MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

Reducerea dimensionalității reprezentării vizuale a imaginilor OCT pentru analiza afecțiunilor oftalmologice folosind contrastive learning

LUCRARE DE LICENȚĂ

Absolvent: **Bianca VESA**

Coordonator științific: **Conf. dr. ing. Anca Nicoleta MĂRGINEAN**

2022

Cuprins

Capitolul 1. Introducere - Contextul proiectului

1.1. Contextul proiectului

Datorită avansului tehnologic ce a avut loc în domeniul inteligenței artificiale, se dorește din ce în ce mai mult aplicarea algoritmilor deep learning în cerințe care implică analiza imaginilor medicale. Aceste imagini provin din diferite ramuri ale medicinei cum ar fi: radiologia, patologia, dermatologia, oftalmologia etc. Cu ajutorul inteligenței artificiale, se utilizează diferite modele de învățare automată pentru a analiza datele medicale în scopul de a descoperi noi perspective care să ajute la diagnosticarea anumitor afecțiuni, tratarea și monitorizarea lor. În ceea ce privește domeniul oftalmologic, imaginile medicale sunt obținute cu ajutorul unei tehnici numită tomografie în coerență optică (*engl.* Optical Coherence Tomography - OCT).

OCT reprezintă o investigație modernă și non-invazivă, care utilizează lumina pentru a genera imagini în secțiune transversală ale țesuturilor organelor.[1] Această investigație este foarte utilă în diagnosticarea afecțiunilor organelor pentru care nu se poate efectua o biopsie, cum ar fi ochiul uman. OCT ajută medicii oftalmologi să analizeze fiecare strat distinct al retinei, care pot avea o grosime de doar 10 microni. Ei pot diagnostica astfel complet și corect diverse afecțiuni oftalmologice și le pot monitoriza evoluția. Măsurătorile pe care le pune la dispoziție OCT sunt esențiale în identificarea și urmărirea răspunsurilor la tratament pentru anumite boli serioase cum ar fi degenerescența maculară legată de vârstă (DMLV). Deoarece mai mult de 80% din tulburările de vedere pot fi prevenite odată ce sunt identificate la timp, este foarte important ca modificările retinei sa fie atent monitorizate.

DMLV reprezintă cea mai răspândită cauză a pierderii vederii, întâlnită la populația cu vârstă de peste 60 de ani. Ea este caracterizată de deteriorarea zonei centrale a retinei, numită macula. Această afecțiune prezintă două forme, una uscată și una umedă. Forma uscată este mai comună în rândul populației și cauzează ușoare pierderi de vedere. Pe de altă parte, forma umedă reprezintă stadiul avansat al formei uscate și afectează aproximativ 15% din pacienții care dezvoltă forma uscată. Această formă umedă se instalează repede și duce la pierderea permanentă a vederii. Așadar, perioada care precede instalarea formei umede este critică pentru administrarea tratamentului care încetinește pierderea vederii. La momentul actual nu există tratamente care să prevină apariția acestei afecțiuni sau care să o vindece. Forma uscată a DMLV prezintă acumulări de țesut, care poartă numele de drusen, iar forma umedă este asociată cu vase de sânge ce cresc dedesubtul maculei și care se pot sparge, eliminând lichide și provocând răni (neovascularizare coroidiană). Aceste caracteristici ale afecțiunii apar în imaginile obținute cu OCT, ceea ce face ca depistarea lor cu ajutorul algoritmilor deep learning să fie o cerință utilă în monitorizarea pacienților și administrarea unui tratament adecvat.

Deep learning este o ramură a machine learning, care include metode de învățare automată bazate pe rețele neuronale artificiale. Aceste rețele imită modul în care creierul uman învață informații noi, având la dispoziție date din mediul înconjurător.

Algoritmii deep learning sunt utili în analiza datelor de tip imagine, deoarce rețelele neuronale sunt alcătuite din straturi interconectate de "neuroni", care extrag progresiv caracteristicile cele mai importante prezente în imagini și detalii semnificative pe care ochiul uman nu le poate identifica. Datorită acestor abilități, rețelele neuronale se dovedesc a fi de ajutor în analiza imaginilor medicale pentru a stabili un diagnostic și pentru a monitoriza stadiul unei afecțiuni. Sursele publice de imagini OCT, cum este setul de date Kermany menționat in articolul [2], prezintă retine cu anomalii specifice DMLV, adică neovascularizare coroidiană (*engl.* choroidal neovascularization - CNV) și drusen, sau alte afecțiuni precum edem macular diabetic (*engl.* diabetic macular edema - DME). Figura 1.1 prezintă niște exemple de anomalii din setul de date Kermany.

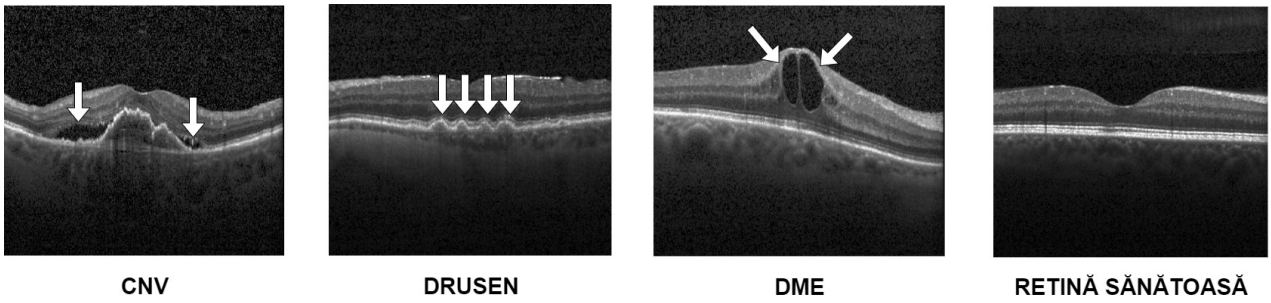

Figura 1.1: Anomalii ale retinei

În modelele clasice deep learning, care implică o arhitectură de tip rețea neuronală, este nevoie de imagini adnotate pentru ca modelul să identifice cât mai clar apartenența fiecărei imagini la clasa corespunzătoare (ex. clasele din setul de date Kermany - CNV, DME, DRUSEN, NORMAL). Modelul pornește de la imaginile de antrenament și încearcă să categorisească fiecare instanță în funcție de clasele existente în setul de date. În realitate acest scenariu este rareori întâlnit, deoarece în urma unei investigații medicale se poate genera un număr mare de imagini care necesită un efort adițional din partea medicilor să fie adnotate, având și un număr mare de pacienți, care la rândul lor pot prezenta multiple afecțiuni oftalmologice. De asemenea procesul de analiză a fiecărei imagini și asigurarea calității ei implică un timp suplimentar pe care specialiștii ar trebui să îl dedice. Un alt aspect care îngreunează acest scenariu este faptul că pentru a putea fi făcute disponibile publicului larg, imaginile trebuie să treacă printr-un proces de anonimizare, în scopul menținerii confidențialității pacienților. Din aceste motive, abordările de învățare nesupervizată (care nu necesită adnotarea imaginilor) sunt din ce în ce mai utilizate în domeniul deep learning.

Prin învățarea nesupervizată, algoritmii învață o reprezentare redusă a imaginii inițiale, care surprinde trăsăturile cele mai importante, fără a utiliza adnotări. Astfel, algoritmul va pune accent pe informațiile prezente în imagini, fără a fi influențat de clasa în care ele se încadrează.

Datorită acestor aspecte caracteristice învățării nesupervizate, abordarea devine de un real ajutor în sfera imaginilor medicale, deoarece sunt des întâlnite cazurile în care o persoană suferă de mai multe afecțiuni, care nu pot fi încadrate într-o singură categorie. De asemenea, cantitatea mare de imagini obținute în urma unei consultații medicale poate fi de ajutor în procesul de învățare prin comparație.

Învățarea nesupervizată cuprinde tipologii de algoritmi, printre care se numără algoritmii de clustering, dar și rețele neuronale de tip autoencoder, care au oferit rezultate bune în ceea ce privește utilizarea lor în studiul anomaliilor din domeniul medical.

O tehnică de învățare nesupervizată care a început recent să aducă rezultate impresionante este contrastive learning, în care reprezentările imaginilor sub forma vectorilor de trăsături sunt învățate prin minimizarea diferenței dintre două reprezentări ale aceleiași imagini și maximizarea diferenței dintre două reprezentări ale unor imagini diferite. Aceste reprezentări se obțin prin operații de augmentare, care alterează imaginea inițială aplicând diverse transformări la nivelul pixelilor (ex. rotații, decupări, zoom, adăugare de noise, modificări de culoare etc.). Efectuând aceste maximizări si minimizări, modelul reușește să identifice informații specifice fiecărei instanțe din setul de antrenare, prin "contrast" cu restul imaginilor. Abordarea descrisă se încadrează la învățarea self-supervised, deoarece se presupune că nu avem la dispoziție adnotări. În cazul în care imaginile sunt adnotate, se poate trece la învățare supervizată, considerând ca fiind similare imaginile care aparțin aceleiași clase, și diferite, imagini din clase distincte.

Această lucrare își propune să evidențieze utilizarea imaginilor OCT, aplicând contrastive learning supervised și self-supervised, pentru a analiza anumite afecțiuni oftalmologice, a îmbunătăți urmărirea evoluției DMLV și a ajuta medicii oftalmologi în monitorizarea afecțiunii și administrarea tratamentului.

Capitolul 2. Obiectivele Proiectului

2.1. Motivația proiectului

În ultimii ani a apărut un real interes în ceea ce privește utilizarea inteligenței artificiale în domeniul medical. Cele mai des întâlnite utilizări sunt în diagnosticarea pacienților cu diferite afecțiuni interpretând date medicale, studiul și dezvoltarea noilor medicamente și tratamente, îmbunătățirea comunicării dintre doctor și pacient și aplicarea tratamentelor la distanță. În computer vision au apărut rezultate impresionante în ceea ce privește procesarea și analiza imaginilor medicale.

Articolul [3] cuprinde o trecere în revistă a celor mai actuale metode dezvoltate în domeniul computer vision, care utilizează imagini medicale. Printre aplicațiile menționate este și [4], în care autorii prezintă mai multe abordări bazate pe deep learning și rețele neuronale convoluționale, care ajută la detecția afecțiunii COVID-19, pornind de la un set de date care conține radiografii pulmonare. Problema definită este una de clasificare, deoarece se împart instanțele din setul de date în două categorii: sănătos sau bolnav de COVID-19. Cu ajutorul deep learning se pot rezolva și probleme care implică segmentarea semantică a imaginilor, așa cum se prezintă în articolul [5], care are ca scop segmentarea tumorilor pe creier din imagini obținute prin rezonanță magnetică (RMN). O altă utilizare este detecția anumitor trăsături sau obiecte în imagini. Un astfel de exemplu [6] reprezintă detecția leziunilor la nivelul mamografiilor, utilizând arhitecturi precum Faster R-CNN.

Având aceste exemple de utilizări ale deep learning în domeniul medical, care au adus rezultate bune ce vin în ajutorul medicilor, această lucrare urmărește aplicarea unor arhitecturi de rețele neuronale pentru a obține rezultate la fel de bune în contextul monitorizării evoluției DMLV. Arhitectura va avea la bază conceptul contrastive learning.

Contrastive learning este folosit pentru a învăța caracteristicile generale ale unui set de date neetichetat, ajutând modelul să distingă datele care sunt similare și cele care sunt diferite. Astfel, se identifică trăsăturile de nivel înalt înainte de a trece la un task precum clasificare sau segmentare. Ceea ce este cel mai util în cadrul contrastive learning este faptul că nu este nevoie ca imaginile să fie etichetate pentru abordarea self-supervised. Așa cum se detaliază în articolul [7], se pornește de la o imagine numită ancoră, cu ajutorul căreia se generează o versiune augmentată a imaginii, aplicând transformări la nivelul pixelilor. Contrastive loss va face ca această pereche ancoră-versiune augmentată să se "atragă", reprezentând aceeași imagine, iar perechile ancoră-altă imagine din același batch să se "respingă", deoarece ilustrează imagini diferite. Un exemplu se poate observa în figura 2.1.

În cazul în care în setul de date există și imagini adnotate, contrastive loss se poate calcula luând în considerare și clasele în care se încadrează imaginile. Astfel, perechile vor fi formate din ancoră-imagine din același batch aparținând aceleiași clase, care se vor "atrage" și ancoră-imagine din același batch aparținând unor clase diferite, care se vor "respinge". Acest exemplu este ilustrat în figura 2.2.

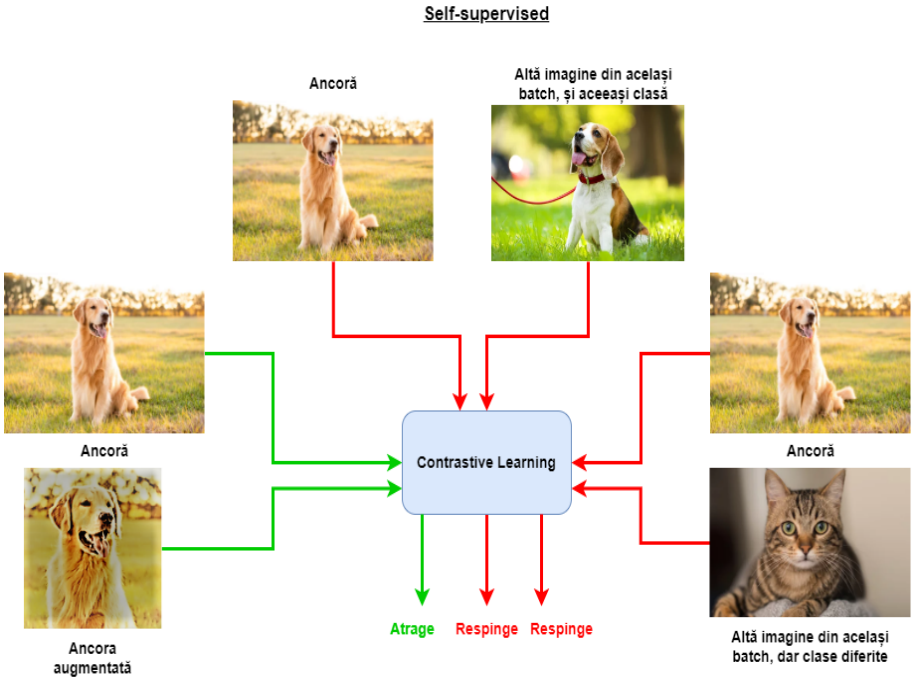

Figura 2.1: Contrastive self-supervised

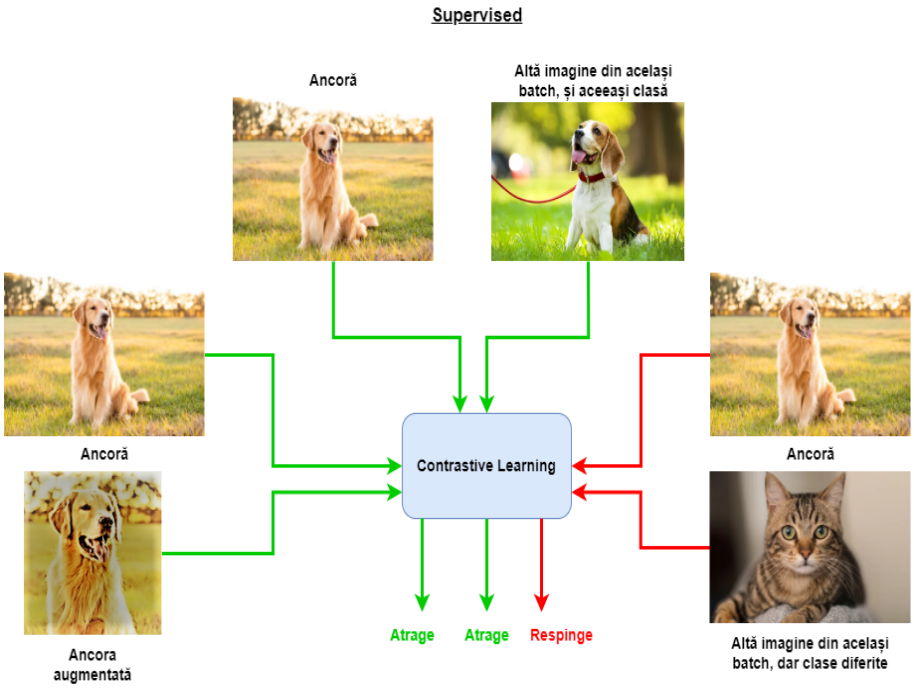

Figura 2.2: Contrastive supervised

Articolul [8] prezintă arhitectura Contrastive Multi-Task Convolutional Neural Network (CMT-CNN), care utilizează contrastive loss pentru a clasifica un set de date format din imagini CT și cu raze X la plămâni și a ajuta la diagnosticarea afecțiunii COVID-19. După cum menționează autorii, CMT-CNN a obținut rezultate care le depășesc pe cele ale modelelor existente deja în literatură, ceea ce poate fi un bun indicator pentru utilizarea contrastive learning în sfera imaginilor medicale.

