzate de prezența altor artefacte în jurul leziunilor – fire de păr, bule de aer sau distorsiuni cauzate de mișcarea pacienților – precum și leziuni diagnosticate cu carcinom cutanat non-melanom (cancer la nivelul celulelor bazale sau scuamoase) au fost înlăturate din studiu [18].

Algoritmi și metode folosite

Setul de date a fost împărțit în două secțiuni cu raport 2:1, eșantioane pentru antrenarea modelului și eșantioane pentru testarea acestuia. Fiecare imagine corespunzătoare unui anumit diagnostic a primit un scor (eng: weight) și a fost clasificată după tipul de leziune – benignă sau canceroasă. Aceste scoruri au fost acordate în funcție de potențialul stabilirii unui diagnostic greșit, cât de ușor a fost stabilirea diagnosticului clinic și frecvența tipului de leziune. Modelul a fost optimizat folosind toate aceste scoruri, informațiile privind istoricul pacienților, precum și prin comparație cu diagnosticile stabilite de către medici [18].

Distribuția frecvențelor de apariție a melanoamelor a fost implementată folosind Modele Gauss cu câte 2, 3 sau 4 componente distincte, folosind tehnici de optimizare pentru maximizarea mediei statistice. Cele mai bune rezultate au fost obținute de modele cu câte 2 componente. Probabilitatea unui rezultat pozitiv (melanom detectat – real pozitiv sau fals pozitiv) a fost calculată ca raport între valoarea probabilității de melanom obținută și valoarea ratelor pentru orice leziune. S-a observat că observațiile matematice și calculele empirice au fost mult mai puțin susceptibile la erori cauzate de zgomot statistic pentru eșantioane aflate în extreme [18].

Rezultate

Prin antrenarea modelului pe setul de 1644 de imagini cu leziuni cutanate, dintre care 260 au reprezentat melanoame, un algoritm de diagnostic automat, capabil de a recunoaște și distinge formațiunile canceroase (melanoame) de leziuni benigne, însă pigmentate suspect. Modelul obținut a fost utilizat și testat pe un set diferit, cu 786 de imagini, dintre care 122 reprezentând melanoame. Rezultatele obținute au arătat o acuratețe bună a modelului, înregistrând valori ale sensibilității de 90-91% și valori ale specificității de 61-65% [18].

Performanța modelului este evidențiată grafic prin trasarea curbei ROC (Receiver operating characteristic). Curvele ROC descriu, prin trasarea grafică, dependența dintre clasificările efectuate cu succes (True Positives) și identificările defectuoase (False Positives), folosind diverse praguri de decizie [15]. Rezultatele False Positive sunt o problemă deosebită în aplicațiile medicale, deoarece pot declanșa acțiuni redundante sau invazive fără să fie necesare – ex. înlăturarea chirurgicală a unui nev benign diagnosticat ca melanom – precum și supunerea pacientului la stres psiho-emoțional.

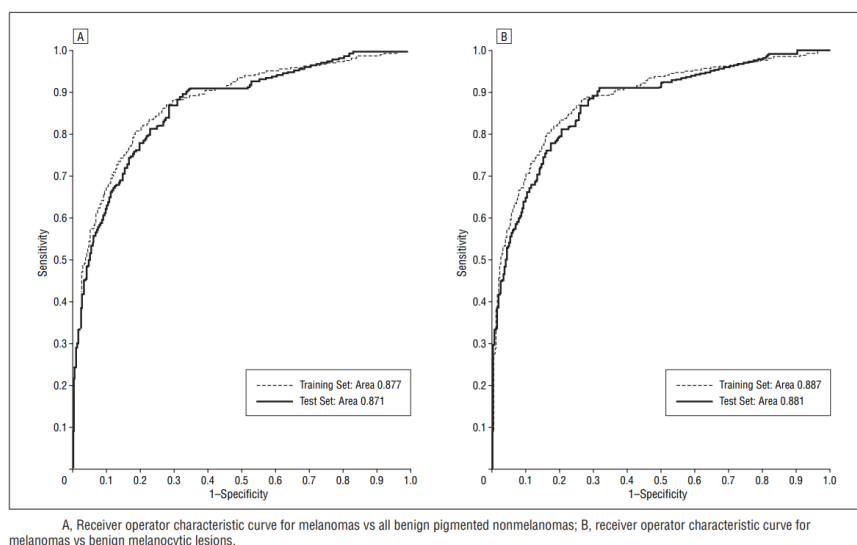


Figura 4: Curba ROC pentru modelul SolarScan [18]

2.2.2 MelaFind

Scurtă prezentare

Un alt instrument menit să asiste în imagistica medicală și în recunoașterea melanoamelor a fost MelaFind, a cărui dezvoltare a început în 2008 la institutul Electro-Optical Sciences, Inc. Irvington, NY, Statele Unite ale Americii și care a fost aprobat de către FDA (US Food and Drug Administration – agenția federală care guvernează reglementările privind sănătatea publică în SUA) în 2011. În urma cercetării depuse, profesorul Michael Elbaum și echipa sa au declarat că modelul antrenat a obținut performanțe foarte bune [26]; în urma procesului de validare încrucișată – *validarea încrucișată* (cross-validation) reprezintă procesul prin care performanța unui model este estimată; setul de date este împărțit în 2 subseturi, unul folosit pentru antrenare, în timp ce testarea capacității de predicție a modelului se face folosind cel de-al doilea subset; validarea încrucișată reprezintă standardul actual în antrenarea modelelor [29] – s-au obținut valori de minim 95% pentru sensibilitate și valori de apropiate de 70% pentru specificitate.

Proiectul și-a propus dezvoltarea unui sistem care să asiste în detectarea melanoamelor în stare incipientă. Pentru dezvoltarea instrumentului au fost colectate 1831 de eșantioane – imagini cu leziuni cutanate – extrase de la peste 1300 de pacienți. Dintre acestea, 1632 de imagini (127 care prezentau țesut malign) au fost desemnate eligibile și relevante pentru obiectivul cercetării [19].

În urma experimentului și a testării sistemului dezvoltat, s-a ajuns la concluzia că MelaFind poate fi folosit cu succes în imagistica medicală, obținând scoruri de performanță foarte bune – sensibilitate de 98.4% și specificitate de 98.3% [19].

În 2017 proiectul a fost suspendat definitiv de către Strata Skin Sciences Inc [39].

Achiziția de date

În afara imaginilor achiziționate special pentru MelaFind, au fost de asemenea folosite 3 imagini adiționale [19], extrase la standarde profesionale:

- 2 au fost realizate folosind camera Fuji FinePix Pro SR, la distanțe de 55, respectiv 20 de cm față de leziune
- cea de-a treia imagine, o captură dermoscopică de contact, a fost realizată folosind camera Nikon Coolpix 4300

Baza de date digitală folosită pentru dezvoltarea sistemului a inclus, pe lângă eşantioanele fotografice, informații demografice despre pacienții diagnosticați, factori de risc pentru cancerul de piele și caracteristici evolutive ale melanoamelor [19] – metodologia ABCD (descrișă în secțiunea 2.1), regresie sau îngrijorare a pacienților. Metodologia finală aplicată pentru colectarea și compunerea setului de date a fost aplicarea unui studiu intensiv asupra țesuturilor fiecărei leziuni („*A histologic specimen with the standard hematoxylin-eosin staining was provided for each lesion.*”) [19].

Mod de funcționare

Principiul de funcționare al MelaFind este evidențiat în tabelul prezentat mai jos [19].

Table 1. Algorithm for Final Histologic Diagnosis ^a				
Primary Pathologists		Pathologist 3	Rereview by Pathologist Diagnosing Melanoma	Final Diagnosis
1	2			
Melanoma	Melanoma	NA	NA	Melanoma
Melanoma	HGDN	NA	NA	Melanoma
Melanoma	Other	Melanoma	NA	Melanoma
Melanoma	Other	HGDN	NA	Melanoma
Melanoma	Other	Other	Melanoma	Melanoma
Melanoma	Other	Other	HGDN	Other
Melanoma	Other	Other	Other	Other
HGDN	HGDN	NA	NA	HGDN
HGDN	Other	HGDN	NA	HGDN
HGDN	Other	Other	NA	Other
Other	Other	NA	NA	Other

Tabelul 1 – Algoritmul folosit de MelaFind

Rezultate

Rezultatele studiului sunt prezentate în tabelul de mai jos [19].

Dermatologic Diagnosis	Histologic Diagnosis, No.		
	Melanoma	HGDN, AMP, or AMH	Other
Melanoma	13	1	6
Melanoma cannot be ruled out			
Likely melanoma (67%-99% likelihood)	30	4	46
Possible melanoma (34%-66% likelihood)	44	14	471
Unlikely melanoma (1%-33% likelihood)	38	27	827
Not melanoma			
Clinical concern only, dermoscopy negative	1	1	27
Nonmelanoma skin cancer	0	0	13
Patient's concern	1	1	55
Physical discomfort	0	0	8
Cosmetic purposes	0	0	4
Total	127	48	1457

Tabelul 2 – Diagnosticul histologic al leziunilor evaluabile

2.3 Cerințe funcționale și nonfuncționale ale aplicației

2.3.1 Funcționalități ale aplicației

În această secțiune sunt descrise toate funcționalitățile aplicației dezvoltate:

- Aplicația primește drept input o fotografie cu o leziune a pielii;
- În cazul în care fotografia este într-un format special (cum este formatul HEIC de la Apple), aceasta este convertită automat în format JPEG;
- După încărcarea și eventuala conversie a imaginii de input, aceasta este cache-uită în memoria programului și redimensionată pe structura pe care modelul o acceptă (64 x 64);

- Aplicația procesează imaginea primită și returnează o predicție de clasificarea a acesteia (Benign sau Malign);
- Conexiune dintre server și client, precum și interfața cu utilizatorul au fost realizate printr-un chatbot pe platforma de mesagerie Telegram;
- De fiecare dată când serverul este pornit, fișierul *users.log*, în care sunt stocați utilizatorii aplicației, trebuie șters manual, iar utilizatorii vor trebui să se înregistreze din nou folosind comanda */start*;
- Dacă serverul poate fi menținut online fără întreruperi, atunci această necesitate devine redundantă, întrucât platforma Telegram apelează automat comanda */start* în momentul în care utilizatorul încearcă să acceseze serviciul unui chatbot pentru prima dată;
- Chatbot-ul dispune de 4 comenzi:
 - o */start* – returnează un mesaj de bun venit utilizatorului și dacă acesta accesează chatbot-ul pentru prima oară de la ultima repornire a serverului, înregistrează id-ul acestuia
 - o */help* – returnează lista de comenzi disponibile, precum și o scurtă descriere a acestora
 - o */info* – returnează o descriere succintă a proiectului, precum și mențiunea că chatbot-ul MelanoMaven nu dispune de performanța de diagnostic a unui medic dermatolog și că este recomandat un control medical pentru orice suspiciune a utilizatorului
 - o */predict* – permite clasificarea leziunii prin intermediul imaginii de input;
- Pentru a proteja serverul împotriva spam-ului sau a atacurilor de tip Denial of Service, serviciul de predicție al aplicației nu poate fi utilizat fără a apela în prealabil comanda */predict*. Trimiterea unei imagini fără apelarea comenzii va returna un mesaj de atenționare asupra utilizatorului și totodată nu fotografia trimisă nu va fi încărcată pe server. Aplicația dispune de un cronometru virtual unic asociat fiecărui utilizator, ce oprește posibilitatea de apelare a comenzii */predict* la intervale de timp mai scurte de 20s. În momentul în care utilizatorul este înregistrat, cronometrul unic asociat acestuia este creat automat iar timpul de așteptare este setat la zero. După prima utilizare a aplicației, cronometrul începe să funcționeze normal, iar limita de minim 20s între utilizări este setată;
- În continuare, pentru a preveni abuzurile, utilizarea comenzii */predict* este restricționată, nu utilizarea aplicației în sine. Dacă utilizatorul apelează comanda, însă în decursul a mai puțin de 20 de secunde nu trimite o fotografie, însă solicită din nou */predict*, acesta va pierde dreptul de utilizarea a aplicației

pe durata intervalului curent și va primi un mesaj de atenționare care anunță că trebuie așteptată terminarea intervalului curent;