Analiza evoluției afecțiunilor oftamologice precum DMLV se poate realiza cu ajutorul unui model care să urmărească evoluția acuității vizuale asociate unui set de imagini OCT. Acuitatea vizuală reprezintă capacitatea ochiului uman de a distinge de la o anumită distanță detalii precum forme sau litere. Aceasta este măsurată utilizând un panou standard cu litere de diferite dimensiuni, cu ajutorul căruia este testată claritatea vederii unui individ. În mod obișnuit, în timpul examinării panoul se află la o distanță de 6 metri de pacient (20 feet în sistem imperial), iar valoarea măsurată indică distanța de la care pacientul nu mai reușește să identifice literele (în literatura americană acuitatea maximă este referită ca 20/20). Această tehnică a fost introdusă de oftalmologul olandez Herman Snellen, motiv pentru care panoul folosit la examinare se numește panou Snellen [9].

În analizele efectuate de oftalmologi s-a observat că există o strânsă legătură între valorile obținute în urma examinărilor acuității Snellen și rezultatele procedurii OCT. Astfel, în această lucrare vor fi utilizate atât setul de date public cu imagini OCT Kermany, cât și un set de imagini de la pacienți monitorizați la Spitalul Clinic Județean de Urgență din Cluj-Napoca, care beneficiază de un set de valori măsurate ale acuităților vizuale pentru 94 de pacienți. La nivel de pacient sunt asociate mai multe vizite, care pot fi la distanțe de câteva luni sau chiar ani. În cadrul unei vizite s-a realizat un OCT și s-a măsurat acuitatea vizuală. Aparatul care realizează imaginile OCT generează un set de mai multe slice-uri în secțiune transversală a retinei, generând un volum de imagini numite B-scans. Un volum de B-scans din setul de date poate fi de tip fast, dense sau p. pole, care conțin 25, 49, respectiv 61 de scanări.

2.2. Obiective

Analiza imaginilor de tip OCT de către medicii oftalmologi implică un grad de subiectivitate, mai ales în ceea ce privește degenerescența maculară legată de vârstă, care este o afecțiune pentru care până în ziua de azi nu s-a descoperit un tratament definitoriu, care să prevină sau să înlăture efectele, ci doar măsuri care încetinesc evoluția și procesul de pierdere a vederii. Din acest motiv, lucrarea detaliază un sistem care să descopere corelații între imagini OCT și acuitatea vizuală măsurată, având la bază abordarea contrastive learning. Acest sistem va putea fi folosit ca bază pentru dezvoltarea ulterioară a unei arhitecturi care să urmărească evoluția în timp a acuității vizuale, bazată pe imagini OCT, care ar ajuta medicii oftalmologi la administrarea timpurie a tratamentului pacienților cu risc ridicat, pentru a preveni conversia DMLV forma uscată în DMLV forma umedă.

Obiectivele principale sunt:

1. Extragerea informațiilor importante din imaginile OCT sub forma vectorilor de trăsături
2. Utilizarea prin transfer a trăsăturilor extrase ca informații generale pentru realizarea unui task de clasificare
3. Analiza reprezentărilor vizuale obținute din aceste informații extrase

4. Aplicarea modelului care extrage trăsături la nivelul unui volum de B-scans, pentru a genera un set de date nou care conține reprezentări compacte ale volumelor de imagini
5. Utilizarea setului de date format din volume de imagini pentru a realiza inferența acuității vizuale corespondente, ca modalitate de a evalua calitatea trăsăturilor extrase

Capitolul 3. Studiu Bibliografic

3.1. OCT și DMLV

Degenerescența maculară legată de vârstă este una din principalele cauze ale pierderii vederii în rândul populației de peste 60 de ani și reprezintă aproximativ 8.7% din cazurile de orbire la nivel global. DMLV prezintă mai multe stadii, de la incipient, la intermediar, până la avansat. În cazul formei avansate a DMLV, cea umedă sau neovasculară, creșterea vaselor de sânge (neovascularizare coroidiană) poate duce la deteriorarea ireversibilă a fotoreceptorilor (celulele care sunt responsabile de interceptarea luminii la nivelul retinei) și pierderea rapidă a vederii. În figura 3.1 se pot observa la nivelul imaginilor de tip fund de ochi și OCT, caracteristicile etapelor DMLV. În prezent, pacienții pot trece de la forma uscată a DMLV la cea umedă fără a prezenta simptome sau orice alte schimbări ușor de detectat și măsurat. Pentru a monitoriza progresia DMLV se analizează imaginile obținute prin tehnica OCT și se urmăresc schimbări în ceea ce privește acuitatea vizuală asociată setului de imagini, măsurată cu ajutorul unui panou Snellen. În cazul ideal aceste investigații ar trebui realizate la intervale constante și frecvente de timp, însă acest lucru implică anumite costuri și eforturi crescute de timp din partea pacienților și a medicilor.

Din aceste motive este de o importanță crescută să se identifice pacienții care prezintă cel mai mare risc de conversie la DMLV umedă, pentru a permite intervenția înainte de deteriorarea permanentă a vederii. Intervenția presupune injectarea intraoculară a unor substanțe care ajută la încetinirea evoluției.

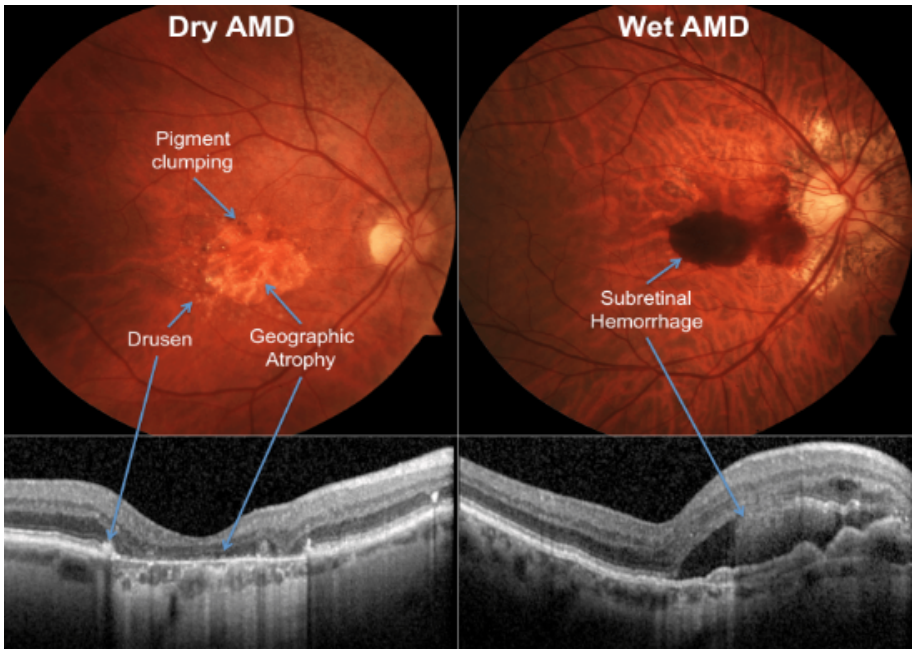

Figura 3.1: În stânga DMLV uscată, în dreapta DMLV umedă; sursa [10]

3.2. Predicția acuității vizuale

Predicția acuității vizuale este un pas important în dezvoltarea unui sistem care să analizeze evoluția DMLV, deoarece s-a demonstrat impactul direct al acestei afecțiuni asupra capacităților vizuale ale omului. Experții au încercat până în prezent diverse abordări pentru a rezolva această problemă, iar unele dintre ele ies în evidență. Una dintre ele este lucrarea scrisă de *Kawczynski et al.* [11], care detaliază o rețea neuronală ce prezice cea mai bună acuitate vizuală corectată, pornind de la imagini OCT care prezintă DMLV. S-a realizat atât clasificarea, cât și regresia pentru valori ale acuității care corespund valorilor Snellen: $<20/40$, $<20/60$, $\leq 20/200$. Clasificarea a oferit o valoare AUC = 0.84, pentru predicția acuității la 12 luni de la cea măsurată. Regresia a rezultat într-o valoare root mean squared error de 0.33.

Aslam et al. [12] au devoltat un model capabil de a prezice valoarea acuității vizuale, utilizând imagini OCT. Modelul de tip regresie a fost antrenat folosind 1210 de imagini. Root mean squared error a arătat o diferență de 8.2 litere între acuitatea prezisă și cea reală, rezultând un coeficient de regresie de 0.85. Ei au demonstrat astfel legătura strânsă care există între modificările ce apar în valorile acuității vizuale și informațiile surprinse de imaginile OCT.

O altă arhitectură care tratează problema predicției este cea din articolul [13], care evidențiază un model ce reușește să prezică valorile câmpului vizual, folosind imagini OCT. S-a folosit de asemenea root mean squared error ca metrică de evaluare a rezultatelor, obținând rezultate remarcabile, care pot fi de ajutor medicilor specialiști în examinarea pacienților.

Lucrarea [14] scoate în evidență o abordare bazată pe extragerea trăsăturilor folosind arhitectura autoencoder. Modelul encoder încapsulat în această arhitectură a fost antrenat pentru a reduce dimensionalitatea imaginilor OCT, folosind ca tehnică de optimizare, eroarea reconstrucției imaginii. Modelul a oferit rezultate promițătoare în integrarea ulterioară într-un sistem care să prezică acuitatea vizuală din imaginile provenite de la Spitalul Clinic Județean de Urgență din Cluj-Napoca.

De asemenea, sistemul a fost extins, oferind posibilitatea de a analiza predicția acuității vizuale viitoare, bazată pe valoarea acuității vizuale curente. Acest aspect este foarte util în ceea ce privește monitorizarea evoluției DMLV, deoarece ia în considerare identificarea pacienților care se află într-un risc crescut de a dezvolta o formă severă a afecțiunii, care să ducă la pierderea vederii.

3.3. Contrastive Learning

Scopul contrastive learning este de a învăța reprezentări de dimensionalitate redusă ale datelor de antrenare în așa fel încât perechi de date care conțin informații similare să rămână apropiate în spațiul reprezentărilor, iar perechile de date diferite să fie îndepărtate.

Contrastive learning poate fi aplicat atât în scenariul supervizat cât și în cel nesupervizat. În contextul nesupervizat, contrastive learning devine una din cele mai puternice abordări din self-supervised learning.

Astfel, s-a demonstrat că o arhitectură nesupervizată bazată pe un encoder și un clasificator liniar, numită SimCLR (*engl.* Simple framework for Contrastive Learning visual Representations), a depășit performanța unui model supervizat pre-antrenat pe setul de date ImageNet, format imagini naturale.

Acest aspect se poate observa in figura 3.2, extrasă din articolul [15].

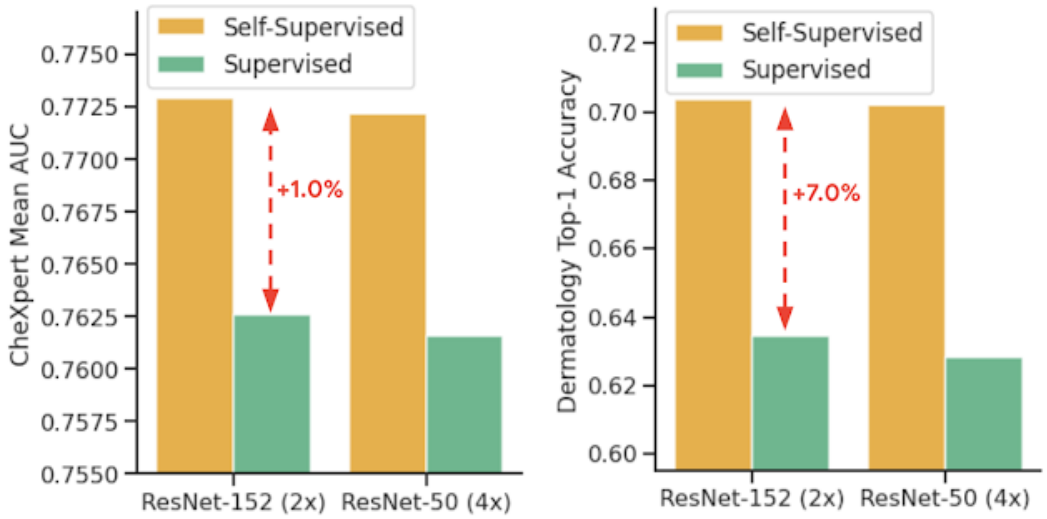

Figura 3.2: Comparație între performanțele obținute de două modele, unul supervised antrenat cu funcția de loss cross-entropy și unul self-supervised antrenat cu funcția de loss contrastive și fine-tuning în stil supervised, pe două seturi de date (unul cu radiografii pulmonare și unul cu imagini cu afecțiuni de piele)

Arhitectura ResNet a fost pre-antrenată pe setul de date ImageNet, folosind abordarea supervised. Pe de altă parte, aceeași arhitectură a fost la bază pre-antrenată în modalitatea self-supervised, folosind contrastive learning pe imagini medicale. Ambele modele au fost mai departe utilizate prin transfer în contextul imaginilor medicale (radiografii pulmonare și afecțiuni ale pielii). În acest fel, modelul pre-antrenat în stil self-supervised depășește semnificativ perfomanța celui supervised.

În plus, în articolul [16], *Khosla et al.* au dus mai departe această metodă și au demonstrat faptul că aplicarea contrastive learning într-un scenariu supervizat aduce un plus de performanță. Ei au propus modificarea funcției de loss de tip contrastive, astfel încât să utilizeze etichetele pe care unele seturi de date le pun la dispoziție. Având aceste etichete, modelul va încerca să ”apropie” imaginile care aparțin aceleiași clase și să ”îndepărteze” imaginile din același batch care aparțin unor clase diferite. Astfel, puterea metodei contrastive learning crește, deoarece modelul reușește să pună în contrast imaginile cu ajutorul cărora este antrenat și extrage trăsături mult mai relevante domeniului problemei.

3.3.1. **SimCLR**

Paradigma transfer learning este tot mai des utilizată în contextul modelelor specializate pe imagini medicale. Adoptând această abordare, se antrenează o arhitectură care folosește un set de date adnotate ce conține un număr mare de imagini, cum este setul ImageNet. Ulterior, se utilizează informațiile generalizate care au fost învățate, într-un task specific domeniului problemei care se dorește a fi rezolvată (de exemplu o problemă de clasificare pe imagini medicale). De la această idee au pornit *Chen et al.* [7] pentru a dezvolta arhitectura SimCLR.

Cel mai cunoscut loss function în domeniul rețelelor neuronale este cross-entropy. Cross-entropy reprezintă o metrică utilizată în probleme de clasificare pentru a monitoriza erorile de învățare ale rețelei. Ea măsoară diferența dintre două sau mai multe distribuții de probabilitate, mai exact între valoarea prezisă de model și cea considerată adevărată. În acest mod, este folosită la optimizarea modelului, asigurând valorile potrivite pentru predicție, astfel încât acestea să fie cât mai apropiate de valorile adevărate. Cu toate acestea, cross-entropy prezintă următoarele dezavantaje: este sensibilă la noise, penalizează semnificativ datele de test care diferă puțin față de datele de antrenare și are o capacitate scăzută de generalizare.

Din acest motiv, SimCLR folosește contrastive loss ca metrică de optimizare. Având un sample din setul de antrenare, și un set de operații de augmentare, se generează două exemple de transformări ale aceleiași imagini. Aceste transformări sunt trecute printr-un encoder pentru a se obține o reprezentare compactă a informației, apoi sunt trecute printr-un projection head, care mapează reprezentările la vectori de trăsături. În final vectorii care reprezintă la bază aceeași imagine sunt aduși mai aproape unul de celălalt în spațiul trăsăturilor. Logica enunțată este ilustrată în figura 3.3.

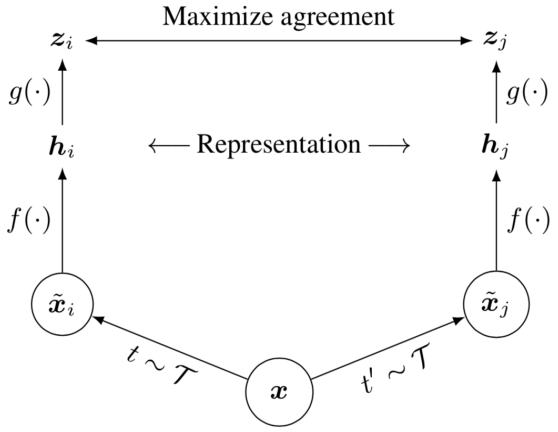

Figura 3.3: Logica SimCLR, sursa [7]

Un alt framework propus în articolul [15] este MICLe, care implică o logică asemănătoare SimCLR, însă folosește mai multe instanțe aparținând aceleiași clase, dacă există etichete în setul de date.

3.4. Extragerea trăsăturilor

Extragerea trăsăturilor din imaginile de antrenare permite modelului să învețe o reprezentare redusă și generalizată a informației specifice domeniului problemei. Pentru a efectua această operație se utilizează o arhitectură de rețea neuronală convoluțională, denumită în literatură encoder. Un encoder poate fi interpretat ca o funcție $h = f(x)$, care mapează o imagine x (în contextul SimCLR o transformare a unei imagini) la un vector de trăsături în spațiul reprezentărilor.

Ideea de bază pornește de la modelul autoencoder, care este format din trei componente: encoder, cod și decoder. Codul este o reprezentare compactă a informației primită ca input la nivelul encoder-ului. Decoder-ul are ca scop reconstrucția informației comprimată sub forma codului. Modelul este reprezentat în figura 3.4

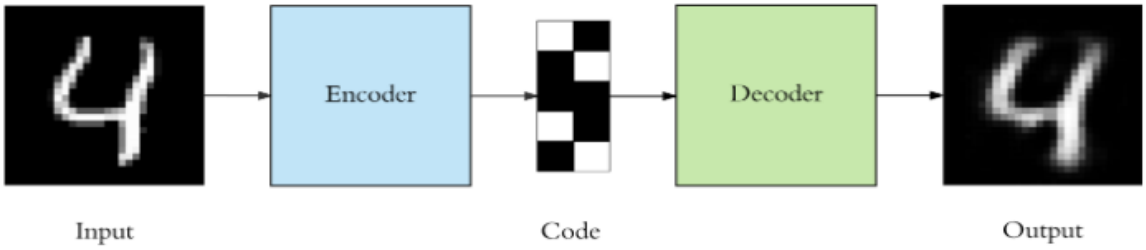

Figura 3.4: Modelul autoencoder, sursa [17]

Avantajele de care beneficiază acest model sunt: faptul că reușește să cuprindă informațiile cele mai importante care sunt specifice domeniului problemei, că este antrenat în stil self-supervised, ceea ce oferă flexibilitate crescută și nu în ultimul rând, faptul că rezultatul final nu este unul identic cu informația primită ca input, ci este predispus unui anumit loss, lucru care ajută la generalizarea datelor învățate.

3.4.1. Encoder

Encoder-ul este o rețea neuronală convoluțională (*engl.* convolutional neural network) formată din una sau mai multe niveluri (*engl.* layers). Arhitectura este una flexibilă, permițând de la un layer până la un model mai complex, cum sunt cele de ultimă oră utilizate în literatură (ex. ResNet, DenseNet, VGG), iar dimensiunea codului este la rândul ei variabilă. *Berchuck et al.* [18] au demonstrat utilizarea unui deep variational autoencoder în predicția ratei de progresie a câmpului vizual în cazul afecțiunii oculare numită glaucom. În această arhitectură se utilizează un encoder care are ca rezultat distribuția de probabilitate a fiecărei trăsături în spațiul reprezentărilor. Cu această abordare au îmbunătățit procesul de predicție a ratei de progresie a glaucomului, având în plus beneficiul de a identifica tipare în degradarea câmpului vizual. De asemenea, *Waldstein et al.* au dezvoltat în [19] o arhitectură deep learning unsupervised, care a reușit să identifice un dicționar de 20 de biomarkeri caracteristici retinei, pe baza analizei unui set de imagini OCT volumetrice provenite de la pacienți diagnosticați cu DMLV. Acești biomarkeri au fost validați prin corelare cu alte aspecte clinice cum este acuitatea vizuală sau morfologia retinei și pot fi folosiți mai departe ca features în alte probleme de clasificare sau regresie.

Articolul detaliază două tipuri de encoder utilizate. Primul encoder transpune fiecare slice de tip B-scan OCT într-un vector de trăsături, urmând ca cel de-al doilea encoder să utilizeze informația în formă compactă întregului B-scan pentru a extrage cele 20 de trăsături, considerate a fi biomarkeri. Arhitectura a reușit să extragă cu o acuratețe ridicată trăsături deja cunoscute în practica medicală, adăugând un set de noi informații relevante în imaginile OCT.

3.5. Transfer Learning

Transfer learning definește o tehnică des utilizată în domeniul deep learning, care presupune utilizarea modelelor deja antrenate pe un set de date, pentru a rezolva o problemă dintr-un domeniu similar. În urma antrenării pe un set de date de dimensiuni mari cum este ImageNet se obțin niște valori pentru weights, care reprezintă informația învățată. Aceste weights se pot utiliza ca punct de pornire pentru antrenarea aceluiași model, dar pe un alt set de date sau a unui model nou care să includă modelul inițial. Astfel, informația învățată este reutilizată, timpul de antrenare este redus, iar erorile produse de model scad. De asemenea, tehnica este de folos în cazul în care setul nou de date disponibil este de dimensiuni reduse, deoarece modelul a acumulat anterior informații dintr-un set mare de imagini. Logica metodei transfer learning este evidențiată în figura 3.5.

ImageNet este un set de imagini adnotate, conceput pentru a fi utilizat în probleme de cercetare în domeniul computer vision. Este format din aproximativ 14 milioane de imagini grupate în aproximativ 21000 de clase și reprezintă setul de date standard utilizat în probleme care implică algoritmi computer vision, fiind des folosit ca set de antrenare pentru a compara performanța modelelor incluse prin transfer.

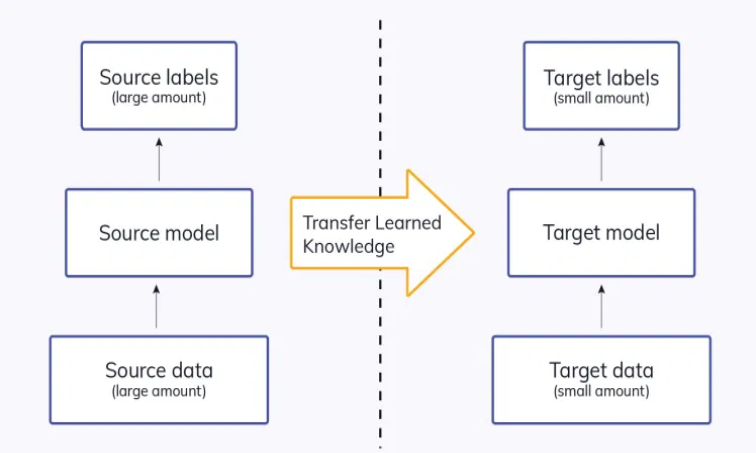

Figura 3.5: Transfer learning, sursa [20]

În cele ce urmează se află o trecere în revistă a câtorva dintre cele mai utilizate modele în tehnica transfer learning:

- **Inception**
  Arhitectura Inception include mai multe componente, denumite module Inception, care sunt de dimensiuni mici și au fost realizate pentru a reduce costurile puterii de calcul necesare în cadrul antrenării unui deep neural network.

Un modul Inception aplică mai multe filtre de convoluție asupra imaginii primită ca input, ceea ce îl face util în probleme de extragere a trăsăturilor. Cei care au dezvoltat arhitectura au demonstrat performanța sa ridicată și costurile reduse de putere de calcul. Modelul a fost utilizat ca backbone prin transfer în arhitectura dezvoltată de *Park et al.* în articolul [13], pentru a prezice valorile câmpului vizual din imagini OCT. Modelului i s-a adăugat un global average pooling layer, urmat de patru dense layers. De asemenea, *Yan et al.* [21] au utilizat Inception în realizarea unui sistem care a fost antrenat să prezică riscul evoluției DMLV utilizând imagini de tip fund de ochi împreună cu informații referitoare la variantele genetice ale DMLV.

- **ResNet**
  Rețeaua ResNet se bazează pe mai multe module denumite ResNet blocks, care includ conexiuni numite skip connections. Aceste skip connections permit ignorarea anumitor blocuri din arhitectură, aspect care a fost implementat pentru a elimina problema *vanishing gradient* - în procesul de back-propagation realizat de rețea, numeroasele operații de înmulțire care au loc pot face ca valoarea gradientului să fie infinitezimală, ceea ce duce la degradarea semnificativă a performanței. *Khosla et al.* [16] au utilizat prin transfer arhitectura ResNet ca modul de extragere a trăsăturilor pentru a evidenția performanța supervised contrastive learning. În articol autorii menționează faptul că modelul a obținut rezultate impresionante în urma antrenării pe setul ImageNet, depășind cu 0.8% performanța celui mai bun model bazat pe această arhitectură la momentul respectiv.

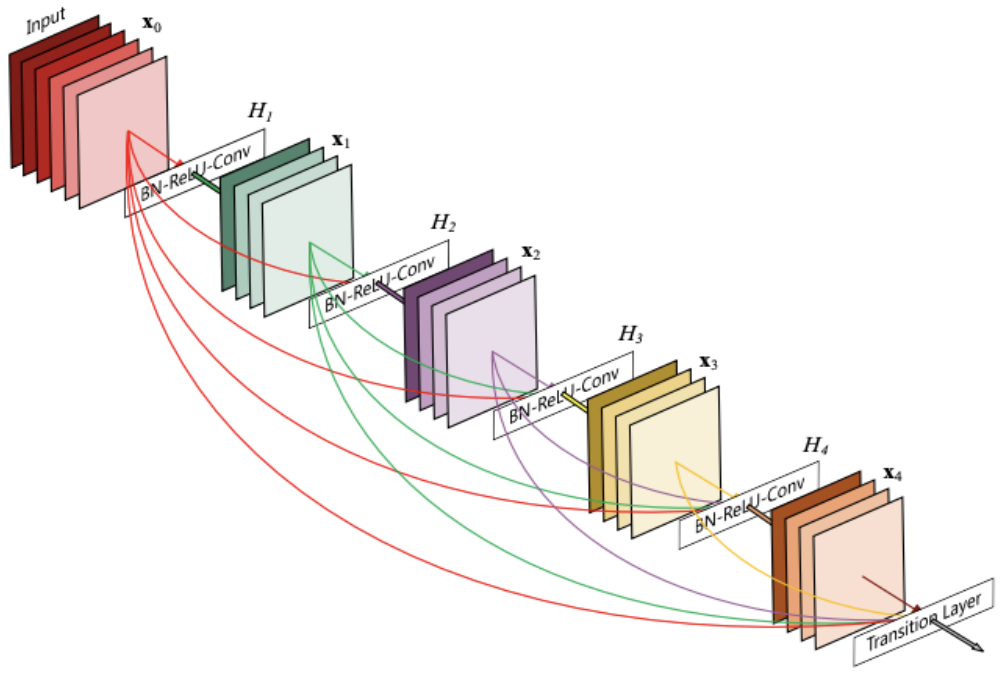

Figura 3.6: Arhitectura DenseNet, sursa [22]

- **DenseNet**
  Ideea dezvoltării modelului DenseNet a pornit de la dorința de a îmbunătăți arhitecturile complexe existente, care s-a dovedit că includeau blocuri redundante. Ideea de bază se aseamănă cu modelul ResNet, însă include câteva beneficii cheie. Toate blocurile de convoluție sunt interconectate, ceea ce semnifică faptul că fiecare layer primește ca input rezultatele tuturor blocurilor anterioare și transferă mai departe informația tuturor blocurilor succesoare.
  În acest mod, ultimul layer are acces la informațiile extrase de orice alt layer, inclusiv primul, aspect care dorește să elimine problema redundanței. Având la dispoziție această interconectare a blocurilor, modelul devine mai compact, iar numărul de parametri scade, ușurând procesul de antrenare. Autorii acestei arhitecturi au înregistrat cea mai bună acuratețe la antrenarea pe setul de date ImageNet, comparativ cu alte modele standard existente în literatură și care au fost la rândul lor antrenate pe acest set [22]. De asemenea, unele experimente au arătat că DenseNet oferă o reprezentare foarte bună a trăsăturilor imaginilor, motiv pentru care a fost utilizată în dezvoltarea arhitecturii detaliate ulterior în această lucrare. Arhitectura DenseNet este ilustrată în figura 3.6

3.6. Rețele neuronale convoluționale

Luând în considerare faptul că un encoder este la bază o simplă rețea neuronală, rezultatele oferite se pot utiliza folosind transfer learning în probleme de clasificare sau regresie. În articolul [23], *Burlina et al.* ilustrează dezvoltarea unei rețele care a fost antrenată pentru a identifica stadiul evoluției DMLV, utilizând imagini tip fund de ochi. Soluția propusă este una de clasificare în două categorii: imagini care nu prezintă DMLV sau prezintă stadiu incipient DMLV, respectiv imagini cu stadiu intermediar și avansat DMLV. Acuratețea obținută variază între 88.4% și 91.6%, fiind apropiată de performanța unui expert uman, conform testelor efectuate.

O altă soluție identificată de *Russakoff et al.* [24] este arhitectura AMDnet, utilizată în predicția evoluției DMLV. S-au utilizat imagini OCT provenite de la 71 de pacienți diagnosticați cu stadiu incipient sau intermediar DMLV. Acești pacienți au fost monitorizați într-un interval de doi ani, colectând imagini OCT la câte un an distanță, urmând să fie grupați în două categorii: cei care nu au fost diagnosticați cu stadiu avansat în cel de-al doilea an, respectiv cei care au trecut la forma avansată după cel de-al doilea an. Modelul a fost antrenat să clasifice date în cele două categorii, depășind performanța modelului din literatură VGG16, în același context. De asemenea, articolul demonstrează faptul că preprocesarea imaginilor pentru a evidenția segmentarea semantică a straturilor retinei ajută semnificativ procesul de antrenare și implicit performanța rezultată.

Încă o abordare care a oferit rezultate apropiate cu cele obținute de expertul uman este cea propusă de *Venhuizen et al.* [25], în care imaginile OCT au fost clasificate în cinci categorii corespunzătoare stadiilor DMLV: fără DMLV, stadiu incipient, stadiu intermediar, stadiu avansat cu atrofie geografică, stadiu avansat cu neovascularizare coroidiană.

În articolul [26] se prezintă clasificarea riscului regresiei la nivel de drusen (depozite de lipide care se acumulează pe retină), aspect care nu s-a reușit a fi realizat în prezent folosind metode convenționale. Aceste depozite sunt direct corelate cu evoluția DMLV. Pacienții au fost monitorizați la perioade de trei luni, într-un interval de 12 luni până la 60 de luni. Metrica urmărită de model este hazard ratio (HR), fiind asociată riscului regresiei la nivel de drusen.

Astfel o valoare HR<1 semnifică risc scăzut, în timp ce o valoare HR>1 este reprezentativă pentru riscul crescut. Metoda oferă un avans în ceea ce privește predicția evoluției DMLV, care să ajute medicii oftalmologi să atribuie tratamente adecvate persoanelor aflate în risc crescut.

Capitolul 4. Analiză și Fundamentare Teoretică

4.1. Analiza problemei

În capitolul anterior am detaliat anumite soluții din literatură care au adus rezultate impresionante în ceea ce privește problema clasificării, în special a imaginilor OCT în funcție de stadiul evoluției DMLV. Într-un scenariu ideal, arhitecturile de rețele neuronale convoluționale de ultimă oră ar trebui să fie capabile să rezolve această sarcină, deoarece au dat dovadă de performanță ridicată în multe contexte în care au fost antrenate folosind diverse tipuri de imagini medicale, ba chiar imagini de tip OCT.

Un astfel de experiment a fost condus în stadiul incipient al dezvoltării acestei lucrări, și anume clasificarea setului de date Kermany în cele patru clase pe care le include. Folosind arhitectura DenseNet121 prin transfer, modelul a atins acuratețea de 96% după doar 20 de epoci de antrenare folosind cross-entropy ca funcție de loss.

Cu toate acestea, în lumea reală există puține scenarii în care setul de date este prelucrat la fel de bine precum setul Kermany. Problema pe care această lucrare încearcă să o rezolve este analiza imaginilor OCT la nivel de volum, aspect pe care setul Kermany nu îl include, fiind format doar din slice-uri independente. Acuitatea vizuală măsurată de către medici cu ajutorul metodei Snellen este asociată întregului volum B-scan. Acest volum cuprinde imagini în secțiune transversală a întregii suprafețe a retinei, iar în urma procedurii pot rezulta 25, 49 sau chiar 61 de imagini. Printre acestea se pot regăsi porțiuni ale retinei care să nu prezinte nicio afecțiune, lucru care poate îngreuna procesul de învățare al unui model de tip rețea neuronală.

O altă problemă care intervine este faptul că măsurarea acuității vizuale reprezintă un proces subiectiv, unde pot să intervină anumite erori. Spre exemplu, anumiți pacienți care suferă de DMLV formă uscată pot să nu prezinte nicio deficiență de vedere, însă imaginile OCT să evidențieze această afecțiune. De asemenea, există cazuri în care în urma începerii tratamentului injectabil pentru încetinirea evoluției DMLV, unii pacienți prezintă o îmbunătățire a acuității vizuale măsurate, iar în altele nu.