- Chatbot-ul nu este configurat pentru a putea simula conversații cu utilizatorii sau pentru a interacționa cu altceva în afară de fotografii:
 - o dacă sunt trimise drept input alte tipuri de fișiere care nu sunt imagini într-un format suportat, acestea vor fi complet ignorate
 - o dacă este trimis vreun mesaj text, chatbot-ul va informa utilizatorul că mesajele de tip text nu sunt procesate și îi va reaminti acestuia de comanda */help*

2.3.2 Cerințe nefuncționale

În această secțiune sunt descrise cerințele nefuncționale precum și performanțele dorite ale aplicației dezvoltate:

- Aplicația este accesibilă de pe orice dispozitiv care poate utiliza platforma Telegram. Nu este necesar utilizatorului decât să își creeze un cont și, în cazul utilizării de pe un dispozitiv mobil, să descarce aplicația Telegram. Chatbot-ul poate fi accesat prin introducerea textului “*MelanoMaven v2*” în bara de căutare;
- Restricții minime asupra fișierelor de input: utilizatorul nu trebuie să formateze într-un anumit fel fotografia pe care se va face predicția. În momentul în care comanda de predicție este apelată, chatbot-ul va oferi utilizatorului indicații despre felul în care trebuie realizată fotografia, împreună cu un exemplu vizual. Interfața Telegram permite realizarea de fotografii instantaneu din interiorul aplicației, precum și decuparea sau scalarea acestora înainte de a fi trimise. Aplicația dezvoltată are integrat ascuns un sistem de conversie între diferite formate de imagini, astfel încât orice fotografie statică este acceptată ca input;
- Pentru a porni și menține un server care să ruleze aplicația (modelul de predicție, utilitățile asociate pentru prelucrarea imaginilor și codul sursă al chatbot-ului) este necesară o mașină cu suport Linux – Ubuntu 22.04. Serverul poate fi rulat pe o mașină virtuală sau direct pe un sistem care are instalată distribuția de Linux menționată, însă necesar pe dispozitivul fizic este un accelerator grafic GPU compatibil cu platforma CUDA. Proiectul folosește limbajul de programare și scripting Python, cea mai nouă variantă (la momentul iunie 2023) a versiunii Python 3.10 (3.10.6). Pentru instalarea tuturor dependențelor proiectului, precum și a pachetelor Python necesare, este suficientă rularea cu drepturi de administrator a scriptului *install.sh*

3. Stadiul actual în domeniu și selectarea soluției tehnice

3.1 Stadiul actual al tehnologiilor utilizate pentru dezvoltarea soluției

3.1.1 Despre Deep Learning și imagistica medicală

Deep Learning (învățare profundă) este o metodologie din cadrul învățării automate (Machine Learning) bazată pe utilizarea de ANN, cele mai des utilizate fiind RNN, folosite cu precădere pentru aplicații din domeniul procesării de date numerice, detectarea și procesare de text sau în crearea de muzică și CNN – utilizate în mare parte pentru prelucrarea și analiza datelor multimedia sau în procesarea de limbaj natural) [28]. Denumirea de “profundă” provine din faptul că aceste rețele au încorporate mai multe niveluri de procesare a datelor [11]. Aceste niveluri sunt plasate secvențial și înglobează un număr mare de operații matematice, astfel încât la fiecare nivel ajunge informație mai abstractizată decât pe cel precedent. Cea mai comună metodă de antrenare este metoda învățatului supervizat, în care setul de date este compus din 2 părți – imagini cu leziuni cutanate și informații (*labels*) despre acestea (ex: canceroase sau benigne) [7].

Tehnicile de Deep Learning, cu precădere CNN, au o aplicabilitate variată în vederea artificială. În imagistica medicală, abordarea folosind aceste rețele a prezentat rezultate promițătoare în domenii precum oftalmologie, dermatologie și radiologie, prin asistarea la diagnosticat, identificat, detectat și clasificat diferite afecțiuni [7]. Motivul din spatele acestui succes este faptul că Rețelele Neuronale Convoluționale se apropie de precizia umană în ceea ce privește clasificarea și segmentarea obiectelor din imagini.

În acest sens, CNN au fost folosite cu succes în Transfer Learning – utilizarea unei rețele deja antrenate pe un set de date larg, folosind procedeul de fine tuning pe setul de date specific problemei în cauză. Modelele sunt antrenate pe seturi de date foarte largi, în 2 etape. Inițial rețeaua este învățată informații primitive simple – linii, curbe, culori, forme de bază – apoi nivelurile mai înalte ale modelului primesc date particularizate și mai specializate (cum ar fi diferite diagnostice) [7]. CNN funcționează prin aplicarea unor filtre de convoluție – corespunzătoare unor matrice de diferite dimensiuni, în general 1x1, 3x3 sau 5x5 – diferite la fiecare nivel al rețelei. Aceste filtre sunt aplicate pe imaginea originală transmisă ca input până când aceasta este transformată într-o repartitie a distribuției claselor de imagini obținute în antrenarea rețelei [7].

3.1.2 Despre Rețelele Neuronale Artificiale

ANN sunt metode de calcul avansat dezvoltate în cadrul învățării profunde. Scopul acestora este implementarea de algoritmi care să se apropie sau să reproducă felul în care funcționează componente ale sistemului nervos din ființele vii.

O rețea neuronală este structurată în mai multe niveluri, fiecare îndeplinind un anumit scop și totodată compusă din unități funcționale numite noduri (omoloage ai neuronilor). Fiecare nod are asociat o valoare de cost și un prag de decizie [38]. Valorile de cost sunt modificate la fiecare iterație din timpul antrenării rețelei, modelul final fiind o variantă a rețelei care minimizează funcția (valorile) de cost. Dacă ieșirea din nodul curent nu depășește valoarea pragului de decizie asociat nodului, neuronul în cauză este “aruncat” (influența acestuia este ignorată) în pasul curent. Fiecare neuron primește drept input valorile de ieșire ale fiecărui neuron de pe stratul anterior și totodată trimite la intrarea fiecărui neuron de pe stratul următor propriul output – excepție fiind neuronii de pe primul, respectiv ultimul strat.

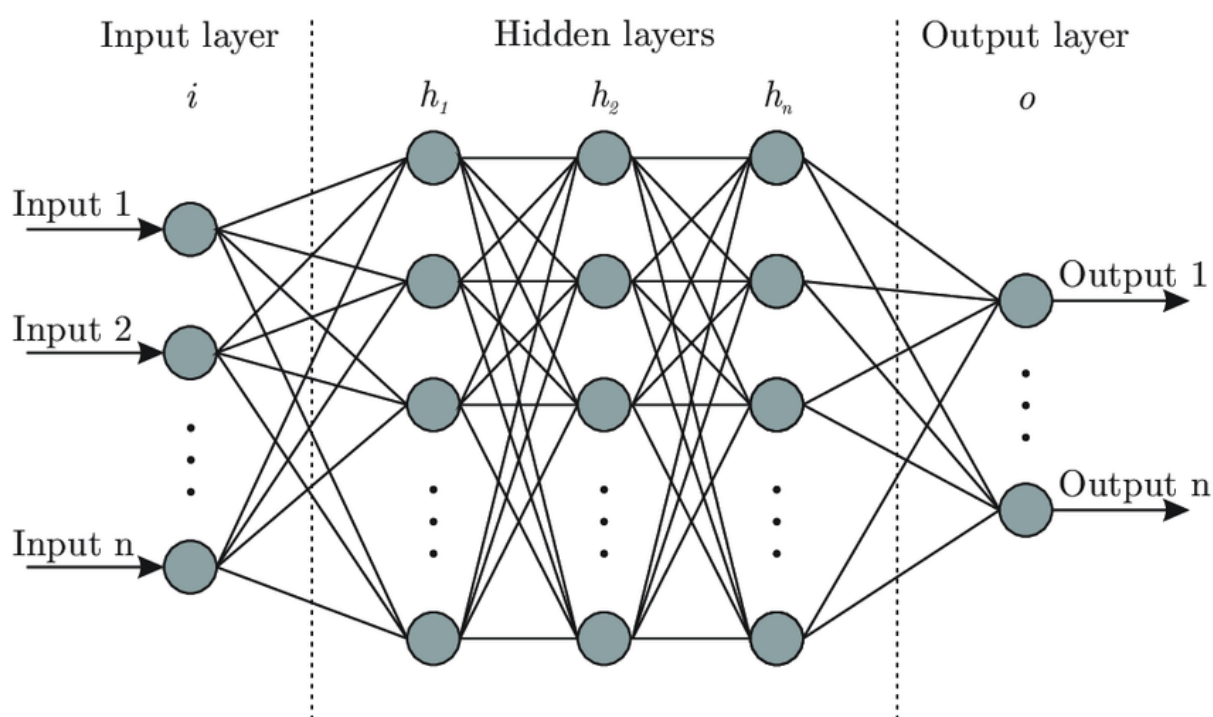


Figura 5: Structura de bază a unei ANN [2]

CNN, o clasă importantă de rețele artificiale, sunt folosite cu precădere în problemele de clasificare sau detectare a obiectelor, recunoaștere facială sau vedere artificială. Creșterea puternică a utilizării acestora se datorează capacității de antrenare eficientă a acestora pe sisteme de calcul GPU [13], utilizarea revoluționară a ReLU care accelerează puternic procesul de antrenare și optimizare a modelului [3] sau accesul facil la seturi largi de date pentru antrenare – precum ImageNet 1K care a pus la dispoziție peste 100.000 de clase de obiecte, fiecare clasă dispunând de aproximativ 1000 de eșantioane (imagini) [32].

O variantă de îmbunătățire a CNN standard au fost propunerile de RPN, respectiv DCCNN. RPN primesc drept input o imagine oarecare și returnează o serie de obiecte

dreptunghiulare care modelează trăsăturile extrase, împreună cu un scor asociat ce reprezintă potrivirea obiectului recunoscut în clasa din care ar trebui să facă parte [22].

Creșterea constantă a numărului de niveluri (implicit a nodurilor) din rețelele noi dezvoltate amplifică problema gradientului neglijabil: în urma mai multor pași de antrenare, gradientul (folosit metodă de optimizare) funcțiilor de cost se apropie puternic de zero, ceea ce pune pauză procesului efectiv de modelare a rețelei [30] – o valoare de cost minimă nu este sinonim cu un model capabil să generalizeze orice input din categoriile pe care a fost antrenat. O soluție propusă de Huang, G. este aceea de DCCNN, rețele în care fiecare nod al fiecărui nivel este adițional conectat cu ceilalți neuroni din același nivel [9]. Rețeaua lui Huang a reușit pe durata antrenării să facă diferența între informațiile învățate în iterațiile anterioare și trăsături noi obținute.