Setul de date de la Spitalul Clinic Județean de Urgență din Cluj-Napoca este un astfel de caz, care reflectă un scenariu real și care include atât volume de imagini OCT pentru fiecare vizită a fiecărui pacient, cât și un document care cuprinde valori pentru acuitatea vizuală asociată acestor volume. Setul înglobează astfel de date pentru 94 de pacienți care au fost consultați. Aspectele care îngreunează lucrul cu setul de date amintit anterior sunt:

- Pentru unii pacienți există valoarea acuității măsurată la o anumită vizită, însă setul nu conține imaginile OCT corespondente acelei vizite
- Pentru unii pacienți există seturi de imagini OCT aferente unei vizite însă tabelul nu include acuitatea vizuală măsurată
- Valorile acuității nu sunt într-un format consistent (ex. unele valori sunt scrise în format Snellen imperial, altele în format Snellen metric, iar altele în format zecimal), motiv pentru care necesită procesare suplimentară pentru a aduce toate valorile în același format

- Acuitatea vizuală nu a fost măsurată în același mod pentru toți pacienții (ex. pentru unii a fost măsurată cu lentilă de probă, pentru unii a fost măsurată cu ochelarii pacientului, iar pentru alții s-a măsurat cu o lentilă specială care îmbunătățește acuitatea vizuală)
- Anumiți pacienți au avut intervenții medicale pentru complicații ale DMLV neovascular, motiv pentru care aceștia au fost excluși din studiu
- Acuitatea vizuală nu a fost măsurată la același interval de timp pentru toate vizitele, existând diferențe de cateva luni sau chiar ani, motiv pentru care valorile acuității vizuale pot fi susceptibile anumitor erori
- Anumitor pacienți li s-a administrat tratament pentru încetinirea evoluției DMLV, în timp ce altora nu
- Pot exista anumite fluctuații în valorile acuității vizuale măsurate de la o vizită la alta, în special în cazul pacienților cărăra li s-a administrat tratamentul injectabil

Cu toate acestea, lucrarea prezentă detaliază un sistem care include procesarea datelor din setul DMLV și utilizarea lor în analiza la nivel de volum a imaginilor OCT. Problema nu este una ușor de rezolvat, datorită aspectelor menționate anterior, motiv pentru care o abordare obișnuită de tip clasificare nu va aduce rezultate utile. Setul cuprinde aproximativ 21226 de imagini OCT individuale, iar grupate la nivel de volum alcătuiesc aproximativ 551 de volume. Deși din perspectivă medicală acest număr este mare, în cazul învățării supervizate nu este suficient, din pricina nevoii modelelor deep learning de a analiza o cantitate mare de date.

Pornind de la abordarea clasică de tip transfer a unui model pentru predicția acuității, au fost analizate mai departe avantajele pe care metodologia contrastive learning le-a adus până în prezent în domeniul imaginilor medicale. S-au condus niște experimente cu arhitectura contrastive SimCLR, care urmează învățarea în stil self-supervised.

Au fost identificate anumite rezultate în urma acestor experimente, însă nu suficient de bune pentru a surprinde cele mai importante trăsături ale imaginilor OCT, deoarece modelul nu surprindea diferențele dintre afecțiunile oftalmologice prezente. Mai apoi, bazat pe informațiile prezente în articolul [16], s-a trecut la antrenarea în stil contrastive supervised, folosind de setul de date Kermany, care este adnotat, la nivel B-scan în 4 clase aferente afecșiunilor prezente. Această abordare s-a dovedit a aduce o performanță mult mai ridicată, comparativ cu antrenarea în stil self-supervised, lucru care s-a observat și în capacitățile encoder-ului inclus în arhitectură de a reprezenta imaginile OCT.

Provocarea pe care o prezintă această metodă este lucrul cu volume de imagini. Din acest motiv, s-au folosit reprezentările pe care encoder-ul antrenat în stil supervised le poate da pentru imaginile din setul DMLV. Dintr-un volum de 25 de imagini OCT au rezultat 25 de vectori de trăsături de dimensiune 128 (dimensiunea ultimului layer al encoder-ului), care au fost concatenați pentru a asocia acuității vizuale măsurate o singură instanță OCT. După obținerea noilor reprezentări ale volumelor imaginilor, au fost investigate rezultatele pe care un model de clasificare și unul de regresie le oferă în ceea ce privește acuitatea vizuală asociată imaginilor OCT. O reprezentare vizuală a procesului de concatenare se poate observa în figura 4.1. Abordarea menționată va fi detaliată în cele ce urmează din punct de vedere teoretic.

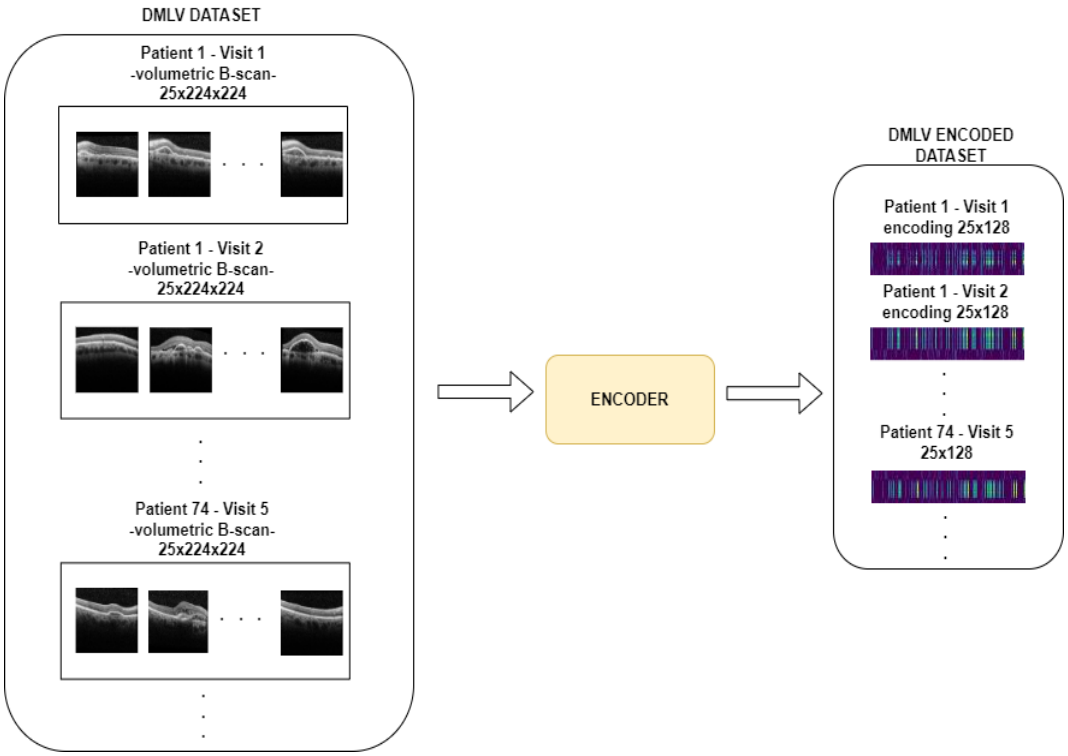

Figura 4.1: Transformarea setului de date DMLV în set de encoding-uri la nivel de volume B-scan

4.2. Deep Learning

Datorită avansului tehnologic în ceea ce privește puterea computațională a calculatoarelor, se acordă din ce în ce mai multă atenție modelelor deep learning, care au dovedit performanțe impresionante în probleme de analiză a imaginilor. Unul dintre avantajele pe care deep learning le oferă este faptul că elimină o parte din nevoia de pre-procesare a datelor. Rețelele neuronale reușesc să identifice informații în datele de intrare, chiar și în cazul datelor neprocesate și nestructurate, deoarece un layer îndeplinește și scopul de a extrage trăsăturile din imagini sau texte, lucru care în sine reprezintă procesare a datelor.

O arhitectură este compusă din mai multe layere formate din noduri, fiecare layer fiind conectat la altele succesoare. Prin procesul numit back-propagation, se folosesc algoritmi precum gradient descent pentru a calcula erori de predicție și a ajusta weight-urile și bias-urile corespunzătoare funcțiilor din fiecare nod și a antrena modelul să învețe informații. În ultimii ani, sistemele medicale din lume au beneficiat de capabilitățile de care au dat dovadă modelele deep learning, în special în aplicații care implică recunoașterea imaginilor, pentru a veni în sprijinul medicilor specialiști. Cu toate acestea, dezavantajul major pe care îl aduc este nevoia de cantități mari de date în procesul de învățare pentru a obține performanță ridicată. În realitate există puține seturi de date calitative, necesare arhitecturilor complexe, care să fie disponibile publicului larg și să fie adnotate. Din acest motiv, multe soluții existente se bazează pe abordările unsupervised și semi-supervised learning.

4.2.1. **Supervised learning**

Supervised learning presupune existența adnotărilor sau a etichetelor în setul de date de antrenare. Un model care se ghidează după acest mod de funcționare învață din setul de date etichetat și apoi este utilizat pentru a face predicții, folosind aspectele recurente din cunoștințele acumulate. Intrarea modelului este un set adnotat, iar rezultatul oferit este o funcție, care poate deduce informații din observații pe care nu le-a întâlnit în etapa de antrenare, prin procesul de inferență. Algoritmul poate compara rezultatul obținut (de exemplu eticheta corespunzătoare instanței de test), cu rezultatul considerat a fi corect (ground truth). Astfel poate identifica erori și poate ajusta parametrii învățați (back-propagation).

Problemele clasice din ramura supervised sunt cele de clasificare și regresie. Prima semnifică faptul că etichetele rezultate sunt niște categorii (de exemplu bolnav/sănătos), iar în cazul celei de-a doua, rezultatul obținut este o valoare reală și continuă (de exemplu prețul unei case). În practică, până în ziua de astăzi, majoritatea soluțiilor deep learning care au adus rezultate bune se bazează pe abordarea supervised learning, însă există modele unsupervised și semi-supervised care au oferit rezultate impresionante și se dorește în continuare dezvoltarea lor, din pricina deficitului de seturi de date adnotate și calitative.

4.2.2. **Unsupervised learning**

În cazul unsupervised learning, informațiile cunoscute sunt doar datele de intrare, fără a avea la dispoziție etichetele acestora și fără a genera o valoare rezultat care să reprezinte maparea datelor de intrare. În această metodă de învățare nu există supervizarea agentului uman, care să ofere modelului informații referitoare la ce anume este reprezentat în datele de intrare. În acest caz, rețeaua are ca scop identificarea similarităților existente în instanțele de antrenare, utilizând o anumită metrică ce va servi ca indice de similaritate. Metrica poate fi reprezentată de distanța dintre două instanțe de antrenare în spațiul reprezentărilor. Valoarea distanței poate fi mai mică sau mai mare, respectiv datele pot fi similare sau diferite.

Algoritmii clasici din sfera unsupervised sunt cei de clustering, de reducere a dimensionalității și de segmentare. Două exemple de metode des utilizate sunt K-means clustering, care grupează datele de intrare în k grupuri sau clustere, bazat pe caracteristici comune și Principal Component Analysis, care are ca scop reducerea dimensionalității, mapând reprezentările datelor de intrare la doar k trăsături. Aceste două metode sunt foarte utile, deoarece oferă avantajul de a extrage informațiile cheie din setul de date de antrenament, fără a fi influențat de conceptul care este reprezentat în fiecare exemplu dat ca intrare.

O ramură cuprinsă în sfera unsupervised este self-supervised learning, care de asemenea implică lipsa etichetelor. Cu toate acestea, un model self-supervised poate fi utilizat în probleme de clasificare. Arhitectura își monitorizează perfomanța calculând o valoare pentru loss. Valoarea poate fi caculată utilizând funcții precum contrastive loss function. Aceste metode self-supervised reprezintă un avantaj pentru a rezolva probleme de clasificare fără a avea nevoie de etichete.

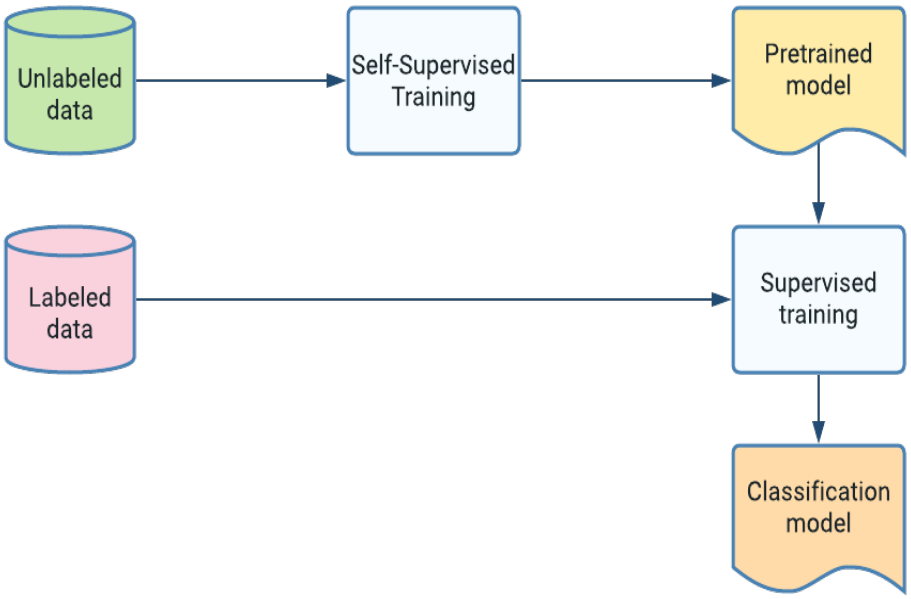

Figura 4.2: Semi-supervised learning, sursa [27]

4.2.3. **Semi-supervised learning**

Semi-supervised learning îmbină caracteristici ale supervised și self-supervised, utilizând seturi de date formate atât din instanțe etichetate cât și neetichetate. Arhitectura este formată dintr-un modul care este preantrenat pe datele neadnotate în stil self-supervised, după care este trecut prin procesul de fine-tuning, fiind antrenat în stil supervised cu ajutorul datelor etichetate. Etapa de fine-tuning îmbunătățește performanța, deoarece modelul este capabil să extraga informația generalizată din setul de date în urma pre-antrenării. Arhitecturile semi-supervised au dat dovadă de acuratețe foarte bună în probleme de clasificare, logica generală fiind ilustrată în figura 4.2.

4.3. Rețele Neuronale Convoluționale

Rețelele neuronale convoluționale (*engl.* Convolutional Neural Networks - CNN) reprezintă algoritmii deep learning cei mai utilizați, în special în contextul analizei imaginilor. Un CNN primește ca intrare o imagine, din care extrage anumite informații caracteristice și ajustează cunoștințele, acumulate sub forma unor parametri numiți weights și biases, cu scopul de a descoperi anumite tipare și a realiza predicții. Avantajul principal al acestor rețele, în comparație cu rețelele neuronale de tip feed-forward este dat de utilizarea a ceea ce se numește convolutional layer.

Un convolutional layer aplică procesul de convoluție asupra imaginilor pentru a extrage trăsăturile cele mai relevante . Operația de convoluție se realizează cu ajutorul unui kernel, care acționează ca un filtru asupra imaginii și efectuează o operație matematică (o serie de adunări și înmulțiri) asupra valorilor pixelilor dintr-o fereastră de o anumită dimensiune. Se parcurge întreaga imagine cu ajutorul acestui kernel și se calculează valorile corespunzătoare fiecărei ferestre de pixeli.

Valorile rezultate ajută la identificarea unor aspecte din imagini, pe care ochiul uman nu le observă, de exemplu muchii și contururi fine. Aplicarea repetată a mai multor tipuri de astfel de filtre duce la obținerea unei hărți de trăsături sau feature map. Harta de trăsături reduce dimensionalitatea problemei, aducând imaginea la o formă mai compactă care cuprinde informațiile cheie pe care rețeaua trebuie să le învețe. Totodată, harta păstrează informațiile și dependințele spațiale în urma convoluției. Procesul de aplicare a unui kernel se poate observa în figura 4.3.

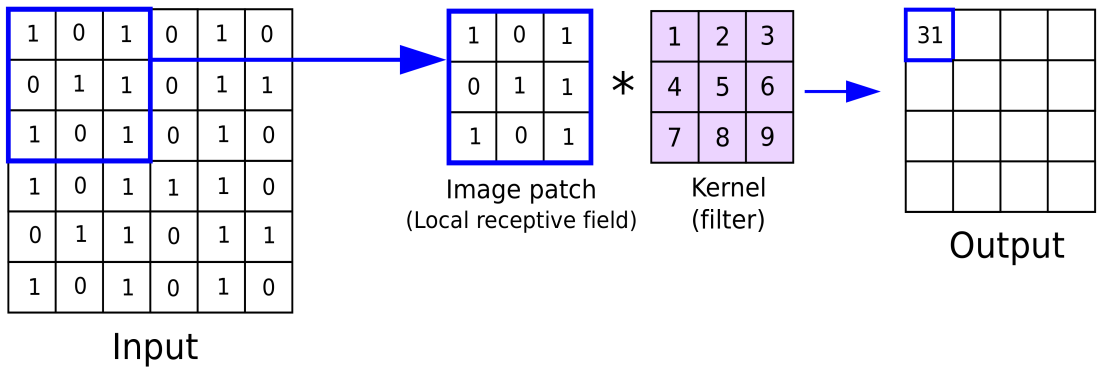

Figura 4.3: Convolutional layer, sursa [28]

În urma procesului de convoluție rețeaua aplică operația de pooling, pentru a reduce dimensiunea datelor și implicit nevoia de putere de calcul necesară pentru a le procesa. În plus, în urma operației se extrag trăsăturile dominante, care sunt invariante din punct de vedere al locației în imagine. Tipurile de pooling care se efectuează în practică sunt: max pooling și average pooling.