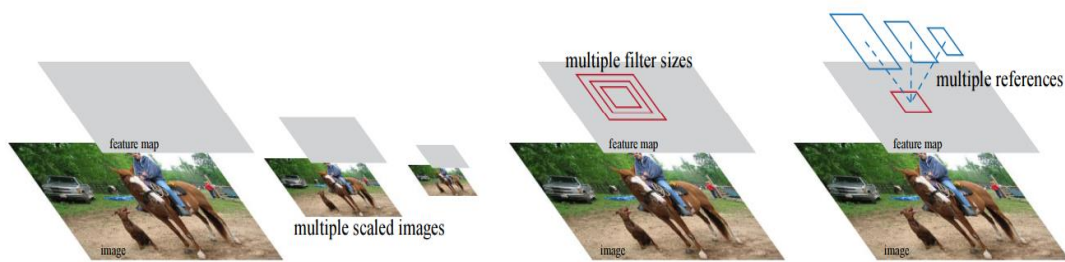


Figura 6: Funcționarea unei RPN [22]

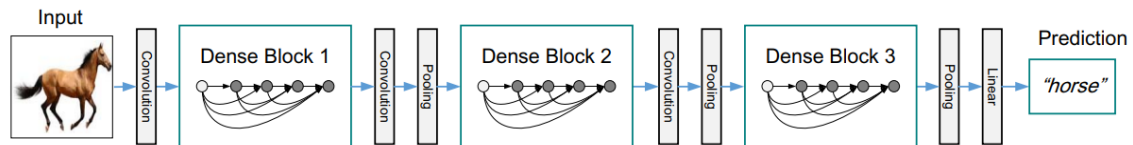


Figura 7: Structura funcțională a unei DCCNN [9]

Layers	Output Size	DenseNet-121($k = 32$)	DenseNet-169($k = 32$)	DenseNet-201($k = 32$)	DenseNet-161($k = 48$)
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Tabelul 3: Rețeaua lui Huang [9]

3.1.3 Regularizare și Dropout

Regularizarea reprezintă procesul prin care complexitatea modelului este redusă în timpul procesului de antrenare prin creșterea artificială a valorilor de cost.

În cele mai utilizate forme de regularizare, la valoarea de cost curentă se adaugă normele 1 sau 2 ale vectorului de valori ai neuronilor rețelei, scalate cu un anumit factor (factor de regularizare) – regularizare L1, respectiv L2. Regularizarea crește valoarea funcției de cost proporțional cu valorile unice ale parametrilor rețelei și numărul acestora; drept urmare, cu cât modelul este mai complex (mai mulți neuroni), cu atât funcția de cost va fi umflată artificial. Scopul regularizării este de a reduce probabilitatea de apariție a problemei de overfitting – fenomenul de overfitting apare atunci când modelul memorează trăsăturile setului de antrenare și nu mai reușește să generalizeze un output corect pentru input nou [6].

O altă metodă de regularizare este metoda de Dropout. Procesul de Dropout presupune ca la fiecare pas al antrenării modelului, valori ale neuronilor să fie complet ignorate până la iterația următoare. În Regularizarea Dropout Bernoulli – cunoscută și ca dropout canonic – fiecare neuron din rețea are asociată o șansă de 50% ca output-ul său să fie ignorat [16]. Regularizarea cu Dropout, în mod similar cu regularizările L1 și L2, dorește prevenirea procesului de overfitting prin reducerea aleatoare a complexității modelului. Dacă în cazurile L1, respectiv L2, complexitatea este redusă indirect folosind “taxarea” acestuia prin creșterea artificială a funcției de cost, procesul de Dropout execută direct această reducere prin anularea efectivă a valorilor neuronilor din rețea.

Network architecture	L_2 regularization		Dropout	
	% accuracy	% Rate of change	% accuracy	% Rate of change
784-100-10	96.97	-	96.51	-
784-200-10	97.02	0.05	97.49	1.02
784-300-10	97.05	0.03	97.82	0.34
784-400-10	97.05	0.00	97.97	0.15
784-500-10	96.90	-0.15	97.99	0.02
784-600-10	96.99	0.09	98.09	0.10
784-700-10	96.96	-0.03	98.19	0.10
784-800-10	96.86	-0.10	98.29	0.10
784-900-10	96.91	0.05	98.22	-0.07
784-1000-10	96.61	-0.31	98.17	-0.05

Tabelul 4: Efectul regularizării asupra performanței diferitelor arhitecturi pe setul de date cu diferite stiluri de cifre arabe MNIST [6]

3.2 Tehnologii utilizate în dezvoltarea aplicației

3.2.1 Arhitectura Inception V3

Descriere generală

Inception V3 este o Rețea Neuronală Convoluțională propusă în articolul “Rethinking the Inception Architecture for Computer Vision” [24], publicat în 2015, de către Christian Szegedy (Google), ca o îmbunătățire a rețelelor Inception anterioare.

Cercetarea în clasa de rețele Inception are ca scop dezvoltarea unor motoare de inteligență artificială mai eficiente din punct de vedere al puterii de calcul necesare; Inception V3 s-a dovedit a fi mai eficientă decât VGGNet sau AlexNet [34].

Rețeaua este utilizată cu precădere în domeniul medical, unde este folosită pentru capacitatea de clasificarea a diferitelor formațiuni. În prezent este folosită pentru cercetarea leucemiei [20].

Arhitectură

La baza rețelei se află Modulele Inception, structuri care au în componență filtre de convoluție – de dimensiuni 1x1, 3x3 și 5x5 – precum și un block 3x3 de max pooling al nivelului anterior [33]; feature pooling reprezintă procesul prin care trăsăturile extrase și răspunsurile filtrelor sunt grupate printr-un procedeu statistic, care va avea drept rezultat o distribuție comună a trăsăturilor identificate [27].

Un puternic dezavantaj al acestor module este că necesită o putere de calcul foarte mare, care crește direct proporțional cu numărul de filtre de convoluție dintr-un nivel. O soluție propusă a fost eficientizarea calculelor cu blocuri simple, înainte ca informația să fie transmisă mai departe. Astfel, filtre de dimensiune 1x1 for executa operații de reducere înaintea operațiilor efectuate de către filtrele de dimensiuni superioare și ulterior procesului de max pooling [23]. Cele 2 structuri arhitecturale ale rețelei sunt evidențiate în figurile 8 [33] și 9 [27].

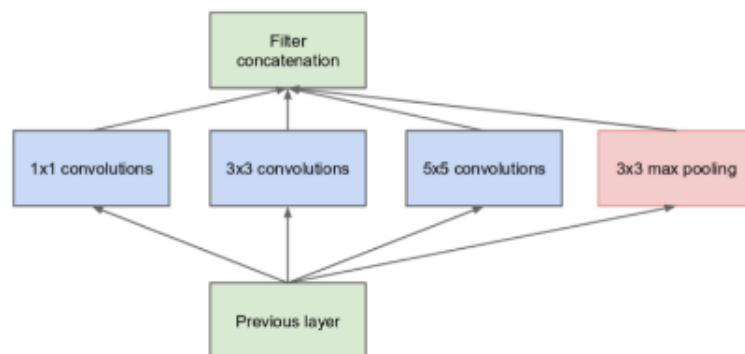


Figura 8: diagrama unui Modul Inception simplu

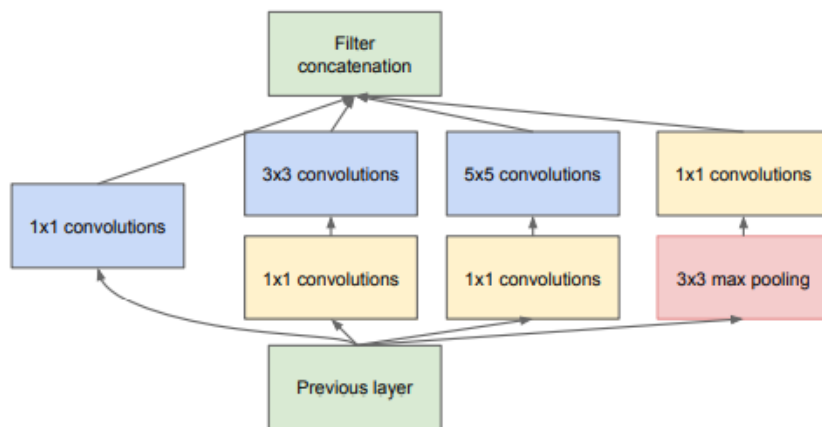


Figura 9 – diagrama unui Modul Inception care implementează operații de reducere

În tabelul 5 [23] este detaliată arhitectura unui exemplu de model care folosește rețeaua Inception V3 – GoogLeNet. Sunt descrise tipurile de operații efectuate în diferite blocuri, dimensiunile acestora precum și ale datelor de ieșire, numărul de operații și consumul de putere de calcul (numărul de operații efectuate)

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Tabelul 5: Arhitectura modelului GoogLeNet

3.2.2 Arhitectura VGG16

Descriere generală

VGGx este o CNN propusă de K. Simonyan și A. Zisserman, membrii ai VGG, Departamentul de Științe Inginerești, Universitatea din Oxford. Aceasta a fost elaborată în cadrul articolului “Very Deep Convolutional Networks for Large-Scale Image Recognition”, publicat în 2015 [14].

Arhitectura VGG dispune de 2 versiuni utilizate intensiv, VGG16 și VGG19 – fiecare prezentând câte 3 niveluri complet conectate, diferența fundamentală fiind dată de numărul de straturi de convoluție: 16, respectiv 19.

Arhitectură și rezultate

Stratul de început al rețelei este configurat să primească drept input imagini în format RGB, de dimensiune 224x224. Rețeaua poate fi configurată și pentru input de dimensiuni diferite prin modificarea stratului de input sau prin introducerea de Adaptive Average Pooling – operație de pooling care primește drept input dimensiunea imaginilor de intrare, dimensiunea output-ului rețelei și calculează dimensiunea kernel-ului rețelei necesar pentru păstrarea limitelor dimensionale transmise [10]. Imaginile primite sunt trecute prin straturile de convoluție ce implementează filtre de dimensiuni mici – 3 x 3; unul dintre straturi folosește filtre de dimensiune 1 x 1. Trăsăturile extrase sunt colectate prin max-pooling în ferestre de dimensiune 2 [14]. Fiecare strat are înglobată rectificare folosind ReLU.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Tabelul 6: Configurații pentru diferite arhitecturi VGG [14]

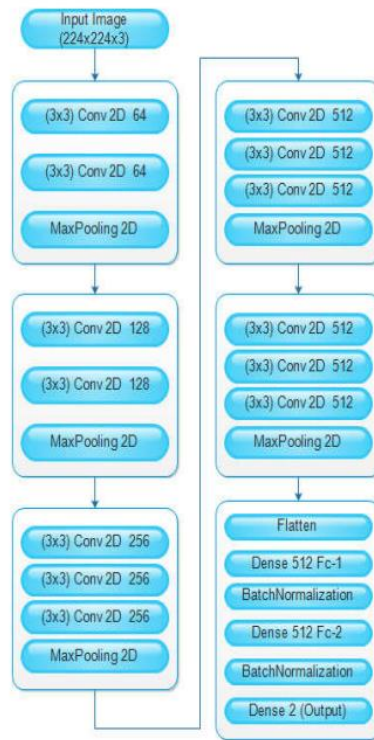


Figura 10: Model de implementare a arhitecturii VGG16 [5]

network	feature layers	MM AUC	SK AUC	avg. AUC
AlexNet	FC8	80.67	94.95	87.81
AlexNet	all FC	82.81	96.65	89.73
VGG16	FC8	82.61	90.94	86.78
VGG16	all FC	82.06	95.46	88.76
ResNet-18	FC	81.00	91.93	86.47
ResNet-18	FC + last convol. layer	82.81	94.22	88.51
AlexNet + VGG16 fusion	all FC	83.56	97.05	90.30
AlexNet + ResNet-18 fusion	all FC	83.53	97.05	90.29
VGG16 + ResNet-18 fusion	all FC	83.69	95.97	89.83
fusion of all networks	all FC	83.83	97.55	90.69

Tabelul 7: Performanța modelului VGG16 pe setul ISIC 2017 [4], comparație cu alte rețele

3.2.3 Telegram chatbots. PTB

PTB [37] este o interfață, dezvoltată independent, pentru limbajul Python a API-ului pentru managementul de chatbots al platformei Telegram. Pachetul pune la dispoziție posibilități de a defini comenzi pentru chatbot, metode de a trata și gestiona comunicarea cu utilizatorii sau definirea de mesaje și răspunsuri automate. Proiectul dispune de un repository și wiki pe GitHub, precum și o documentație publică.