Procesul presupune, la fel ca în cazut convoluției, utilizarea unui kernel cu care se parcurge imaginea dată ca intrare. Din interiorul ferestrei de pixeli corespunzătoare kernel-ului se alege valoarea maximă (în cazul max pooling) sau se calculează media aritmetică a valorilor pixelilor (pentru average pooling). Efectuarea etapei de pooling este evidențiată în figura 4.4. Max pooling este utilizat mai des în practică, deoarece elimină pixeli de tip zgomot sau noise, ceea ce s-a demonstrat că îmbunătățește performanța. Perechile layer de convoluție-layer de pooling sunt adăugate de mai multe ori în arhitectură, pentru a extrage cât mai multe informații și a identifica detalii de nivel redus.

După aplicarea operațiilor repetate de convoluție și pooling, modelul a reușit să extragă trăsăturile cheie prezente în imagini. Observațiile colectate sunt aduse într-o formă mai compactă, cum este un vector de o singură dimensiune, sunt înmulțite cu valorile weight-urilor și însumate cu valorile bias-urilor, iar apoi sunt trecute prin layere de tip dense sau fully-connected, care aplică funcțiile de activare, pentru a începe procesul de învățare. Funcțiile de activare sunt necesare, deoarece dorim ca modelul să reușească să învețe informații complexe din datele de antrenare, ceea ce nu se poate întâmpla efectuând doar operații de înmulțire și adunare, care reprezintă operații liniare.

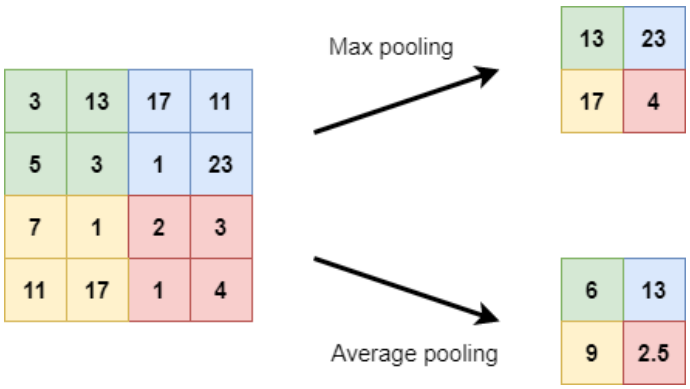

Figura 4.4: Tipuri de pooling

Un layer aflat între cel de intrare și cel de ieșire se numește hidden layer. Acesta nu primește în mod direct datele de intrare și nu calculează rezultatul final al modelului. Cea mai des utilizată funcție de activare pentru nodurile dintr-un hidden layer este Rectified Linear Unit Function, sau pe scurt ReLU. Este o funcție ușor de implementat (formula este evidențiată la 4.1, iar reprezentarea grafică se poate observa la 4.5), care s-a demonstrat că oferă performanță ridicată la antrenare și reduce semnificativ problema vanishing gradient, amintită în secțiunea 3.5.

$$f(x) = max(0, x) \tag{4.1}$$

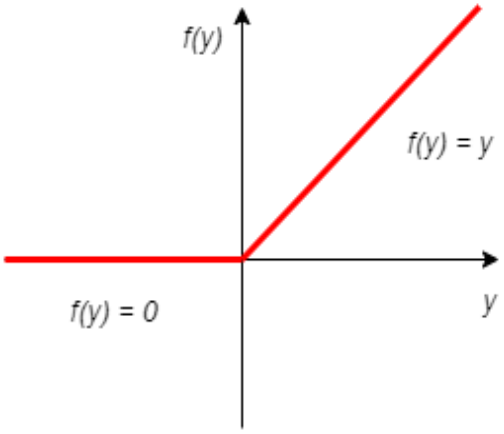

Figura 4.5: Rectified Linear Unit

Alte funcții de activare utilizate în practică sunt: Sigmoid (formula 4.2) și Tangentă hiperbolică, sau Tan-h (formula 4.3). Sigmoid oferă o reprezentare normalizată a rezultatelor, aducându-le în același interval, ceea ce o face utilă în probleme de clasificare, însă este foarte susceptibilă la problema vanishing gradient și necesită mai multă putere de calcul, comparativ cu ReLU. O alternativă la Sigmoid este Tan-h, care oferă o reprezentare mai bună pentru valori negative sau apropiate de 0, date la intrare. Cu toate acestea, Tan-h este la fel de predispusă problemei vanishing gradient.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

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

În ceea ce privește funcția de activare pentru layer-ul de ieșire, cele mai des utilizate sunt Sigmoid și Softmax, care au ca rezultat probabilitatea ca informația dată ca input să aparțină unei anumite clase. Sigmoid este utilizată în cazul clasificării binare, adică în doar două clase, iar Softmax (formula 4.4) în contextul predicției cu mai mult de două clase. Pentru probleme de regresie se preferă utilizarea unei funcții liniare.

$$f(x_i) = \frac{e^{x}}{\sum_{i=1}^{M} e^{x}}, \text{ unde M este numărul de clase} \tag{4.4}$$

4.3.1. **Funcții de loss**

Un aspect important în ceea ce privește procesul de învățare al unui algoritm deep learning este funcția de loss sau loss function. Ea reprezintă o metrică de evaluare a performanței modelului. Analizând valorile funcției de loss putem observa cât de bine reușește algoritmul să modeleze datele de intrare, bazat pe cunoștințele acumulate. Cu cât valoarea funcției este mai mică, cu atât mai aproape de lumea reală vor fi predicțiile modelului. Pe parcursul învățării, cu ajutorul unor funcții de optimizare, funcția de loss reușește să reducă erorile de predicție. Există numeroase tipuri de funcții de loss, care se folosesc în practică, depinzând de problema care se dorește a fi rezolvată. În continuare se află o enumerare a tipurilor de funcții de loss utilizate în această lucrare.

1. **Cross-entropy** este funcția de loss utilizată cel mai des în probleme de clasificare. Ea măsoară performanța modelului, al cărui rezultat este o valoare de probabilitate, între 0 și 1, ca instanța dată la intrare să aparțină unei anumite clase. Formula se regăsește la 4.5. Poate fi de următoarele trei tipuri:
   - **Binary cross-entropy** - pentru clasificare în două clase
   - **Sparse cross-entropy** - pentru clasificare în două sau mai multe clase
   - **Categorical cross-entropy** - pentru probleme în care etichetele sunt de tip categoric și trebuie trecute prin one-hot encoding, de exemplu pentru trei clase: [0, 0, 1], [0, 1, 0], [1, 0, 0]

   $$L_{CE} = \sum_{i=1}^{M} t_i log(s_i) \tag{4.5}$$

   unde M este numărul de clase, t_i este eticheta reală, iar s_i este probabilitatea Softmax a clasei i
2. **Contrastive loss**
   Dezavantajele principale identificate în cazul cross-entropy sunt: sensibilitatea la existența zgomotelor în imagini și o separare puternică a imaginilor învățate, fapt care generează penalizări semnificative pentru datele care diferă puțin față de datele din restul setului de antrenare.

Din acest motiv, *Chopra et al.* [29] au propus contrastive loss ca metodă de a învăța reprezentări ale datelor astfel: pornind de la exemplele de intrare, se extrag cu ajutorul unui encoder reprezentările compacte ale imaginilor și se ajustează parametrii arhitecturii astfel încât reprezentările imaginilor similare să fie cât mai apropiate în spațiul trăsăturilor, iar cele diferite să fie cât mai depărtate. Această logică este ilustrată în figura 4.6.

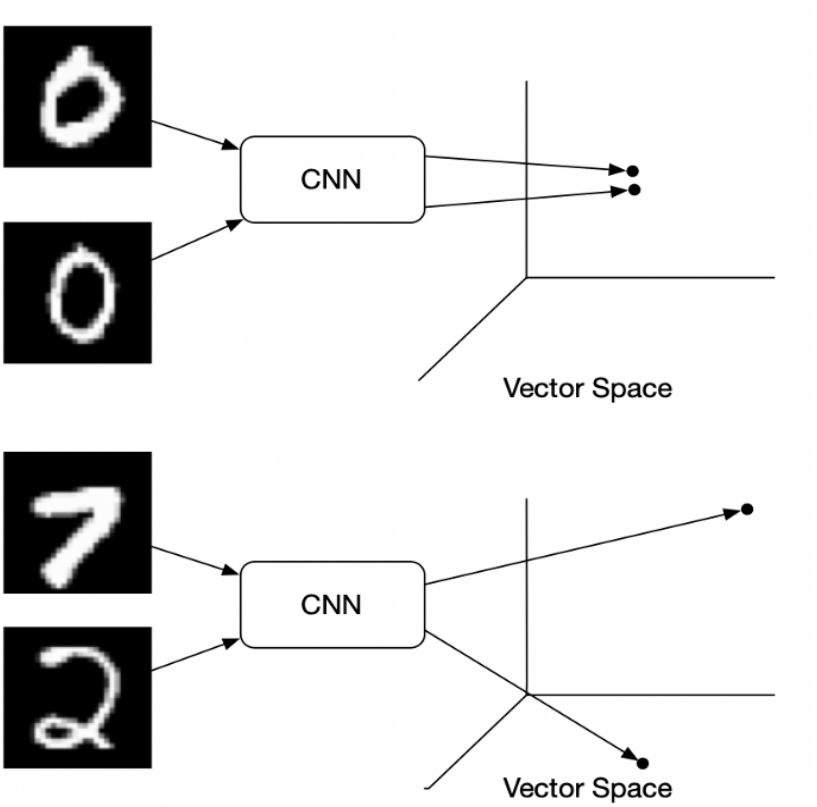

Figura 4.6: Exemple similare vs. diferite în spațiul reprezentărilor, sursa [30]

În ceea ce privește subtipurile principale ale contrastive loss, ele se diferențiază prin abordarea de învățare utilizată:

- **Self-supervised contrastive loss**
  În cazul self-supervised contrastive loss, imaginile ale căror trăsături se extrag reprezintă transformări ale aceleiași imagini, numită "ancoră". Se aplică o serie de operații de augmentare pe imaginea ancoră, după care se încearcă micșorarea în spațiul trăsăturilor a distanței dintre cele două, maximizând similaritatea 4.6.

$$l_{i,j} = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} exp(sim(z_i, z_k)/\tau)} \tag{4.6}$$

  unde z_i, z_j reprezintă transformările aceleiași imagini, z_k sunt transformările celorlalte imagini din același batch, sim este cosine similarity ca metrică de calcul pentru distanța dintre reprezentări, împărțită la factorul τ, denumit temperature

- **Supervised contrastive loss** implică existența unui set de date adnotat, deoarece în calculul erorii nu se folosesc transformări diferite ale aceleiași imagini, cum este în cazul self-supervised, ci imagini aparținând aceleiași clase. În acest context se pot utiliza două sau mai multe imagini, depinzând de funcția aleasă. Mai jos se găsește o listă neexhaustivă a funcțiilor contrastive loss supervised:

  - ***Triplet loss*** selectează un exemplu pozitiv (din aceeași clasă cu imaginea ancoră) și unul negativ (dintr-o clasă diferită de clasa imaginii ancoră). Formula triplet loss se găsește în 4.7, unde z_i este reprezentarea imaginii ancoră, z_j a imaginii pozitive, z_k a imaginii negative, iar m este parametrul margine, care păstrează distanța între perechile ancoră-imagine pozitivă și ancoră-imagine negativă.

$$l_{z_i,z_j,z_k} = max(0, ||z_i - z_j||_2^2, -||z_i - z_k||_2^2 + m) \tag{4.7}$$

  - ***Max margin contrastive loss*** se calculează folosind două reprezentări: ancora z_i și o altă imagine z_j, care poate avea aceeași etichetă sau una diferită. Ecuația este evidențiată în 4.8

$$l_{z_i,z_j} = ||z_i - z_j||_2^2 + max(0, m - ||z_i - z_j||_2)^2 \tag{4.8}$$

  - ***Supervised NT-Xent loss***, denumită SupCon în articolul în care a fost descrisă de către *Khosla et al.* [16]. SupCon se poate observa la 4.9, unde N este numărul de exemple din aceeași clasă.

$$l_{i,j} = -\frac{1}{2N_{y_i} - 1} \sum_{j=1}^{2N} log \frac{exp(z_i \cdot z_j/\tau)}{\sum_{k=1}^{2N} exp(z_i \cdot z_k/\tau)} \tag{4.9}$$

4.4. Regresie

Datorită faptului că acuitatea vizuală reprezintă o valoare reală, s-a evaluat calitatea encoder-ului folosind și un model de regresie liniară care să prezică acuitatea vizuală asociată encoding-urilor la nivel de volum. Pentru a rezolva această problemă, metricile urmărite sunt root mean squared error și mean absolute error. Mean absolute error măsoară magnitudinea medie a erorilor predicțiilor modelului și se calculează ca media diferențelor absolute dintre predicțiile modelului și valorile adevărate (formula 4.10). Root mean squared error, în schimb, extrage rădăcina pătrată a mediei diferențelor ridicate la puterea a doua dintre predicțiile modelului și valorile reale (funcția 4.11). Astfel, penalizarea este mai mare pentru erorile mari care pot sa apară.

$$MAE = \frac{1}{n} \sum_{j=1}^{n} |y_i - \hat{y_j}| \tag{4.10}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (y_i - \hat{y_j})^2} \tag{4.11}$$

4.5. Contrastive learning

A Simple framework for Contrastive Learning of Visual Representations (pe scurt SimCLR) a fost propus de *Chen et al.* [7] și reprezintă arhitectura cel mai frecvent menționată când vine vorba de metodologia contrastive learning. Arhitectura, ilustrată în figura 4.7, este alcătuită din următoarele elemente: un modul de augmentare a datelor, un encoder și un modul denumit projection head.

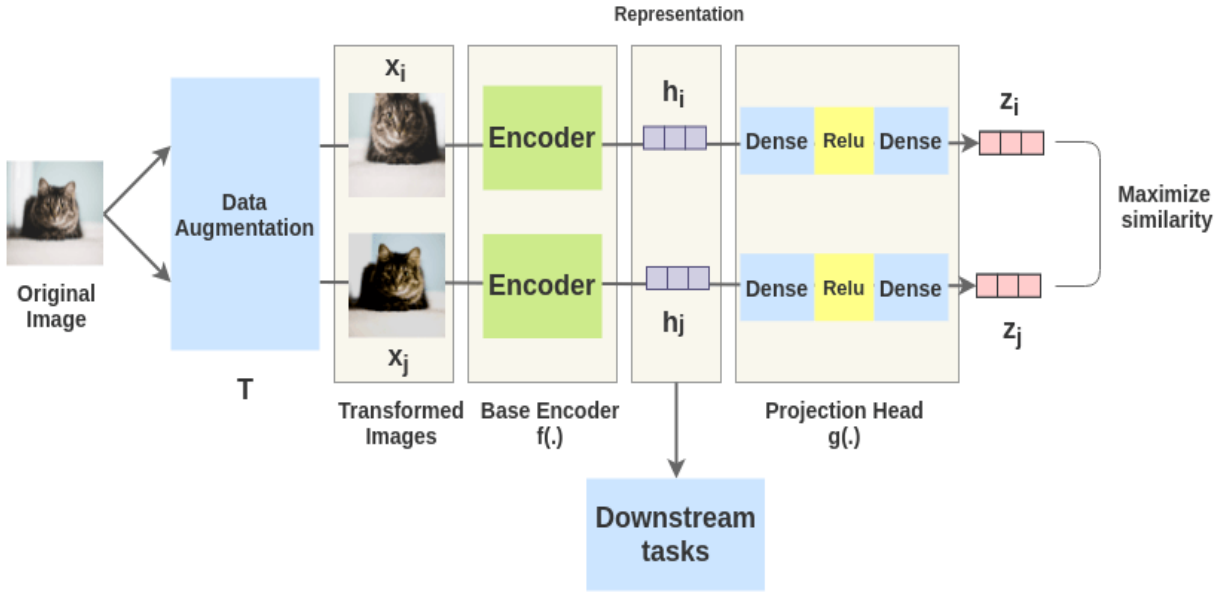

Figura 4.7: Arhitectura SimCLR, sursa [31]

4.5.1. Augmentarea datelor

Modulul de augmentare a datelor are rolul de a aplica un set de transformări asupra imaginilor de antrenare, pentru a genera exemple pozitive, din care modelul să extragă trăsăturile cheie. Astfel, modelul devine invariabil la transformări spațiale și de culoare.

Autorii menționează în articol faptul că au testat multiple combinații de transformări, dintre care unele spațiale (cropping, resizing, flipping, rotation, cutout), iar altele de culoare (color distortion, brightness, contrast, saturation, Gaussian blur, Sobel filtering). Aceste transformări efectuate se pot observa în figura 4.8.

Conform observațiilor, o singură transformare nu este suficientă pentru ca modelul să identifice trăsăturile necesare. Aceștia menționează că folosind perechea de transformări random cropping și color distortion s-au obținut cele mai bune rezultate. Explicația pentru utilizarea celor două transformări este că aplicând color distortion, modelul este forțat să privească dincolo de aspecte precum histograma de culori a imaginii, iar comparând două "bucăți" ale aceleiași imagini, obținute prin random cropping, reușește să identifice aspecte generale care pot fi prezente în diferite zone. Au demonstrat de asemenea faptul că este nevoie de transformări mai puternice pentru contrastive learning, în comparație cu supervised learning folosind cross-entropy.

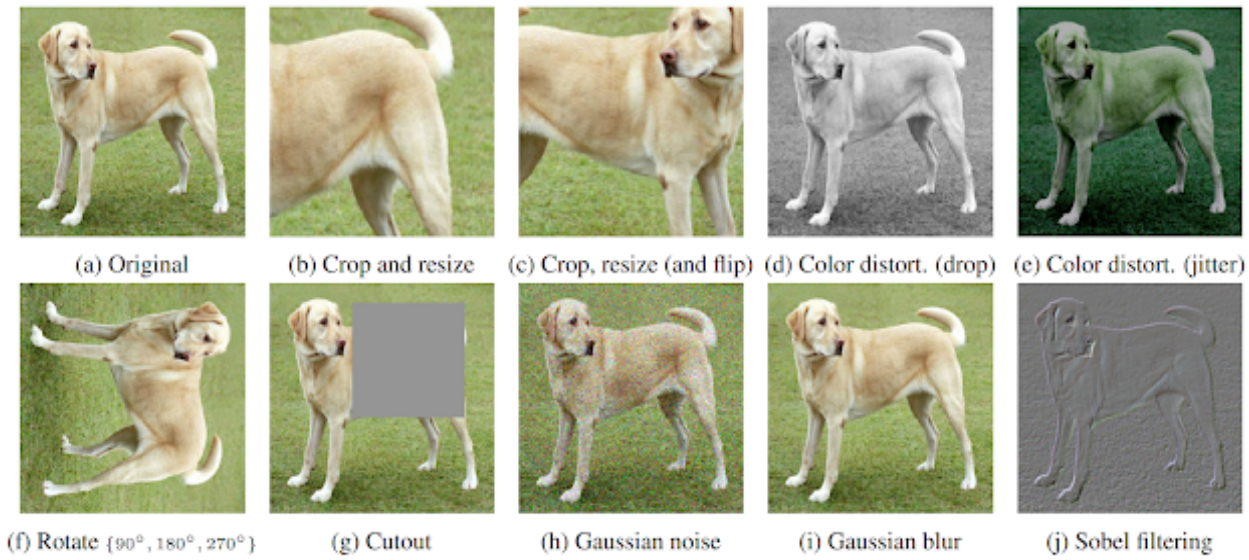

Figura 4.8: Exemple de transformări aplicate pe imaginile ancoră, sursa [7]

Modelul amintit în articol a fost testat pe setul de date ImageNet, iar în figura 4.9 se poate observa o analiză comparativă a impactului transformărilor alese pentru augmentare asupra acurateței rezultatelor. Pe diagonala principală se regăsesc valorile pentru o singură transformare, iar pe celelalte poziții din tabel sunt valorile corespunzătoare combinațiilor de transformări.

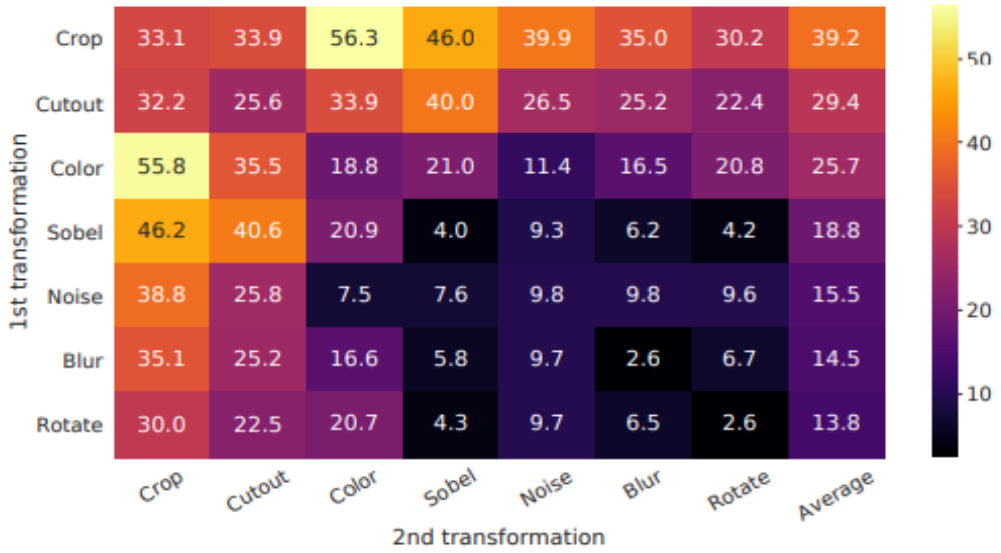

Figura 4.9: Impactul pe care îl au transformările incluse în procesul de augmentare asupra acurateței modelului liniar antrenat pe setul ImageNet, sursa [7]

4.5.2. Encoder

Cel de-al doilea modul al arhitecturii este encoder-ul. Fiecare imagine augmentată este trecută prin acest model pentru a extrage vectorul de trăsături. Modelul ales pentru a îndeplini funcția de encoder este generic și poate fi înlocuit cu orice altă arhitectură. El poate fi alcătuit din câteva layere de convoluție, sau poate fi reprezentat de o rețea neuronală convoluțională complexă. Autorii SimCLR au ales ca backbone arhitectura ResNet-50. Intrarea reprezintă o imagine RGB, cu trei canale, de dimensiuni (224, 224, 3), iar rezultatul encoder-ului este un vector de dimensiune 2048, reprezentând trăsăturile extrase. Pentru a obține o reprezentare mai compactă se poate opta pentru o dimensiune mai mică a vectorului de trăsături.

În urma antrenării modelului, rezultatul encoder-ului se poate utiliza ca bază pentru a rezolva alte probleme, cum sunt cele de clasificare. S-a demonstrat că acest proces de transfer aduce un plus de performanță față de modelele clasice supervised.

4.5.3. Projection head

Trăsăturile pe care le oferă encoder-ul sunt trecute ulterior printr-un modul numit projection head, care aplică o transformare neliniară a vectorului și îl mapează în spațiul reprezentărilor. Unul dintre experimentele efectuate de autori a fost alegerea acestui projection head. Arhitecturile care au inclus acest modul au demonstrat performanță mai bună față de cele care l-au omis. După antrenare, modulul se poate elimina pentru a utiliza rezultatul encoder-ului într-o altă problemă, denumită în practică "downstream task".

Projection head este format din layere de tip fully-connected, care aplică funcția de activare ReLU. Testele efectuate au arătat că două, până la 4 layere de tip dense ajută modelul să identifice aspecte importante în reprezentările obținute de encoder.

4.5.4. Minimizarea loss-ului

Odată identificate reprezentările imaginilor date ca input, se aplică funcția de loss de tip contrastive, care are ca scop minimizarea distanței dintre punctele corespunzătoare acelorași imagini în spațiul reprezentărilor și maximizarea distanței dintre punctele care corespund unor imagini diferite. În această arhitectură se poate utiliza contrastive loss de tip supervised sau self-supervised, în funcție de setul de date disponibil pentru antrenare

Articolul [7] evidențiază rezultatele obținute cu ajutorul contrastive loss self-supervised, în timp ce [16] motivează performanța mult mai bună a variantei supervised. Aceste aspecte depind semnificativ de resursele disponibile pentru antrenare. Principalele observații menționate, referitoare la arhitectura SimCLR și care se aplică la ambele tipuri de funcție de loss sunt:

- Scalarea arhitecturii și a resurselor imbunătățește semnificativ performanța, ceea ce se traduce în:
  - un batch size mai mare, pentru ca funcția de loss contrastive să analizeze comparativ cât mai multe imagini pentru a identifica trăsături cheie
  - utilizarea unei arhitecturi mai mari pentru encoder, pentru ca modelul să extragă cât mai multe detalii importante
- Antrenarea pentru mai multe epoci ajută la optimizarea rezultatelor funcției de loss în cazul contrastive, mai mult decât în contextul clasic supervised (unde în anumite situații se poate ajunge la saturație sau overfit)

După procesul de antrenare, encoder-ul poate fi inclus în rezolvarea unui așa-numit downstream task, care poate fi spre exemplu o problemă de clasificare. Reprezentările generale se extrag cu ajutorul encoder-ului, adăugat prin transfer ca backbone pentru un clasificator, care utilizează cross-entropy ca funcție de loss.

Capitolul 5. Proiectare de Detaliu și Implementare

5.1. Arhitectura sistemului

Figura 5.1 prezintă arhitectura sistemului. Modulul principal urmează framework-ul SimCLR, descris în secțiunea 4.5. Este alcătuit dintr-un modul de augmentare a datelor, un encoder și un projection head. Modulul de augmentare are rolul de a aplica transformări asupra imaginilor de intrare pentru a genera unele noi, care să fie mai departe reprezentate sub forma unui vector de trăsături. Encoder-ul generează acest vector, care este trimis mai departe către projection head pentru a fi mapat în spațiul trăsăturilor și a se optimiza valoarea funcției de loss de tip contrastive.

Modelul bazat pe metodologia contrastive learning include un downstream task și anume clasificarea setului de date Kermany [2] în cele patru clase pe care le include (CNV, DME, DRUSEN, NORMAL). Pentru a rezolva această problemă, modelul include o componentă numită linear probing, care este un simplu layer fully-connected cu 4 noduri, ce execută operația de clasificare. Rezultatul va fi clasa corespondentă imaginii input.

Cea de-a doua componentă a sistemului este clasificatorul. În acest modul, un nou set de date, care este structurat în funcție de valorile acuităților vizuale corespunzătoare imaginilor, este trecut prin primele două etape din framework-ul SimCLR. Imaginile sunt întâi transformate, iar apoi se extrag trăsăturile cele mai importante pe care le înglobează, folosind encoder-ul pre-antrenat în clasificarea setului Kermany. Setul de date utilizat este format din volume de imagini sau B-scans, care includ un minim de 25 de imagini. Din acest motiv, vectorii de trăsături proveniți dintr-un singur volum de imagini vor fi concatenați, pentru a obține un vector de trăsături de dimensiune (25, 128), unde 128 reprezintă dimensiunea vectorului extras de encoder pentru o imagine. Aceste imagini obținute prin concatenare vor constitui noul set de date pentru antrenarea unui clasificator care sa prezică acuitatea vizuală, măsurată cu ajutorul panoului Snellen.

5.2. Seturi de date

5.2.1. **Modelul contrastive**

Pentru modelul de tip contrastive learning am utilizat setul de date Kermany [2], care este la momentul actual cel mai mare set de date public de imagini OCT. El conține un subset de 83484 imagini de antrenare și un alt subset de 1000 de imagini de test, ambele structurate în cele 4 clase, care corespund anomaliilor retinei:

- Neovascularizare coroidiană (CNV)
- Edem macular diabetic (DME)
- Drusen (DRUSEN)
- Retină normală (NORMAL)

Acesta a fost încărcat folosind un API numit **tensorflow.data**, care include clasa **Dataset**, ce este foarte utilă pentru încărcarea seturilor de date de dimensiuni mari. Clasa pune la dispozitie metode care execută împărțirea setului în batch-uri, oferă posibilitatea de shuffle, este eficientă și ușor de folosit.

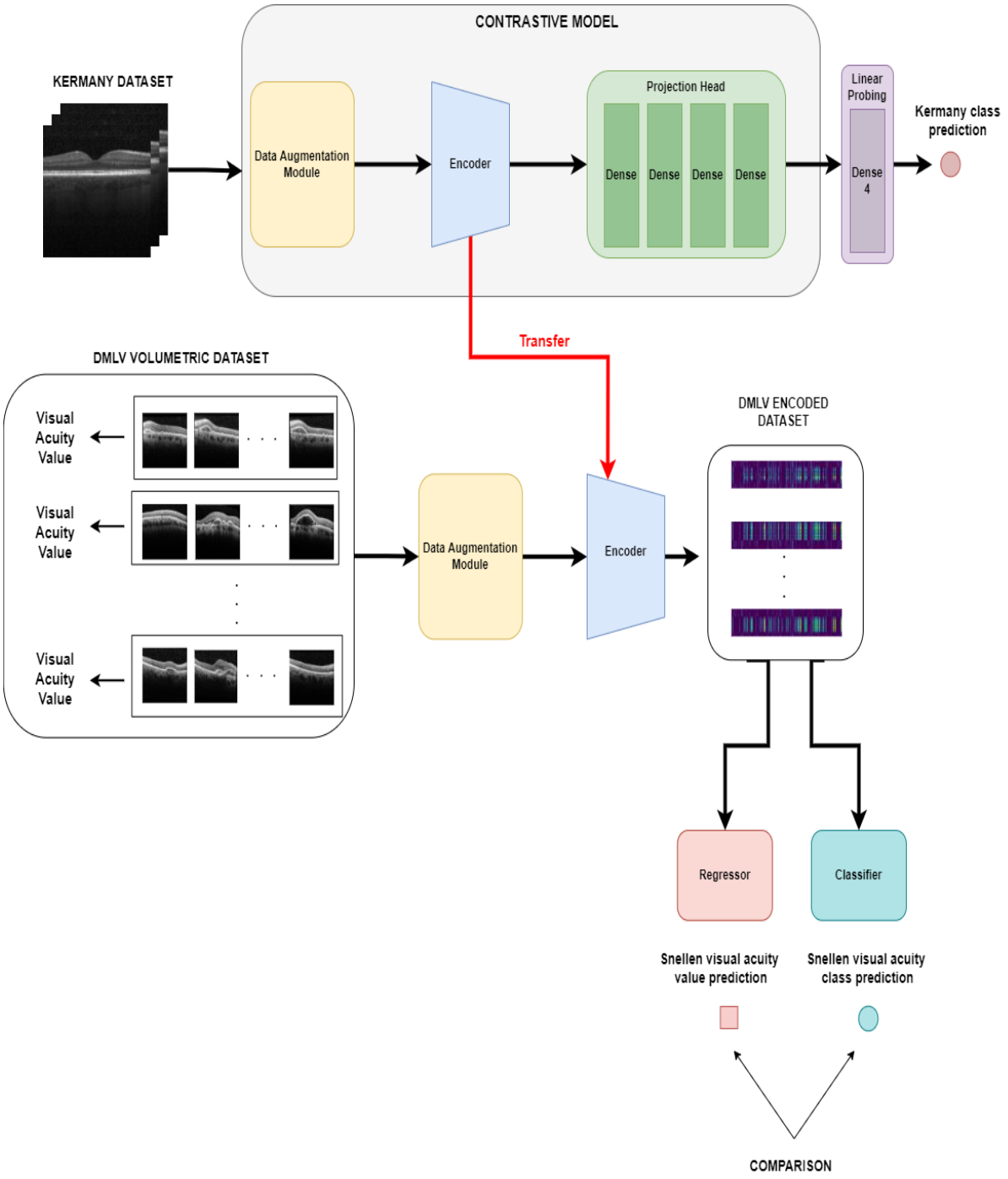

Figura 5.1: Arhitectura sistemului

5.2.2. **Predicția acuității vizuale**

În ceea ce privește predicția acuității vizuale, am utilizat setul de date provenit de la pacienți consultați la Spitalul Clinic Județean de Urgență din Cluj-Napoca, care suferă de DMLV. Imaginile sunt anonimizate, având la dispoziție doar id-ul pacientului, pentru a păstra confidențialitatea. Setul este structurat în 94 de foldere, care corespund celor 94 de pacienți consultați. În fiecare dintre cele 94 de foldere se găsesc vizitele pacienților de la o anumită dată, iar fiecare dintre ei are cel puțin două vizite. Fiecărei vizite îi corespund câte două seturi de B-scans, pentru cei doi ochi (stâng și drept). Un B-scan poate fi de tip fast, dense sau pole, adică include 25, 49, respectiv 61 de scanări OCT în secțiune transversală.

Setul de imagini vine însoțit de un tabel care conține valori ale acuității vizuale, măsurate în urma procedurii OCT, care corespund vizitelor. Cu toate acestea, anumiți pacienți au suferit complicații ale DMLV și au urmat anumite intervenții chirurgicale sau prezintă alte afecțiuni precum retinopatie diabetică, motiv pentru care, la recomandarea medicilor specialiști de la Spitalul Clinic Județean de Urgență Cluj-Napoca, au fost excluși din studiu, eliminând imaginile corespunzătoare. O altă problemă care intervine este faptul că în cazul anumitor vizite ale unor pacienți acuitatea vizuală nu a fost măsurată în aceeași zi în care s-a realizat procedura OCT. Pentru acele situații s-a luat în considerare valoarea acuității de la cea mai apropiată vizită, la o diferență de maxim 30 de zile de la efectuarea scanării OCT.

Pentru a putea extrage doar pacienții eligibili analizei imaginilor OCT, am ales să folosesc câte un obiect **pandas.Dataframe** pentru a stoca informațiile referitoare la volumele de imagini OCT, respectiv informațiile referitoare la valorile acuității vizuale. În primul Dataframe am păstrat pentru fiecare imagine din setul de date indexul folderului pacientului, id-ul său, numărul vizitei, data vizitei și ochiul căruia îi corespunde imaginea (se poate observa în figura 5.2).