Aplicația dezvoltată este distribuită prin intermediul Telegram pentru a asigura accesul la aceasta fără a exista restricții asupra platformei utilizate – o aplicație creată de la zero ar fi necesitat suport și pentru sistemele IOS și pentru sistemele Android în cazul dispozitivelor mobile și suport multi-platformă pentru sisteme desktop. Cu toate că implementarea accesului la aplicație printr-o interfață WEB ar fi rezolvat aceste probleme, s-a dorit utilizarea unei platforme a cărei funcționalitate și mod de lucru erau cunoscute *a priori*. Totodată a fost considerat că publicarea soluției dezvoltate printr-un mediu de sine stătător (o aplicație mobilă/desktop preexistentă) prezintă mai multă accesibilitate și eficiență comparativ cu utilizarea tehnologiilor de dezvoltare WEB.

Pachetul PTB a fost utilizat doar pentru gestionarea, controlul și definirea logicii chatbot-ului. Crearea în sine a acestuia a fost efectuată din interiorul platformei Telegram, prin intermediul serviciului BotFather.

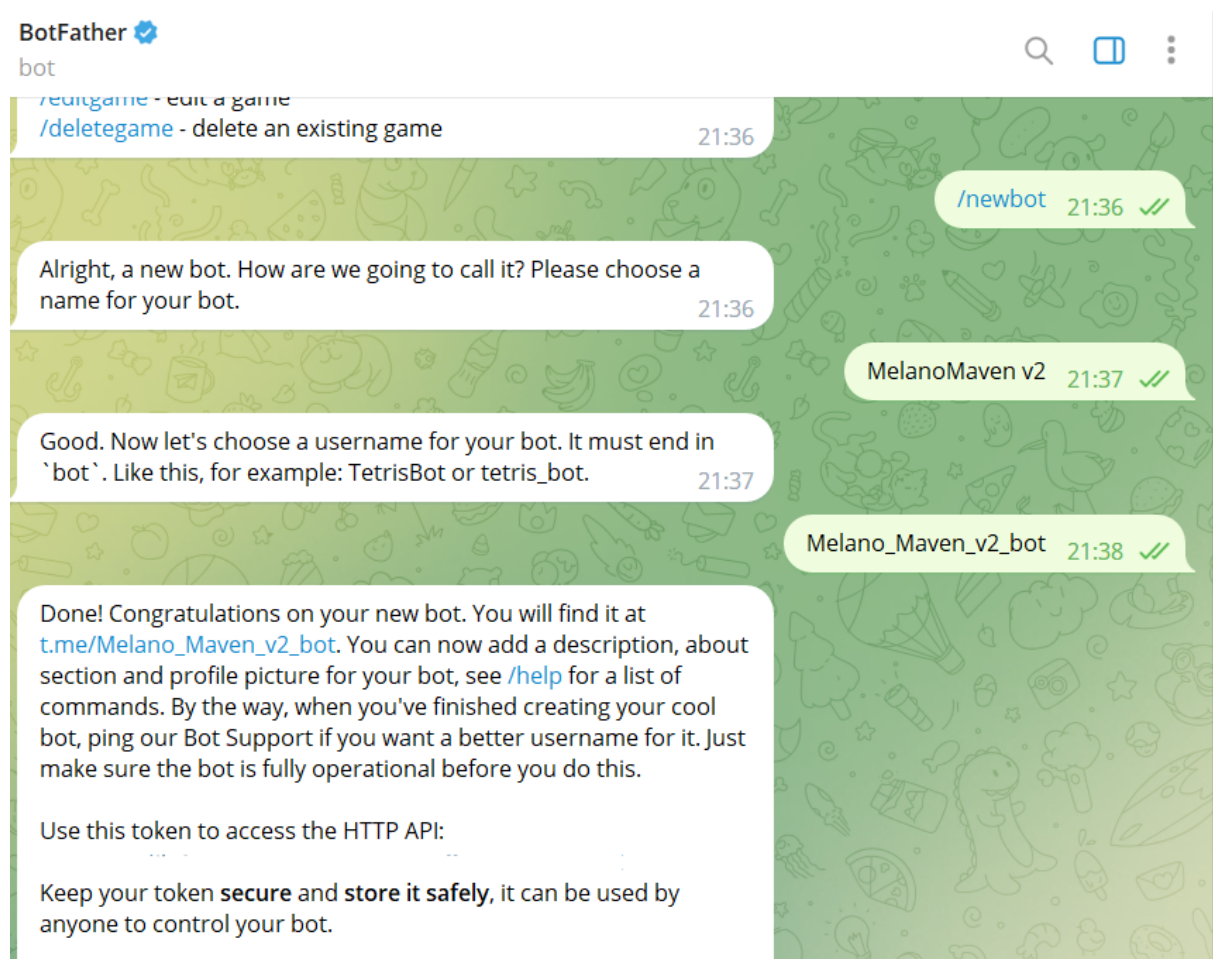


Figura 11: Exemplu de creare de chatbot folosind BotFather

4. Dezvoltarea aplicației

4.1 Arhitectură

Aplicația este alcătuită din 2 componente principale – modelul care face predicția diagnosticelor și chatbot-ul pentru Telegram care funcționează drept interfață cu utilizatorul. În dezvoltarea modelului, am implementat 2 soluții diferite:

- MelanoMaven v1, folosind CNN Inception v3 prin intermediul librăriei Keras
- MelanoMaven v2, unde am dezvoltat propria rețea neuronală folosind librăria PyTorch, bazându-mă pe arhitectura VGG16

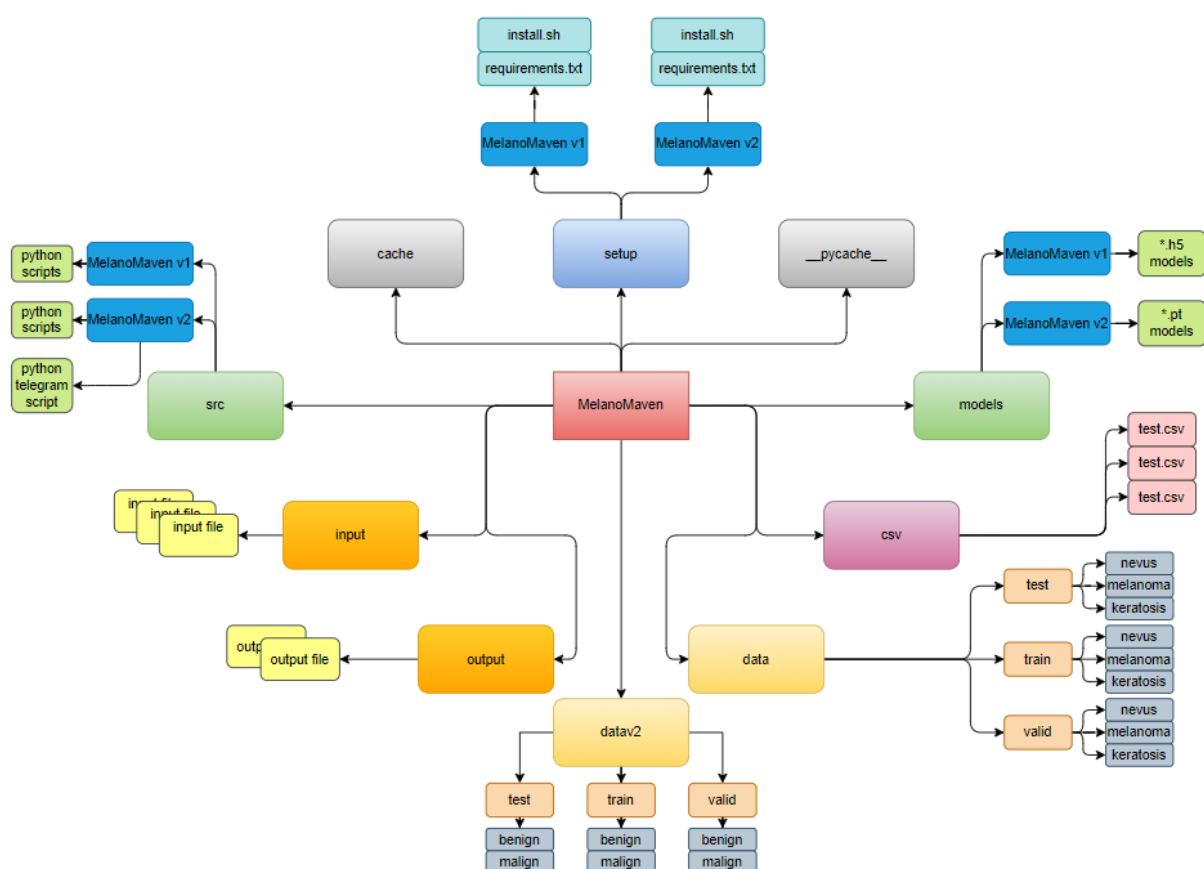


Figura 12: organizarea structurală a directorului aplicației

4.1.1 MelanoMaven v1

Setul de date utilizat a fost pus la dispoziție de către ISIC (The International Skin Imaging Collaboration). Organizația lansează anual provocări în domeniul vederii artificiale utilizate în aplicații medicale, cu precădere clasificarea de leziuni benigne și maligne. Setul de date folosit este cel propus pentru ISIC Challenge 2017 [4]. Acesta dispune de 2750 total de eșantioane, împărțite în 3 categorii:

- 2000 de imagini în subsetul pentru antrenarea modelelor, dintre care 374 melanoame și 1626 leziuni benigne (1372 nevi și 254 cheratoze)
- 150 de imagini în subsetul pentru validarea modelelor, dintre care 30 melanoame și 120 leziuni benigne (78 nevi și 42 cheratoze)
- 600 de imagini în subsetul pentru testarea modelelor, dintre care 117 melanoame și 483 leziuni benigne (393 nevi și 90 cheratoze)

Arhivele corespunzătoare setului de date (stocate în AWS S3 Buckets) sunt descărcate și organizate în directorul de lucru folosind scriptul *getdata.py* (figura 13) implementat cu utilitarul *get_file* din pachetul tensorflow.

```
import os
import zipfile
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
from tensorflow.keras.utils import get_file

def get_data():                                     # function to download and save the data

    training = "https://s3-us-west-1.amazonaws.com/udacity-dlnfd/datasets/skin-cancer/train.zip"
    validation = "https://s3-us-west-1.amazonaws.com/udacity-dlnfd/datasets/skin-cancer/valid.zip"
    testing = "https://s3-us-west-1.amazonaws.com/udacity-dlnfd/datasets/skin-cancer/test.zip"

    for cur, link in enumerate([training, validation, testing]):
        cur_file = f"cur{cur}.zip"
        name = os.path.join(os.getcwd(), cur_file)
        data_f = get_file(origin=link, fname=name)
        with zipfile.ZipFile(data_f, "r") as f:
            f.extractall("../data")
        os.remove(cur_file)
```

Figura 13: scriptul *getdata.py* folosit pentru salvarea și organizarea setului de date

După salvarea imaginilor, acestea sunt prelucrate și pregătite pentru antrenarea modelului. Astfel eșantioanele sunt redimensionate pentru a corespunde formatului arhitecturii Inception V3 – 299 x 299 – și le este atribuită o etichetă – benign sau malign – în format CSV (comma separated values) ce conține calea relativă a imaginilor, urmând să fie înregistrate ca dataframes folosind pachetul pandas; formatul dataframe permite utilizarea cu ușurință a seturilor de date pentru învățare automată.

4.1.2 *MelanoMaven v2*

Pentru îmbunătățirea versiunii anterioare a proiectului, a fost determinată ca fiind necesară echilibrarea setului de date. Folosind setul ISIC 2017 extins cu melanoamele din setul ISIC 2020 pentru a completa datele pentru antrenare și validare, s-a ajuns la un total de 3283 de eșantioane, distribuite astfel:

- 2439 de imagini în subsetul pentru antrenarea modelelor, dintre care 814 melanoame și 1625 leziuni benigne
- 244 de imagini în subsetul pentru validarea modelelor, dintre care 124 melanoame și 120 leziuni benigne
- 600 de imagini în subsetul pentru testarea modelelor, dintre care 117 melanoame și 483 leziuni benigne

Setul de date a fost organizat manual și încărcat pentru utilizare generală pe platforma *Hugging Face* [35]. Descărcarea și salvarea acestuia este realizată prin intermediul scriptului *getdata.py*.

```

training = "https://huggingface.co/datasets/Dimit/MelanoMaven_ISIC_2017-2020/resolve/main/train.zip"
validation = "https://huggingface.co/datasets/Dimit/MelanoMaven_ISIC_2017-2020/resolve/main/valid.zip"
testing = "https://huggingface.co/datasets/Dimit/MelanoMaven_ISIC_2017-2020/resolve/main/test.zip"

try:
    os.mkdir("../Licenta_cod/datav2")
    os.mkdir("../Licenta_cod/datav2/train")
    os.mkdir("../Licenta_cod/datav2/valid")
    os.mkdir("../Licenta_cod/datav2/test")
except:
    pass

def download_data(url, destination):
    response = requests.get(url)
    with open(destination, "wb") as file:
        file.write(response.content)
    with zipfile.ZipFile(destination, "r") as zip_ref:
        zip_ref.extractall(destination.replace(destination.split('/')[-1], ''))
    os.remove(destination)

```

Figura 14: snippet din codul pentru salvarea noului set de date

4.2 Implementarea aplicației

4.2.1 *MelanoMaven v1*

Folosind rețeaua Inception V3, implementată în TensorFlow, și setul de date menționat în secțiunea 4.1.1, a fost dezvoltată o aplicație care primește drept input o imagine cu o leziune cutanată și trimite drept output un scor pentru leziunea respectivă, evaluând probabilitatea de a fi benignă sau canceroasă.

Pentru instalarea tuturor instrumentelor necesare funcționării aplicației, se va rula, din directorul “setup/MelanoMaven v1”, cu drepturi de administrator, scriptul *install.sh*.

Imaginile pe care se dorește aplicat modelul de predicție vor fi plasate în directorul “input”, iar aplicația va fi rulată din directorul “src/MelanoMaven v1”, folosind scriptul *getpredict.py*, împreună cu argumentele opționale pentru scalarea sau prelucrarea imaginii de input:

python3 getpredict.py 1.5 False

În cazul în care sunt utilizate imagini într-un format care nu este suportat sau dacă se dorește redimensionarea imaginii sau centrarea pe leziunea cutanată, prin transmiterea parametrilor, așa cum este prezentat în exemplul anterior, de zoom – valoare numerică – sau de procesare input – valoare booleană True sau False – vor fi apelate funcții dedicate prelucrării de imagini din scriptul *zoomcroptransform.py*, care utilizează utilitare din pachetele OpenCV și TensorFlow, precum și MagickWand. Totodată scriptul se ocupă de identificarea și citirea automată a tuturor fișierelor din directorul input.

```
def zoom(img, scale):
    # function to zoom in to the center of the image to ensure the lesion
    # of interest is the main focus of the image
    height, width, _ = [ scale * val for val in img.shape ]
    cx, cy = width/2, height/2

    img = cv.resize( img, (0, 0), fx=scale, fy=scale)
    img = img[ int(round(cy - height/scale * 0.5)) : int(round(cy + height/scale * 0.5)),
               int(round(cx - width/scale * 0.5)) : int(round(cx + width/scale * 0.5)),
               : ]

    return img

def transform(source_img):
    # function to transform the image to jpeg format

    save = "input/" + source_img.split('.')[0] + '.jpeg'
    img = Image(filename=source_img)
    img.format = 'jpeg'
    img.save(filename=save)
    os.remove(source_img)

def decode(img):
    # function to decode image and resize it to 299x299, as required by InceptionV3 model

    ext = img.split('.')[1]
    img_path = tf.io.read_file(img)

    match ext:
        case 'jpg':
            img_proc = tf.image.decode_jpeg(img_path, channels=3)
        case 'jpeg':
            img_proc = tf.image.decode_jpeg(img_path, channels=3)
        case 'png':
            img_proc = tf.image.decode_png(img_path, channels=3)
        case _:
            print("Error: Image format not supported")
    img_proc = tf.image.convert_image_dtype(img_proc, tf.float32)
    return tf.image.resize(img_proc, [299, 299])
```

Figura 15: metode implementate în scriptul *zoomcroptransform.py*

Modelul a fost antrenat pe Rețeaua InceptionV3, folosind Transfer Learning, pachetul TensorFlow Hub și setul de date în format dataframe. Deoarece scopul este automatizarea unei probleme de clasificare binară – ce presupune obținerea unor răspunsuri numerice în intervalul [0, 1], unde nivelul de 0 reprezintă un scor negativ (benign) iar nivelul de 1 un scor pozitiv (formațiune malignă) – pentru antrenarea și optimizarea modelului s-au folosit drept parametrii funcția de activare sigmoid, funcția de cost binary cross entropy și algoritmul de optimizare rmsprop. Modelul a fost trecut prin 100 de epoci, cu batch size constant de 64 de eșantioane per pas. La fiecare pas se salvează valoarea pierderilor și este păstrat modelul cu pierderea minimă.


```

Epoch 36: val_loss did not improve from 0.38606
31/31 [=====] - 117s 4s/step - loss: 0.2685 - accuracy: 0.8967 - val_loss: 0.4297 - val_accuracy: 0.8203
Epoch 37/50
31/31 [=====] - ETA: 0s - loss: 0.2732 - accuracy: 0.8866
Epoch 37: val_loss did not improve from 0.38606
31/31 [=====] - 118s 4s/step - loss: 0.2732 - accuracy: 0.8866 - val_loss: 0.4329 - val_accuracy: 0.8281
Epoch 38/50
Epoch 44: val_loss did not improve from 0.37568
31/31 [=====] - 116s 4s/step - loss: 0.2745 - accuracy: 0.8810 - val_loss: 0.4292 - val_accuracy: 0.8359
Epoch 45/50
31/31 [=====] - ETA: 0s - loss: 0.2675 - accuracy: 0.8942
Epoch 45: val_loss improved from 0.37568 to 0.35301, saving model to models\melanoma_model_0.353.h5
31/31 [=====] - 118s 4s/step - loss: 0.2675 - accuracy: 0.8942 - val_loss: 0.3530 - val_accuracy: 0.8516
Epoch 46/50
31/31 [=====] - ETA: 0s - loss: 0.2634 - accuracy: 0.8891
Epoch 46: val_loss did not improve from 0.35301
31/31 [=====] - 115s 4s/step - loss: 0.2634 - accuracy: 0.8891 - val_loss: 0.4365 - val_accuracy: 0.8203
Epoch 47/50

```

Figura 16: Procesul de antrenare prin epoci

4.2.2 *MelanoMaven v2*

După evaluarea rezultatelor obținute de MelanoMaven v1 (rezultate care vor fi prezentate în secțiunea 4.3), a fost luată hotărârea de a aborda diferit problema. Un motiv fundamental pentru rezultatele slabe obținute a fost setul de date neechilibrat, problemă care a fost rezolvată după cum a fost elaborat în secțiunea 4.1.2. S-a considerat că doar prin simpla completare a datelor nu va fi obținută o diferență majoră în rezultate, așa că a fost aleasă construirea unei soluții asupra căreia să fie exercitat mai mult control. Pentru aceasta s-a decis utilizarea librăriei PyTorch pentru a configurarea propriei rețele (figura 17), bazată pe arhitectura VGG16. Din rețeaua preexistentă a fost eliminat ultimul strat și înlocuit cu un clasificator binar implementat cu componentele liniare utilizate în arhitectura VGG16 (a căror dimensiune a fost redusă; s-a luat decizia de a sacrifica o parte din complexitatea rețelei pentru a scădea posibilitate de overfitting și totodată puterea de calcul necesară pentru antrenarea modelului), 2 componente ReLU și 2 componente de Dropout; de asemenea a fost adăugat modelului o componentă pentru Adaptive Pooling, în cazul în care se folosesc la input imagini de dimensiuni diferite (implicit am setat modelul să accepte dimensiuni de 64 x 64, din nou pentru a reduce puterea de calcul necesară). S-a experimentat cu mai multe valori și în final a fost stabilit că o probabilitate de Dropout de 30% oferă rezultate adecvate.

```

import torch.nn as nn
from torchvision import models

class VGG16Model(nn.Module):
    # custom model for binary classification based on VGG16
    def __init__(self):
        super().__init__()
        self.features = nn.Sequential(*list(models.vgg16(weights=None).features.children())[:-1]) # remove last layer
        self.avgpool = nn.AdaptiveAvgPool2d((4, 4))
        self.classifier = nn.Sequential(
            nn.Linear(512*4*4, 4096),
            nn.ReLU(True),
            nn.Dropout(0.3),
            nn.Linear(4096, 512),
            nn.ReLU(True),
            nn.Dropout(0.3),
            nn.Linear(512, 2),
        )

    def forward(self, x):
        x = self.features(x)
        x = self.avgpool(x)

```


Figura 17: Implementarea modelului MelanoMaven v2

Pentru antrenarea modelului, a fost aleasă o dimensiune de batch redusă, 8 eşantioane per batch, iar modelul a fost antrenat timp de 100 de epoci, salvând doar cel mai bun model de la fiecare iterație, i.e. doar dacă acesta a produs o valoare de cost mai mică decât cel mai bun model anterior. Pentru optimizarea modelului, a fost utilizată regularizarea L2, cu o constantă de regularizare de 0.0005. La fiecare epocă de antrenare modelul a fost validat pe setul de date specific, pentru a măsura capacitatea de generalizare a datelor a acestuia.

```
def build(model, epochs = 20, batch_size = 16, criterion = None, optimizer = None, train_ds = None, valid_ds = None):

    best_val_loss = inf

    for epoch in range(epochs):

        train_loss = train(model, batch_size, criterion, optimizer, train_ds)
        valid_loss = validate(model, batch_size, criterion, valid_ds)

        print(f'Epoch: {epoch+1}/{epochs}.. Training loss: {train_loss}.. Validation Loss: {valid_loss}')

        if valid_loss < best_val_loss:

            print(f'Better model found: \n\t{best_val_loss:0.4f} -> {valid_loss:0.4f} \n Saving model...")
            torch.save(model.state_dict(), 'MelanoMaven_VGG16.pt')
            best_val_loss = valid_loss

def main():

    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model = VGG16Model().to(device)

    dataset = MelanoMavenDataset()
    epochs, batch_size = 100, 8
```

Figura 18: snippet din scriptul de antrenare *train.py*

Modelul este accesibil prin interfața de Telegram implementată ca un chatbot – MelanoMaven v2. Imaginea pentru care se dorește predicția va fi trimisă ca input direct prin chat-ul platformei Telegram, iar chatbot-ul va trimite automat un mesaj cu rezultatul predicției. Scriptul chatbot-ului trebuie menținut activ în permanență; la fiecare resetare a acestuia, fișierul *users.log* trebuie curățat manual iar utilizatorii vor trebui să se înregistreze folosind comanda */start*. Motivul pentru care acești pași trebuie îndepliniți este implementarea unui sistem minimalist de securitate împotriva abuzurilor și atacurilor DOS. În momentul în care un utilizatorul pornește chatbot-ul pentru prima dată (moment în care comanda */start* este apelată automat), acesta este înregistrat în sistem și îi este asignat un cronometru virtual. Utilizatorul nu poate cere mai multe predicții la intervale mai scurte de 20 de secunde. Deoarece nu a fost dezvoltată o bază de date pentru acest cronometru virtual, acesta există doar cât timp serverul este pornit. La resetarea acestuia, cronometrul virtual este instanțiat și resetat, drept urmare utilizatorii trebuie să se înregistreze din nou. Chatbot-ul este configurat să accepte 4 comenzi, să primească imagini și să răspundă cu instrucțiuni de utilizare la orice mesaj de tip text pe care îl primește. Dacă este trimis un fișier care nu este poză, acesta va fi ignorat. Imaginea folosită drept poză de profil pentru MelanoMaven v2 a fost generată folosind DALL·E 2 al OpenAI [40].