	Image name	Image index	Patient folder	Patient ID	Visit	Date	Eye
0	325139E0.tif	1	1	246	1	09.02.2017	right
1	32561BE0.tif	2	1	246	1	09.02.2017	right
2	325AD6D0.tif	3	1	246	1	09.02.2017	right
3	326202C0.tif	4	1	246	1	09.02.2017	right
4	3266BDB0.tif	5	1	246	1	09.02.2017	right
...	...	...	...	...	...	...	...
19031	90AC70.tif	21	94	5018	4	11.03.2019	left
19032	956760.tif	22	94	5018	4	11.03.2019	left
19033	9C9350.tif	23	94	5018	4	11.03.2019	left
19034	A14E40.tif	24	94	5018	4	11.03.2019	left
19035	A60930.tif	25	94	5018	4	11.03.2019	left

Figura 5.2: Dataframe-ul cu informațiile referitoare la imaginile OCT

În cel de-al doila Dataframe am păstrat informațiile referitoare la indexul folder-ului pacientului, id-ul său, data vizitei, ochiul al cărei acuitate a fost măsurată, valoarea acuității și modul în care a fost măsurată (ex. fc - fără corecție, ccp - cu corecție proprie, cps - cu punct stenopeic etc.). În figura 5.3 este ilustrat Dataframe-ul cu valorile acuității. Astfel, procesul de filtrare a informațiilor a fost simplificat semnificativ.

	Patient folder	Patient ID	Visit Date	Eye	Acuity	Acuity types
0	1	246	04.03.2019	OD	0.050	fc
1	1	246	04.02.2019	OD	0.010	fc
2	1	246	01.08.2018	OD	0.500	fc
3	1	246	27.06.2018	OD	0.100	fc
4	1	246	23.05.2018	OD	0.100	fc
...	...	...	...	...	...	...
743	94	5018	06.02.2019	OS	0.025	ccp
744	94	5018	09.01.2019	OS	0.025	ccp
745	94	5018	05.12.2018	OS	0.025	fc
746	94	5018	05.11.2018	OS	0.025	fc
747	94	5018	24.09.2018	OS	0.320	ccp

Figura 5.3: Dataframe-ul cu informațiile referitoare la valorile acuității vizuale

În urma excluderii imaginilor menționate anterior a rezultat un număr de 551 de B-scans, de câte 25 de imagini (pentru B-scans de tip dense și pole am selectat doar 25 de imagini din volum). Aceste 551 de B-scans au fost transformate cu ajutorul encoder-ului în vectori de trăsături de dimensiune (25, 128), care formează setul de date pentru predicția acuității vizuale a unui pacient. Valorile rezultate de encoder au fost normalizate, pentru a putea converti vectorii în imagini.

În secțiunile de cod 5.1, 5.2 și 5.3 am atașat logica generării setului de date pentru predicția acuității. Pentru fiecare imagine din setul de date OCT se creează o imagine, prin conversia vectorului de trăsături și normalizarea lui. Datorită faptului că vectorul este rezultatul encoder-ului, este reprezentat de un obiect de tip **tensor** de dimensiune (25, 128). Informațiile referitoare la imaginile OCT ți valorile acuității au fost extrase din cele două obiecte Dataframe menționate anterior.

```
1 def generate_volumetric_dataset(model, images_dir, image_size,
    normalize_factor):
2     # Generates dataset made of volumetric images, normalized with
    normalize_factor
3
4     b_scan_paths = glob.glob(images_dir)
5     images = glob.glob(images_dir + '/*.tif')
6     for p in b_scan_paths:
7         generate_volumetric_image(model, images, p, image_size,
    normalize_factor)
```

Secțiunea de cod 5.1: Generarea setului de date pentru predicția acuității

```
def generate_volumetric_image(model, images, b_scan_path, image_size,
    normalize_factor):
     #Converts a 3D tensor generated by get_volumetric_tensor into an
    image of size 25 x encoder_size
     .   .   .

     if not is_removable(patient, eye):
         # get acuity that corresponds to patient, at visit_date, for
    eye
         acuity = get_acuity(patient, visit_date, eye)
         if acuity is not None:
             image_names = get_image_names_for_B_scan(patient,
    visit_date, eye)
             local_paths = get_local_paths_from_image_names(images,
    image_names) # Get local paths corresponding to images names in
    dataframe
             volumetric_tensor = get_volumetric_tensor(model,
    local_paths, image_size)
             volumetric_tensor = tf.math.divide(volumetric_tensor,
    normalize_factor)

             tensor_file_name = "Patient_" + patient + "_visit_" +
    visit_nr + "_" + visit_date + "_eye_" + eye + "_acuity_" + str(
    acuity) + acuity_type
             write_tensor_in_dataset(tensor_file_name, volumetric_tensor
    )
```

Secțiunea de cod 5.2: Generarea imaginilor care constituie setul de date

```
def get_volumetric_tensor(encoder_contrastive, image_paths_list,
    image_size):
     #Generates a 3D tensor from the values generated by the encoder
    from images in image_paths_list
     input_img = keras.Input(shape=(image_size, image_size,
    image_channels))
     encoder_model = keras.Sequential(
         [
             input_img,
             get_augmenter(**classification_augmentation),
             encoder_contrastive
         ],
         name="encoder",
     )
     encodings = []

     for image_path in image_paths_list:
         image = tf.keras.utils.load_img(image_path, target_size=(
    image_size, image_size, 3))
         img_array = tf.keras.utils.img_to_array(image)
         img_array = tf.expand_dims(img_array, 0)
         prediction = encoder_model.predict(img_array)
         encodings.append(prediction)

     imgs.append(image_paths_list)
     volumetric_tensor = layers.Concatenate(axis=0)(encodings)
     return volumetric_tensor
```

Secțiunea de cod 5.3: Generarea unui tensor de trăsături

5.3. Modelul contrastive

5.3.1. Modulul de augmentare

În secțiunea 4.5.1, am menționat experimentele efectuate de autorii arhitecturii SimCLR, care au identificat transformările care au cel mai mare impact pozitiv asupra rezultatelor encoder-ului și anume cropping, random flip și color distortion.

Aceste transformări au fost integrate în dezvoltarea aplicației, pentru a augmenta imaginile OCT. Am folosit librăria **tensorflow.keras.layers**, care pune la dispoziție layere de preprocesare a imaginilor, care să fie integrate direct în procesul de antrenare. Exemple de astfel de layere, pe care le-am utilizat, sunt: **Rescaling**, **RandomFlip**, **RandomZoom**, **RandomTranslation**. Nu în ultimul rând, pentru a obține color distortion am aplicat un factor de modificare a luminozității și unul de jitter, pentru a transforma valorile pixelilor din cele trei canale ale imaginii. Modelul contrastive obține performanță mai bună atunci când se aplică transformări mai puternice, motiv pentru care augmentările pentru contrastive learning sunt mai intense decât cele pentru clasificare cu cross-entropy.

Avantajele integrării unui pipeline de augmentare a imaginilor direct în procesul de antrenare sunt:

- Augmentarea va beneficia de putere de calcul, dacă modelul este antrenat folosind unitate de procesare grafică sau GPU, ceea ce elimină constrângerile de resurse, care trebuie alocate pentru pipeline-ul de procesare și de care un CPU este posibil să nu dispună
- Ușurează procesul de deployment, datorită faptului că este încapsulat în model

În experimentele efectuate am inclus și alte transformări precum cutout, jigsaw și random grid shuffle, însă niciuna nu a reușit să aducă un impact la fel de puternic asupra performanței encoder-ului cum este impactul oferit de combinația cropping - random flip - color distortion.

5.3.2. Encoder

În articolul [7], autorii menționează un aspect important identificat în urma experimentelor pe care le-au efectuat și anume faptul că SimCLR beneficiază de performanță mai ridicată atunci când se utilizează o arhitectură mai mare și cu mai multe layere, deoarece encoder-ul reușește să extragă mai multe trăsături importante și specifice imaginilor de antrenare. Având această informație, alături de aspectele menționate în secțiunea 3.5 am optat să includ modelul DenseNet121 ca backbone pentru encoder, folosind tehnica transfer learning. Arhitectura encoder-ului este ilustrată în figura 5.4.

Secvența de cod 5.4 prezintă construcția arhitecturii encoder-ului. Am pornit de la modelul pre-antrenat pe setul de date ImageNet, din librăria **keras.applications.densenet**, inițializând weights cu cele corespunzătoare setului ImageNet. Parametrul trainable poate lua 3 valori:

- pentru valoarea 0 modelul nu va putea fi antrenat, iar în acest fel poate fi utilizat prin transfer în altă arhitectură, cu weight-urile configurate din antrenarea pe un alt set de date
- pentru valoarea 1 modelul va putea fi antrenat în întregime
- folosind valoarea 2 modelul va putea fi fine-tuned, oferind posibilitatea de a antrena pe un alt set de date primele 5 layere din arhitectura DenseNet121

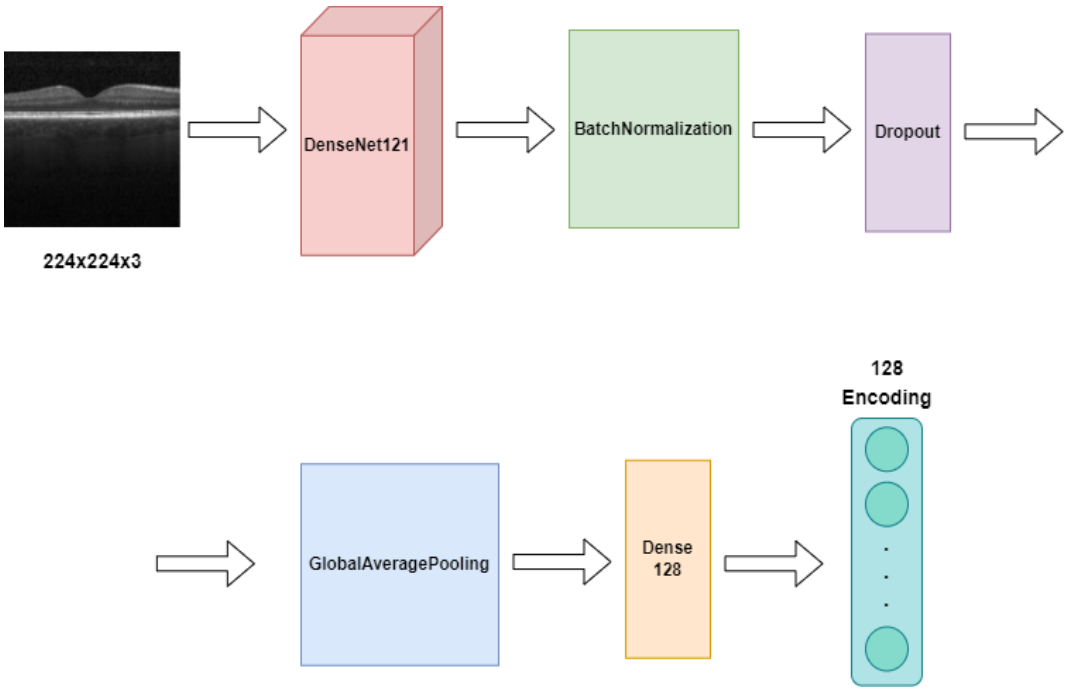

Figura 5.4: Arhitectura encoder-ului

În continuarea modelului DenseNet121 am adăugat încă 4 layere. Batch normalization va face rețeaua mai rapidă și mai stabilă. Această operație presupune normalizarea valorilor obținute ca output la modelul transferat, la nivelul fiecărui batch de imagini. În urma normalizării, toate valorile din batch vor fi distribuite în același interval. Layer-ul de Dropout a fost introdus pentru a preveni scenariul de overfit. Acționează prin a "ignora" output-urile anumitor neuroni aleși la întâmplare cu o frecvență dată (în acest caz 0.2). Ultimele două layere sunt cele de Global average pooling și unul de output de tip Dense. Cele două layere combinate vor oferi rezultatul extragerii trăsăturilor, ca feature vector de o anumită dimensiune (valoarea aleasă este 128, în urma experimentelor cu mai multe valori).

```
1  def get_encoder_dense(image_size, image_channels, code_width, trainable
       ):
2      model = DenseNet121(weights="imagenet", include_top=False,
       input_shape=(image_size, image_size, image_channels))
3
4      # configure model's trainable layers
5      if trainable == 0:
6          model.trainable = False
7      elif trainable == 1:
8          model.trainable = True
9      elif trainable == 2:
10         layers_dense_trainable = model.layers[:5]
11         layers_dense_nontrainable = model.layers[5:]
12         for layer_dense in layers_dense_trainable:
13             layer_dense.trainable = True
14         for layer_dense in layers_dense_nontrainable:
15             layer_dense.trainable = False
16
17     batch_norm_layer = layers.BatchNormalization()
18     dropout_layer = layers.Dropout(0.2)
```

```
19      global_avg_pooling = layers.GlobalAveragePooling2D()
20      dense_layer = layers.Dense(code_width, activation='relu')
21
22      return models.Sequential([
23          model,
24          batch_norm_layer,
25          dropout_layer,
26          global_avg_pooling,
27          dense_layer
28      ], name="encoder")
```

Secțiunea de cod 5.4: Modelul encoder

5.3.3. **Projection head**

Projection head are rolul de a evidenția trăsături invariante în imaginile de antrenare și de a maximiza abilitatea rețelei de a identifica diferite transformări ale aceleiași imagini. Autorii SimCLR framework menționează că testele efectuate au demonstrat îmbunătățirea performanței encoder-ului cu până la 10% folosind acest modul.

Modulul este alcătuit din niște simple layere de tip Dense, de dimensiunea encoder-ului, adică 128. Am ales să includ 4 astfel de layere, după ce am observat o îmbunătățire a acurateței modelului de a reprezenta imaginile OCT, comparativ cu doar două layere.

5.3.4. **Minimizarea loss-ului**

Pentru a antrena modelul și a ajusta cât mai bine parametrii encoder-ului astfel încât să reprezinte cât mai potrivit imaginile OCT, am folosit contrastive loss ca funcție de optimizare. Am abordat ambele alternative, atât self-supervised cât și supervised. În următorul capitol voi rezuma rezultatele obținute, în urma cărora am decis să utilizez mai departe encoder-ul antrenat în stil supervised, iar în această secțiune voi detalia ambele metode.

- **Contrastive self-supervised**
  În ceea ce privește abordarea self-supervised, o imagine ancoră trece prin primele două module, de augmentare și encoding, rezultând doi vectori de trăsături de dimensiune 128. Ei sunt mapați cu ajutorul projection head în spațiul trăsăturilor, iar rezultatele sunt folosite ca valori pentru calculul self-supervised contrastive loss (formula 4.6), iar parametrul temperature este setat la valoarea 0.1 (valoare recomandată de autorii SimCLR). Mai apoi, ca în orice rețea neuronală clasică, prin procesul back-propagation se optimizează parametrii modelului, pentru a obține o reprezentare cât mai bună a imaginilor OCT.

- **Contrastive supervised**
  În scenariul supervised am folosit 4.9 ca formulă de calcul a contrastive loss. Valorile cu care se calculează acest loss sunt reprezentate de imaginile OCT trecute prin etapele menționate anterior (augmentare, encoding și projection head), împreună cu etichetele lor. Având la dispoziție setul etichetat Kermany, se iau în considerare ca imagini similare, instanțe din aceeași clasă (CNV, DME, DRUSEN, NORMAL), iar ca exemple negative, instanțe din clase distincte. Mai departe, la fel ca în contextul self-supervised, se aplică back-propagation pentru a ajusta rețeaua.

Cu scopul de a evalua performanța antrenării modelului contrastive am optat pentru un downstream task care să execute clasificarea setului Kermany. Pentru acest task am adăugat un layer de tip Dense suplimentar, denumit în literatură linear probing, care conține 4 neuroni ce vor avea scopul de a oferi probabilitatea apartenenței unei imagini OCT la una dintre cele 4 clase din setul Kermany.

Funcția de loss cu care se optimizează acest layer este cross entropy, implementată cu ajutorul clasei **keras.losses.SparseCategoricalCrossentropy**. Performanța întregului clasificator a fost monitorizată folosind **keras.metrics.SparseCategoricalAccuracy**.

5.4. Modelul de predicție a acuității vizuale

Modelul contrastive oferă rezultate impresionante în clasificarea setului de date Kermany. În continuare am optat să analizez rezultatele pe care le oferă encoder-ul în analiza imaginilor OCT afectate de DMLV și predicția acuității vizuale asociate. Pentru asta, am folosit encoder-ul și am generat un set de date format din imagini volumetrice la nivel de B-scan. Fiecare imagine dintr-un B-scan, care corespunde unuia dintre ochii unui pacient de la o anumită vizită, a fost transformat într-un vector de trăsături de dimensiune 128.

Concatenând cei 25 de vectori, care reprezintă cele 25 de imagini aferente unui B-scan, obținem un vector de trăsături de dimensiune 25x128, care are asociată o acuitate vizuală, pe care am convertit-o în valoare zecimală. Conversia a fost făcută conform tabelei 5.1. Valoarea acuității reprezintă clasa căreia aparține acel vector de dimensiuni 25x128.