4.3 Rezultate

După cum a fost indicat în secțiunea 4.2.2, MelanoMaven nu a obținut rezultate foarte bune. Cu toate că acuratețea de 79% poate lăsa de înțeles că modelul a avut o performanță destul de bună, aceste rezultate sunt înșelătoare. Din cauza setului de date deloc echilibrat, modelul a prezentat dificultăți puternice în recunoașterea formațiunilor maligne, în ciuda setării unui prag de decizie scăzut (0.23). Acuratețea de 79% s-a datorat faptului că setul de test este compus din 80% imagini cu leziuni benigne. Prin analiza matricei de confuzie (figura 20) se observă performanța reală (și scăzută) a modelului.

```
-----
Accuracy and loss before thresholding:
Loss: 0.4543, Accuracy: 0.7883
-----

19/19 [=====] - 47s 2s/step
2023-05-29 16:52:48
-----
Accuracy after thresholding:
Accuracy: 0.7883
-----

2023-05-29 16:52:48
```

Figura 19: rezultatele modelului MelanoMaven v1

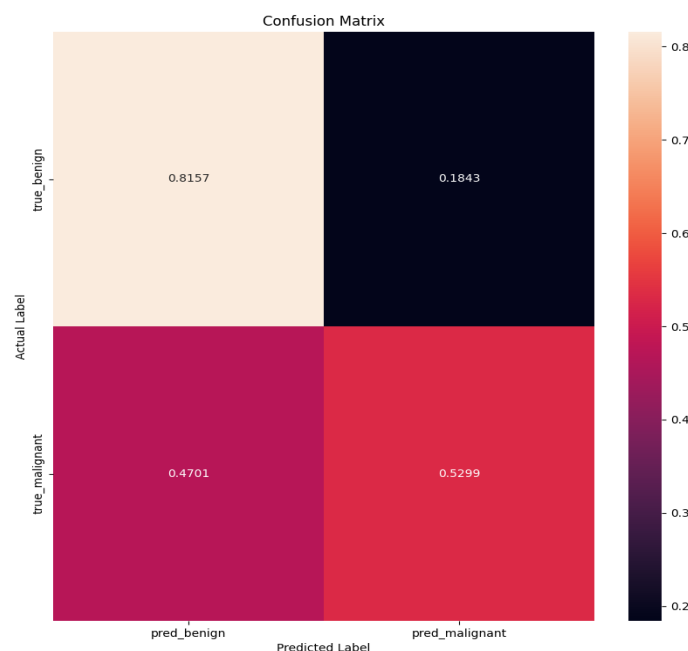


Figura 20: Matricea de confuzie a rezultatelor MelanoMaven v1

A doua versiune a modelului a arătat o îmbunătățire în rezultate. Cu toate că nu a obținut cele mai bune rezultate (setul de antrenare fiind în continuare nu foarte echilibrat, 66% fiind imagini cu leziuni benigne), MelanoMaven v2 a arătat că învață să detecteze caracteristicile leziunilor benigne precum și ale melanoamelor. Rezultatele obținute, folosind un prag de decizie de 0.45, pot fi observate în figurile 21 și 22.

```
MelanoMaven VGG16 Model
Test Accuracy of Benign: 67.0807%
Test Accuracy of Malignant: 35.0427%
Total Prediction Accuracy: 60.8333%
Total Prediction Report: 365.0/600.0
Test Loss: 0.0448
Sensitivity: 0.3504
Specificity: 0.6708
Precision: 0.2050
Recall: 0.3504
F1 Score: 0.2587
```

Figura 21: Performanțele modelului

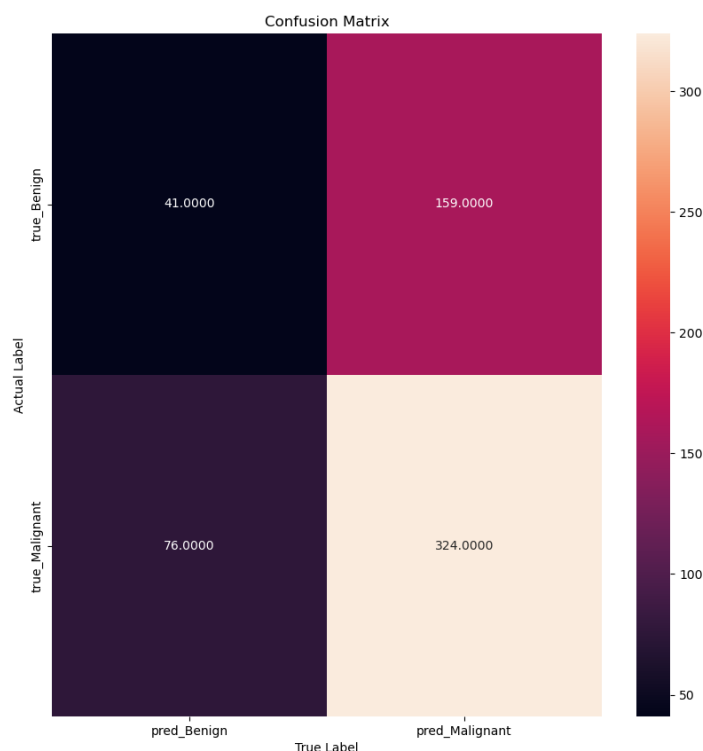


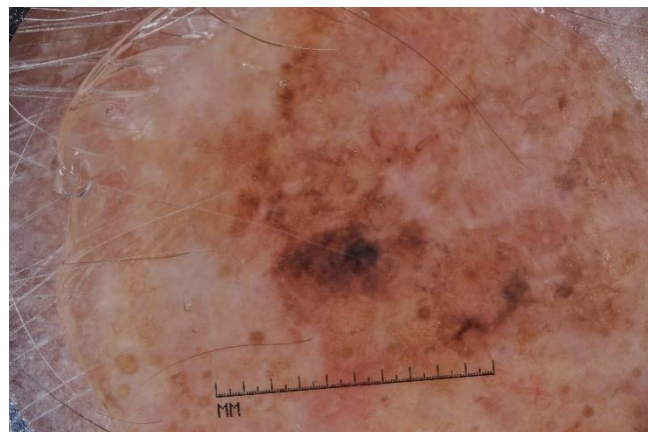
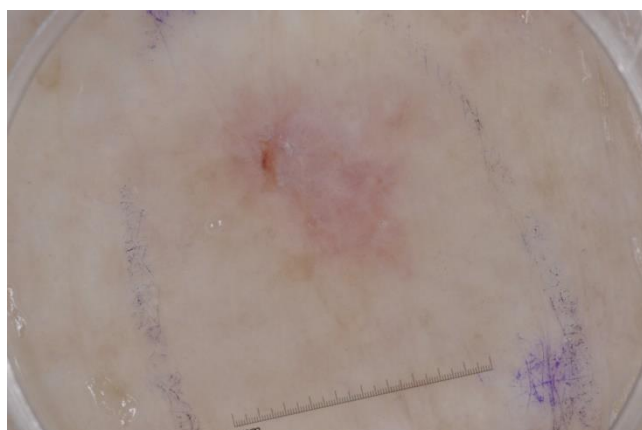
Figura 22: Matricea de confuzie a modelului MelanoMaven v2

```
26
27     img = Image.open(image_path)
28     img = transform(img)
29     img = img.cuda() if torch.cuda.is_available() else img
30
31     img = img.unsqueeze(0)
32     output = model(img)
33     pred = torch.softmax(output, dim=1)
34     if pred[0, 1] >= threshold:
35         pred = 'Malignant'
36     else:
37         pred = 'Benign'
38     print(f"Prediction: {pred}")
39     return pred
40
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
• dimit@DESKTOP-AB2BP0E:~/Licenta_cod/src/MelanoMaven v2$ python3 predict.py ../../datav2/test/benign/ISIC_0012266.jpg
Prediction: Benign
• dimit@DESKTOP-AB2BP0E:~/Licenta_cod/src/MelanoMaven v2$ python3 predict.py ../../datav2/test/malign/ISIC_0013473.jpg
Prediction: Malignant
• dimit@DESKTOP-AB2BP0E:~/Licenta_cod/src/MelanoMaven v2$
```

Figura 23: Rezultate pe imagini de test benigne și maligne alese aleatoriu



Figurile 24a (benign) și 24b (malign): Leziunile clasificate corect

5. Studiu de caz

Pentru utilizarea aplicației sunt necesari mai mulți pași, atât din perspectiva utilizatorului, cât și a dezvoltatorului. Aceștia vor fi descriși în secțiunile următoare.

5.1 Utilizare

Pentru utilizarea aplicației, este necesar să ca utilizatorul să aibă înregistrat un cont de Telegram valid. În cazul utilizării unui dispozitiv mobil, este necesară și instalarea aplicației mobile; pentru folosirea de pe un calculator, instalarea aplicației desktop este opțională – se poate utiliza Telegram Web.

După conectarea cu succes pe Telegram, utilizatorul va tasta în bara de căutare (figura 23) “MelanoMaven”

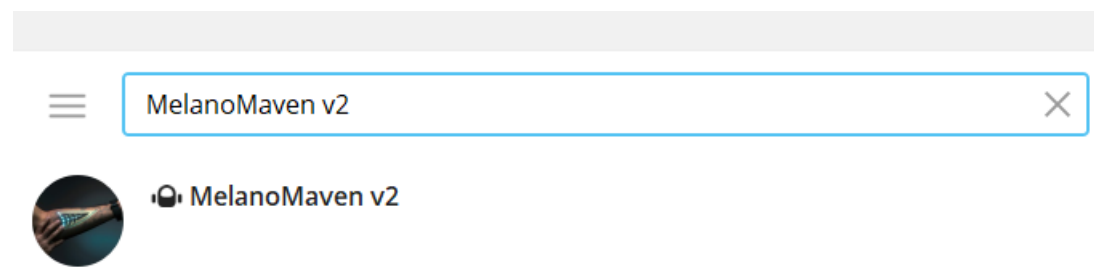


Figura 25: Adăugare chatbot-ului în lista utilizatorului

Pentru prima interacțiune cu chatbot-ul, utilizatorul va apăsa pe butonul de start (care apelează implicit comanda `/start`), va fi înregistrat în program și i se va asocia un cronometru. Este recomandată apelarea comenzii `/help` (figura 23) pentru a vizualiza toate comenzile disponibile și funcționalitățile acestora.

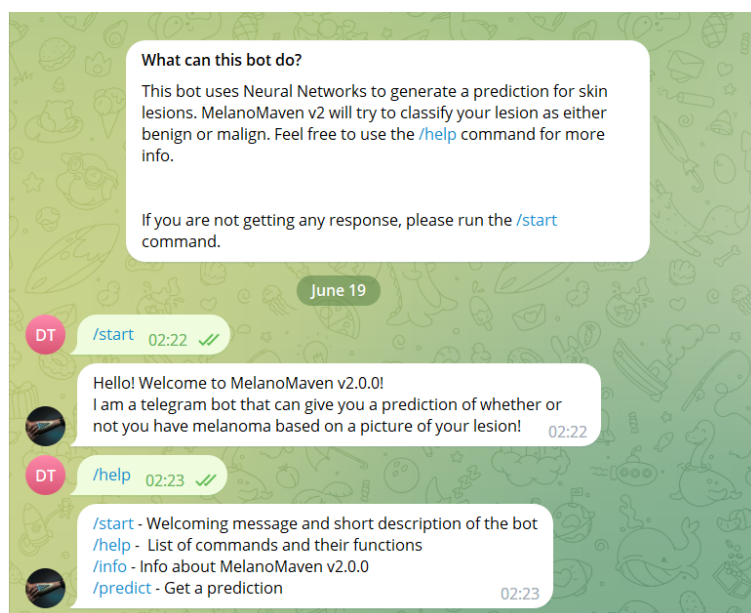
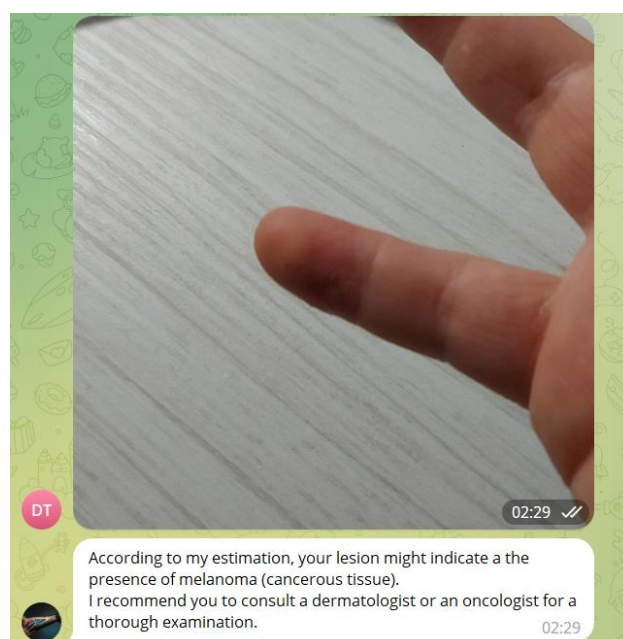
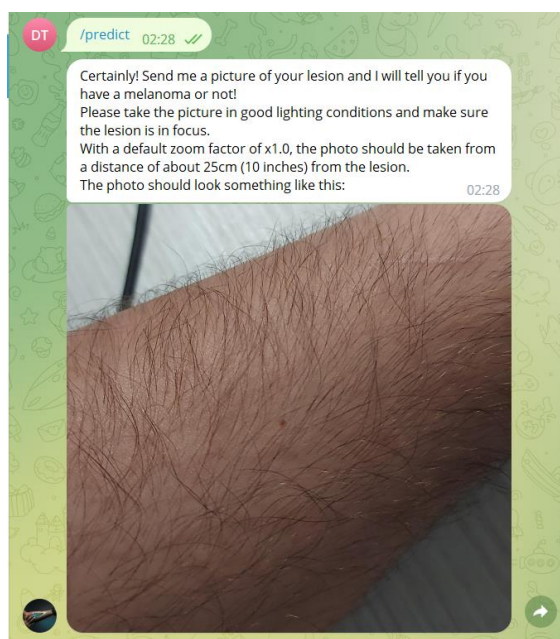


Figura 26: Comenzile de start și de ajutor ale chatbot-ului

Pentru solicitarea unei predicții, utilizatorul va folosi comanda */predict*. Dacă acesta nu se află în intervalul de pauză de 20 de secunde, va fi întâmpinat cu un mesaj menit să detalieze cum ar trebui efectuată fotografia, precum și un exemplu. După ce utilizatorul trimite imaginea de input, chatbot-ul va răspunde cu predicția asociată.



Figurile 27a și 27b: Utilizatorul a solicitat o evaluare a leziunii sale

Dacă utilizatorul trimite o poză fără să utilizeze comanda */predict*, acesta va fi informat de necesitatea acestui pas. Totodată, chatbot-ul va răspunde mereu cu același text la orice mesaj scris.

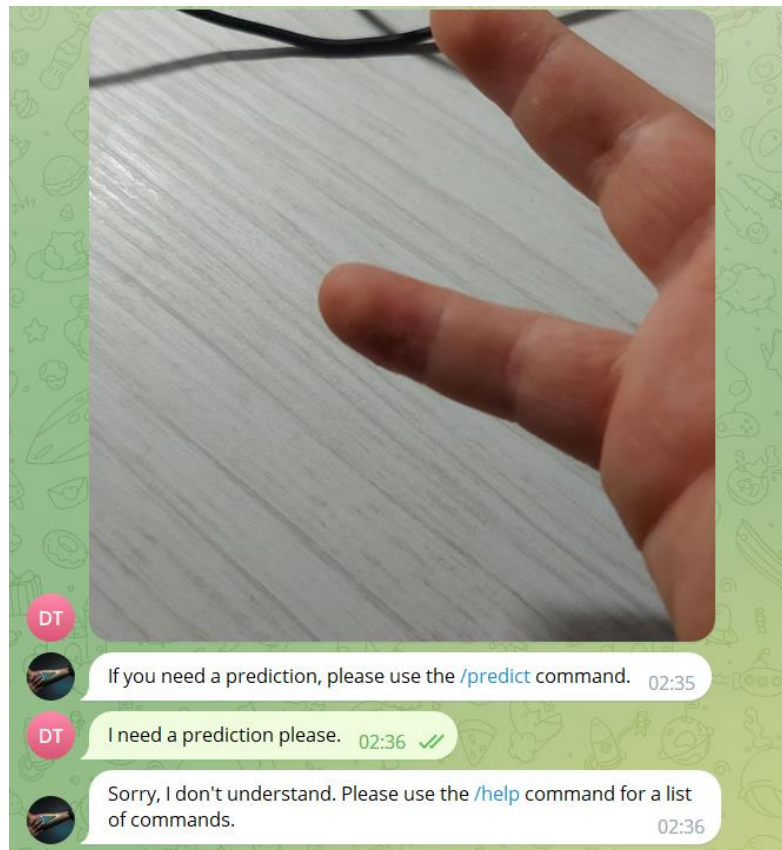
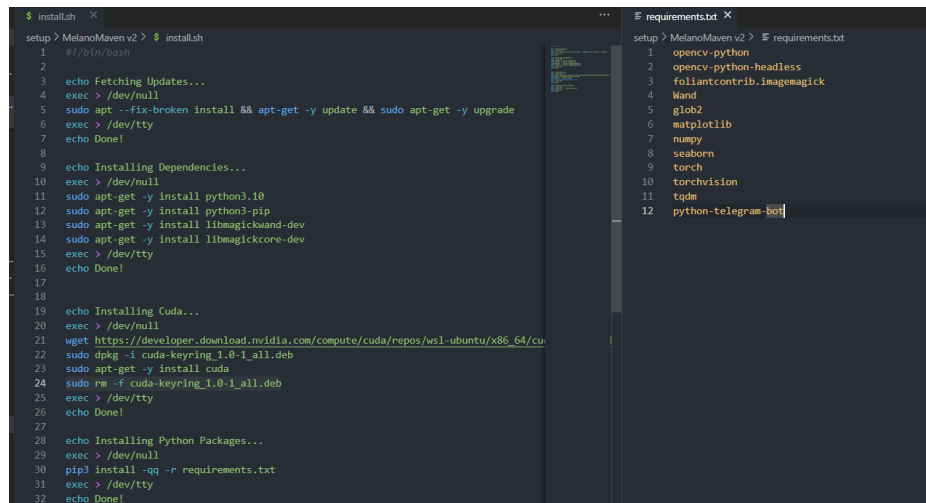


Figura 28: Utilizarea incorectă a chatbot-ului

5.2 Lansare proiect

Pentru funcționarea corectă a aplicației, este necesar ca un sistem care funcționează pe o distribuție de Linux – Ubuntu 22.04. Pentru pregătirea serverului, este suficientă rularea scriptului *install.sh*, aflat în directorul “setup/MelanoMaven v2/” cu drepturi de administrator. Acesta va instala toate utilitățile necesare și totodată va folosi fișierul *requirements.txt*, care conține toate pachetele de Python necesare pentru configurarea aplicației.

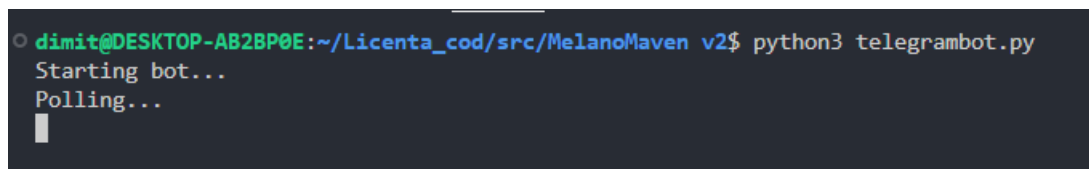


```
$ install.sh
setup > MelanoMaven v2 > $ install.sh
1 #!/bin/bash
2
3 echo Fetching Updates...
4 exec > /dev/null
5 sudo apt --fix-broken install && apt-get -y update && sudo apt-get -y upgrade
6 exec > /dev/tty
7 echo Done!
8
9 echo Installing Dependencies...
10 exec > /dev/null
11 sudo apt-get -y install python3.10
12 sudo apt-get -y install python3-pip
13 sudo apt-get -y install libmagickwand-dev
14 sudo apt-get -y install libmagickcore-dev
15 exec > /dev/tty
16 echo Done!
17
18 echo Installing Cuda...
19 exec > /dev/null
20
21 wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.0-1_all.deb
22 sudo dpkg -i cuda-keyring_1.0-1_all.deb
23 sudo apt-get -y install cuda
24 sudo rm -f cuda-keyring_1.0-1_all.deb
25 exec > /dev/tty
26 echo Done!
27
28 echo Installing Python Packages...
29 exec > /dev/null
30 pip3 install -q -r requirements.txt
31 exec > /dev/tty
32 echo Done!
```

```
requirements.txt
setup > MelanoMaven v2 > requirements.txt
1 opencv-python
2 opencv-python-headless
3 foliantcontrib.imagemagick
4 Wand
5 glob2
6 matplotlib
7 numpy
8 seaborn
9 torch
10 torchvision
11 tqdm
12 python-telegram-bot
```

Figura 29: Scriptul *setup.sh*

După finalizarea tuturor instalărilor, aplicația poate fi lansată din interiorul directorului “src/ MelanoMaven v2/”, utilizând scriptul *telegrambot.py*, apelând comanda *python3 telegrambot.py* (figura 27). Este important de menționat că fișierul *users.log* din directorul “resources/” trebuie curățat și resetat manual înainte de a lansa porni aplicația.



```
dimit@DESKTOP-AB2BP0E:~/Licenta_cod/src/MelanoMaven v2$ python3 telegrambot.py
Starting bot...
Polling...
█
```

Figura 30: Lansarea aplicației

6. Concluzii și contribuții

6.1 Concluzii

Scopul acestei lucrări a fost acela de a cerceta și de a propune o soluție ce folosește tehnici de învățare automată și rețele neuronale pentru a asista în imagistica medicală și pentru a stimula o creștere în număr a diagnosticelor corecte și precoce ale melanoamelor, precum și a obține o comparație între instrumentul obținut de mine și alte sisteme deja existente.

În primul capitol au fost menționate obiectivele lucrării și totodată oferită o descriere a domeniului. Au fost detaliate situația medicală a cancerelor de piele, importanța diagnosticării la timp a acestora, precum și metode clinice de clasificare și identificare a formațiunilor maligne. Am pus accent pe metodologia ABCD (*asymmetry, irregular border, inconsistent colour, diameter greater than 6 mm*), care poate fi implementată și în tehnicile de învățare profundă pentru a optimiza modelele utilizate.

În cel de-al 2-lea capitol a fost prezentată situația soluțiilor pre-existente în ceea ce privește automatizarea imagisticii medicale, prezentând și explicând diverse tehnici și metodologii din cadrul învățării automate și a Rețelelor Neuronale Artificiale (ANN), cu un interes deosebit asupra Rețelelor Neuronale Convoluționale (CNN), utilizate cu succes în vederea artificială, cu precădere pentru probleme care presupun procesare și identificarea anumitor caracteristici în aplicațiile multimedia. Totodată a fost efectuat un studiu de caz asupra a 2 instrumente deja existente pentru identificarea melanoamelor – SolarScan și MelaFind – având în vedere motivația cercetării acestora, procesul de achiziție de date și rezultatele obținute. Dintre acestea, MelaFind s-a dovedit a fi un succes, fiind utilizat în imagistica medicală până în 2017 [39]. În aceeași direcție, a fost elaborată o listă de cerințe și obiective ale aplicației dezvoltate

În cel de-al 3-lea capitol a fost elaborată o introducere detaliată a tehnologiilor utilizate. Au fost discutate modelarea sistemelor de inteligență artificială folosind rețele neuronale artificiale, precum și tehnici de optimizare ale acestor modele. S-a pus accent pe descrierea arhitecturală a celor 2 rețele utilizate în dezvoltarea aplicației, Inception V3 și VGG16.

În capitolul 4 a fost prezentată aplicația dezvoltată de mine. Au fost trecute în revistă arhitectura celor 2 modele propuse și a fost detaliat metodele prin care au fost implementate. În final au fost exemplificate rezultatele obținute de către MelanoMaven v1, respectiv MelanoMaven v2. Modelele dezvoltate în cadrul acestui proiect s-au dovedit a fi mai puțin performante decât soluția celor de la institutul Electro-Optical Sciences, MelaFind.

În cel de-al 5-lea capitol a fost realizat un studiu de caz asupra aplicației, cu detalii despre cum poate fi lansată și utilizată. Necesari pentru accesarea acesteia în calitate de utilizator este doar un cont valid pe serviciul de mesagerie Telegram. Din punct de vedere al

dezvoltării și lansării, MelanoMaven în forma sa de aplicație nu utilizează foarte multe resurse, fiind necesar pentru a rula doar un dispozitiv gazdă ce rulează pe un kernel de Linux și un spațiu de stocare redus (sub 300 Mb) pentru a salva modelul și scripturile necesare – pentru realizarea, antrenarea și testarea modelului este necesar mult mai mult spațiu de stocare, asociat seturilor de date pentru învățare, validare și testare.

Prin implementarea interfeței cu utilizatorul prin intermediul Telegram, aplicația ar putea fi extinsă pentru aplicabilitate în telemedicină. Lucrarea prezentată dorește să atragă atenția asupra problemelor medicale constituite de formele de cancer de piele și propune o soluție de diagnostic automat a melanoamelor – cea mai comună formă de leziune cutanată malignă.

6.2 Contribuții personale

- Sinteza de informații asupra soluțiilor similare, precum și a stadiului actual al tehnologiilor utilizate (toate sursele utilizate sunt menționate în secțiunea 7, bibliografie);
- Integrarea și aplicarea tehnologiilor studiate în dezvoltarea aplicației: utilizarea conceptelor de regularizare și dropout pentru a optimiza procesul de învățare, utilizarea arhitecturilor InceptionV3 și VGG16 prin intermediul a 2 librării utilizate cu precădere în domeniul inteligenței artificiale (Keras și Pytorch), precum și utilizarea Telegram BOT API pentru generarea și automatizarea unei interfețe cu utilizatorul;
- Dezvoltarea propriei aplicații, MelanoMaven v2

6.3 Dezvoltări ulterioare

MelanoMaven v2 poate fi îmbunătățit drastic printr-o serie de pași:

- Utilizarea unui set de date mai larg și mai echilibrat
- Se pot utiliza mai multe modele simultan pentru a căuta trăsături specifice: folosind metodologia ABCD, se pot antrena câte 4 modele diferite, fiecare cu scopul de a extrage altceva – asimetrie, margini, culoare sau diametru. Un al 5-lea model ar putea fi creat pentru a instanția pe celelalte 4 și pentru a genera un scor în baza trăsăturilor specifice extrase
- Se poate antrena un model separat care să segmenteze imaginile cu leziuni de piele de imagini oarecare, pentru a spori robustețea aplicației la abuzul chatbot-ului
- Se poate incorpora opțiunea de a înregistra istoricul pacientului

7. Bibliografie

- [1] A. Mahbod, G. Schaefer, C. Wang, R. Ecker and I. Ellinge, "Skin Lesion Classification Using Hybrid Deep Neural Networks," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 1229-1233, doi: 10.1109/ICASSP.2019.8683352.
- [2] Bre, Facundo & Gimenez, Juan & Fachinotti, Víctor. (2017). Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. *Energy and Buildings*. 158. 10.1016/j.enbuild.2017.11.045.
- [3] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.
- [4] Codella, Noel. "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)". arXiv: 1710.05006 [cs.CV]
- [5] E. Ayan and H. M. Ünver, "Diagnosis of Pneumonia from Chest X-Ray Images Using Deep Learning," 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), Istanbul, Turkey, 2019, pp. 1-5, doi: 10.1109/EBBT.2019.8741582.
- [6] E. Phaisangittisagul, "An Analysis of the Regularization Between L2 and Dropout in Single Hidden Layer Neural Network," 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Bangkok, Thailand, 2016, pp. 174-179, doi: 10.1109/ISMS.2016.14.
- [7] Esteva, Andre. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29. doi:10.1038/s41591-018-0316-z
- [8] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.
- [9] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.
- [10] G. J. van Wyk, "Evolutionary Neural Architecture Search for Image Restoration", IJCNN 2019, 2019, doi: 10.48550/arXiv.1812.05866
- [11] He, Kaiming "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

- [12] Hoshyar, Azadeh Noori. (2011). Review on automatic early skin cancer detection. 2011 International Conference on Computer Science and Service System (CSSS). doi:10.1109/csss.2011.5974581
- [13] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.
- [14] K. Simonyan, "Very Deep Convolutional Networks for Large-Scale Image Recognition" (2015), doi: 10.48550/arXiv.1409.1556
- [15] Kerekes, John. "Receiver Operating Characteristic Curve Confidence Intervals and Regions," in IEEE Geoscience and Remote Sensing Letters, vol. 5, no. 2, pp. 251-255, April 2008, doi: 10.1109/LGRS.2008.915928.
- [16] L. Liu, Y. Luo, X. Shen, M. Sun and B. Li, " β -Dropout: A Unified Dropout," in IEEE Access, vol. 7, pp. 36140-36153, 2019, doi: 10.1109/ACCESS.2019.2904881.
- [17] LaBerge, Gregory. "Recent Advances in Studies of Skin Color and Skin Cancer." The Yale journal of biology and medicine vol. 93,1 69-80. 27 Mar. 2020
- [18] Menzies, Scott. The Performance of SolarScan: An Automated Dermoscopy Image Analysis Instrument for the Diagnosis of Primary Melanoma. Arch Dermatol. 2005;141(11):1388–1396. doi:10.1001/archderm.141.11.1388
- [19] Monheit, Gary. (2011). The Performance of MelaFind. Archives of Dermatology, 147(2), 188. doi:10.1001/archdermatol.2010.302
- [20] Ramaneswaran, S. (2021). Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification. Computational and Mathematical Methods in Medicine. 2021. 1-10. 10.1155/2021/2577375.
- [21] Ren, Suping. "The impact of genomics in understanding human melanoma progression and metastasis." Cancer control : journal of the Moffitt Cancer Center vol. 15,3 (2008): 202-15. doi:10.1177/107327480801500303
- [22] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.
- [23] Szegedy, Christian. "Going Deeper with Convolutions", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2015, pp 1-9
- [24] Szegedy, Christian. "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.
- [25] V. Tiwari, C. Pandey, A. Dwivedi and V. Yadav, "Image Classification Using Deep Neural Network," 2020 2nd International Conference on Advances in Computing,

Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 730-733, doi: 10.1109/ICACCCN51052.2020.9362804.

[26] Vestergaard, Malene E. "Automated diagnostic instruments for cutaneous melanoma." *Seminars in cutaneous medicine and surgery* vol. 27,1 (2008): 32-6. doi:10.1016/j.sder.2008.01.001

[27] Wang, Bin. "Positive and negative max pooling for image classification," 2013 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2013, pp. 278-279, doi: 10.1109/ICCE.2013.6486894.

[28] Wang, Jiang. "CNN-RNN: A Unified Framework for Multi-label Image Classification," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2285-2294, doi: 10.1109/CVPR.2016.251.

[29] Yadav, Sanjay. "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, 2016, pp. 78-83, doi: 10.1109/IACC.2016.25.

[30] Z. Hu, J. Zhang and Y. Ge, "Handling Vanishing Gradient Problem Using Artificial Derivative," in *IEEE Access*, vol. 9, pp. 22371-22377, 2021, doi: 10.1109/ACCESS.2021.3054915.

[31] Zeeshan, Rabia. "Cancer metastasis - tricks of the trade." *Bosnian journal of basic medical sciences* vol. 17,3 172-182. 20 Aug. 2017, doi:10.17305/bjbms.2017.1908

[32] About ImageNet; <https://www.image-net.org/about.php>; accesat în iunie 2023

[33] Inception Module; <https://deeptai.org/machine-learning-glossary-and-terms/inception-module>; accesat în mai 2023

[34] Inception v3 Architecture;
<https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/>; accesat în mai 2023

[35] ISIC Datasets Archive;
https://huggingface.co/datasets/Dimit/MelanoMaven_ISIC_2017-2020/viewer/Dimit-MelanoMaven_ISIC_2017-2020; creat în iunie 2023

[36] Melanoma Survival;
<https://www.cancerresearchuk.org/about-cancer/melanoma/survival>; accesat în aprilie 2023

[37] Python-Telegram-Bot; <https://github.com/python-telegram-bot/python-telegram-bot>; accesat în iunie 2023

[38] What are Neural Networks?; <https://www.ibm.com/topics/neural-networks>; accesat la data de 18.05.2023

[39] <https://www.nsmedicaldevices.com/news/strata-skin-sciences-signs-licensing-deal-for-several-melafind-assets/>; accesat în mai 2023

[40] <https://openai.com/dall-e-2>; accesat în mai 2023