Valoare Snellen	Valoare zecimală
20/20	1.0
20/25	0.8
20/32	0.63
20/40	0.5
20/50	0.4
20/63	0.32
20/100	0.2
20/200	0.1
20/400	0.05
20/800	0.025
20/2000	0.01
0	0.0

Tabela 5.1: Conversia acuității vizuale din sistemul Snellen imperial în sistem zecimal

Am ales să evaluez reprezentarea rezultată de encoder atât printr-un task de clasificare, cât și unul de regresie.

În ceea ce privește clasificarea, modelul include trei layere de convoluție, urmate fiecare de câte un layer de tip max pooling. După aceste blocuri am adăugat layere de tip Dropout, deoarece setul de date nu este echilibrat ca număr de instanțe la nivelul fiecărei clase, ceea ce poate produce overfit.

Hidden layer-ul de tip fully-connected folosește metoda de regularizare de tip L2, care are de asemenea scopul de a preveni scenariul de overfit. Ultimul layer este cel de output, care are ca rezultat probabilitatea Softmax a apartenenței imaginii de intrare la fiecare din cele 12 clase. Optimizarea s-a realizat cu funcția cross entropy. Arhitectura este conturată în figura 5.5.

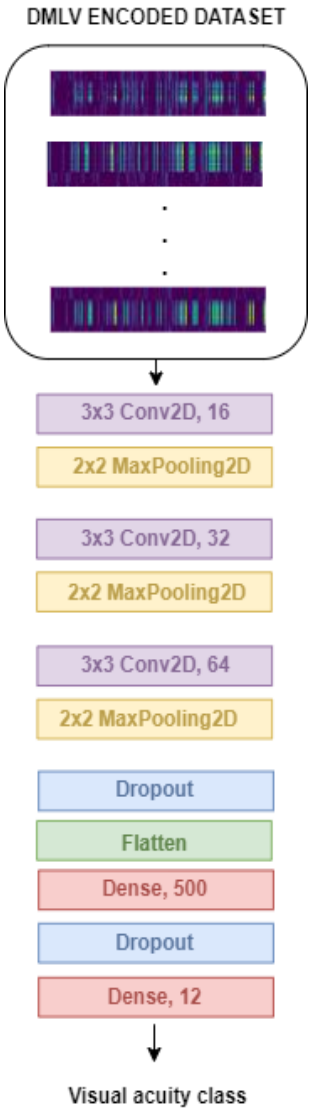

Figura 5.5: Arhitectura clasificatorului

Datorită faptului că acuitatea vizuală este o valoare reală, dar și pentru că măsurarea ei reprezintă un proces subiectiv din partea medicilor specialiști, am ales să experimentez și un model de regresie pentru predicție.

Diferența dintre o problemă de clasificare și una de tip regresie este dată de faptul că în cazul regresiei, rezultatul obținut nu este o etichetă a cărei valoare aparține unei mulțimi finite de clase, ci este o valoare reală și continuă care este situată în același interval cu valorile etichetelor corespunzătoare instanțelor de antrenare.

Algoritmul ales este Support Vector Regression sau SVR, care este un model liniar ce va prezice valoarea acuității vizuale. Modelul are ca obiectiv minimizarea erorilor, reprezentate de diferențele dintre valoarea prezisă și cea adevărată. Features pentru acest algoritm reprezintă media valorilor de pe fiecare poziție din fiecare vector de trăsături, la nivelul volumului de B-scans. Astfel, din fiecare vector de dimensiune 25x128, aferent unei vizite a unui pacient pentru unul dintre ochi, se vor obține doar 128 de features, pe care modelul SVR le va folosi, reducând și mai mult dimensionalitatea reprezentărilor. Avantajul SVR este dat de faptul că oferă posibilitatea modificării pragului erorii, pentru a nu penaliza anumite predicții care pot fi foarte apropiate de valoarea reală. Aplicarea algoritmului SVR este evidențiată în figura 5.6

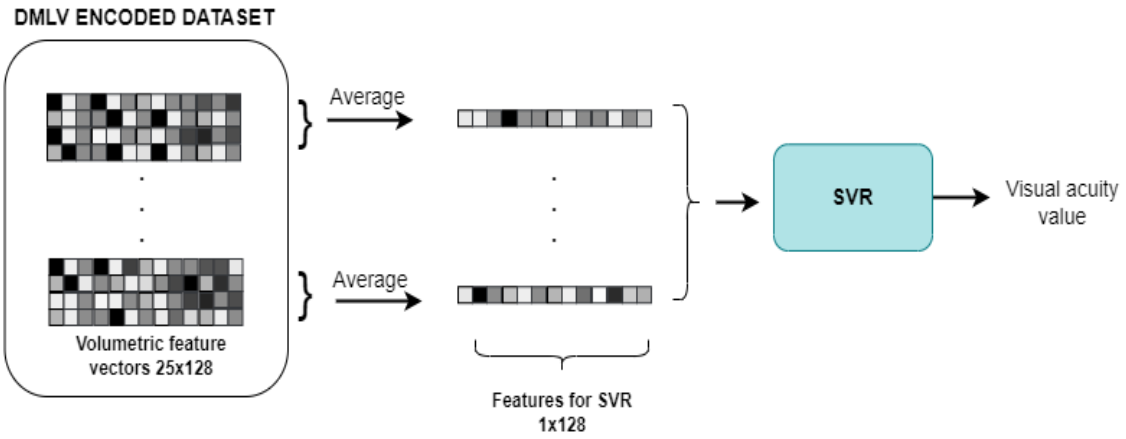

Figura 5.6: Aplicarea algoritmului SVR

Capitolul 6. Testare și Validare

6.1. Date de test

6.1.1. Modelul contrastive

Setul de date de test utilizat pentru evaluarea clasificării modelului contrastive, și implicit a performanței encoder-ului, este setul de test Kermany, care conține 1000 de imagini, câte 250 din fiecare clasă prezentă și în setul de antrenare (CNV, DME, DRUSEN, NORMAL). Setul de antrenare conține 83484 de imagini, care reprezintă B-scans individuale ale pacienților.

6.1.2. Predicția acuității vizuale

Din setul de date format din vectorii de trăsături, obținuți cu ajutorul encoder-ului din setul de date DMLV, am selectat 70 de pacienți, din cei 80 pe care i-am păstrat în studiu în urma eliminării celor care nu au fost eligibili. Instanțele aferente acestor pacienți au fost folosite pentru antrenare, în timp ce vizitele celor 10 pacienți rămași au fost păstrate pentru testarea clasificării și a regresiei.

6.2. Modelul Contrastive

6.2.1. Experimente

Pe parcursul dezvoltării acestui sistem am condus o serie de experimente, în urma cărora am ales arhitectura detaliată în capitolul anterior ca fiind cea mai bună pentru atingerea obiectivelor acestei lucrări. Criteriile după care am evaluat diferitele configurații de modele încercate sunt: valoarea funcției de loss de tip contrastive în timpul antrenării pe setul Kermany, valoarea funcției de loss cross-entropy pentru problema de clasificare a setului Kermany și nu în ultimul rând acuratețea obținută la clasificare pe setul Kermany de antrenare, validare și test. Setul de antrenare este format din 83484 de imagini, structurate în 4 clase, distribuite conform figurii 6.1, din care am utilizat 80% pentru antrenare și 20% pentru validare.

Voi detalia în cele ce urmează o parte din observațiile pe care le-am colectat în urma experimentelor.

1. **Self-supervised vs. supervised**
   După cum am menționat în capitolul anterior, arhitectura contrastive a fost antrenată atât folosind funcția de loss de tip self-supervised, cât și alternativa supervised. Autorii articolului [16] au demonstrat recent faptul că supervised contrastive learning este abordarea care a oferit perfomanță mai ridicată pe setul de date ImageNet. Am început cu abordarea self-supervised, antrenând modelul folosind setul Kermany, fără a utiliza etichetele în calculul contrastive loss. Am încercat diferite configurații ale hiper-parametrilor arhitecturii, modificând dimensiunea unui batch (valorile încercate au variat de la 64, la 128, 512 și chiar 1024), numărul de epoci de antrenare și aplicând tehnici de regularizare cum ar fi early stopping sau reduce learning rate on plateau.

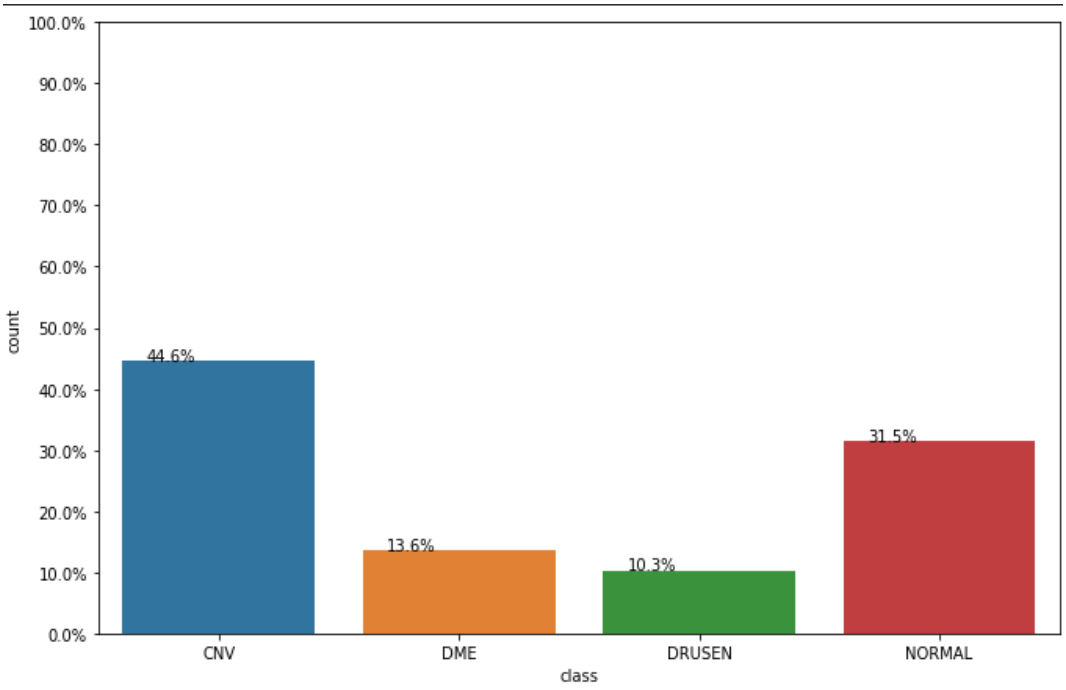

Figura 6.1: Distribuția claselor în setul de antrenare Kermany

Cel mai bun model obținut a fost antrenat timp de 110 de epoci, folosind arhitectura DenseNet121 ca encoder, cu un batch size de 64 de imagini și dimensiunea encoding-ului de 1024 de trăsături. Am folosit 20% din setul de antrenament Kermany pentru validare și am testat pe setul de test Kermany.
Am folosit aceleași configurații ale arhitecturii pentru a antrena un model în stil supervised. De această dată am utilizat funcția de loss contrastive supervised, motiv pentru care am avut nevoie de etichetele setului de antrenare Kermany.
O comparație a rezultatelor obținute pentru cele două modele se poate observa în tabela 6.1.

Configurații	**Self-supervised**	**Supervised**
Encoder	DenseNet121	DenseNet121
Dimensiune encoding	1024	1024
Epoci	110	100
Dimensiune batch	64	64
Acuratețe maximă pe setul de validare	88%	95%
Acuratețe Kermany test	95%	98%
Număr de predicții fals-pozitive	19	0

Tabela 6.1: Analiza comparativă a utilizării funcțiilor de loss contrastive self-supervised și contrastive supervised

Având aceleași configurații, modelele s-au comportat diferit, demonstrând superioritatea utilizării funcției contrastive de tip supervised, care a obținut o acuratețe mai bună la testarea pe setul Kermany și a rezultat 0 predicții fals-pozitive (instanțe de imagini care prezentau afecțiuni ale retinei însă au fost clasificate ca fiind normale).

O altă analiză pe care am efectuat-o a fost să calculez cosine similarity între predicțiile oferite de cele două modele, pentru a investiga reprezentările ca vector de trăsături pe care encoder-ul le oferă.
Am ales la întâmplare câte o instanță din fiecare clasă din setul de test, am transformat-o în vector de trăsături, folosind cele două modele și am calculat cosine similarity între toate perechile de clase. În urma câtorva teste am obținut rezultatele afișate în figurile 6.2 și 6.3.

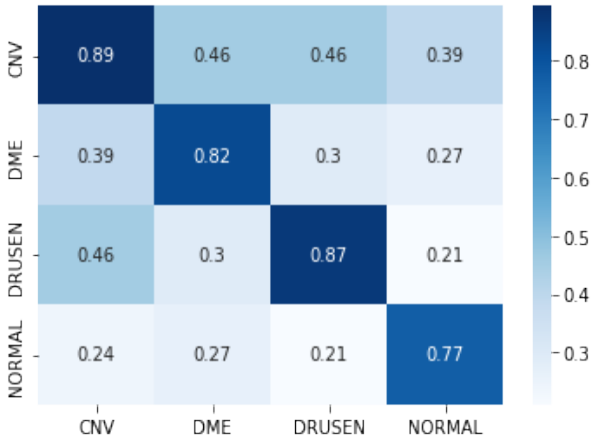

Figura 6.2: Valorile cosine similarity, calculate pentru vectorii de trăsături rezultați de encoder-ul antrenat cu contrastive self-supervised loss

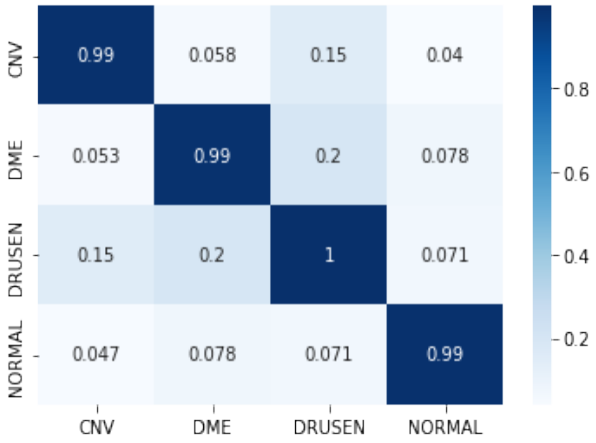

Figura 6.3: Valorile cosine similarity, calculate pentru vectorii de trăsături rezultați de encoder-ul antrenat cu contrastive supervised loss

Așadar, se poate observa cu ușurință faptul că graficul din figura 6.2 include valori mai mari pe pozițiile care nu aparțin diagonalei principale, ceea ce înseamnă că encoder-ul antrenat cu funcția de loss self-supervised distinge mai greu aspectele specifice fiecărei afecțiuni reprezentate de cele patru clase. Pe de altă parte, figura 6.3 evidențiază rezultatele mult mai bune pe care abordarea supervised le oferă.

Din aceste motive, sistemul detaliat în această lucrare include encoder-ul antrenat în stil supervised.
O altă configurație pe care am încercat-o este antrenarea unui encoder folosind ambele seturi de date în stil self-supervised, atât setul DMLV cât și o parte din Kermany și validarea cu ajutorul setului Kermany-test pentru clasificare. Pentru asta, am selectat diferite valori ale batch size, astfel încât să includ din ce în ce mai multe imagini din setul Kermany, pentru a urmări impactul lor asupra acurateței de reprezentare a encoding-ului. Rezultatele sunt prezente în tabela 6.2.

Batch size	Nr. imagini DMLV din batch	Nr. imagini Kermany din batch	Acurateţe antrenare self-supervised
128	122	5	61%
512	512	0	66%
512	493	23	67%
512	104	409	78%
1024	208	818	81%

Tabela 6.2: Analiza comparativă a utilizării setului de date DMLV împreună cu setul Kermany pentru antrenarea self-supervised

După cum se poate observa din tabela de mai sus, un batch size mai mare, care să includă cât mai multe imagini din setul Kermany, rezultă într-o acurateţe crescută, ceea ce poate fi un indiciu pentru utilizarea setului adnotat în scenariul supervised.

2. **Arhitectura encoder-ului**
Am ales trei modele principale ca arhitectură pentru encoder: SegNet, ResNet și DenseNet. Am efectuat diferite experimente, modificând hiper-parametrii modelului. Concluziile acestor experimente au fost că arhitectura SegNet (formată din 6 blocuri de convoluție) nu este suficient de complexă încât să surprindă caracteristicile cheie din imaginile OCT, atunci când este antrenată folosind contrastive learning. Pe de altă parte, arhitectura ResNet50, menționată și de autorii frameworkului SimCLR poate fi o alternativă bună la modelul DenseNet121, deoarece ambele au obținut performanțe apropiate. Cu toate acestea, cel mai bun model obținut este cel care include modelul DenseNet ca encoder. Analiza comparativă este ilustrată în tabela 6.3.

Encoder	Acurateţea maximă cu encoder self-supervised - clasificare set Kermany
SegNet	70%
ResNet	89%
DenseNet	95%

Tabela 6.3: Analiza comparativă a utilizării diferitor ahitecturi pentru encoder

3. **Dimensiunea encoding-ului**
   Un alt aspect care merită discutat este alegerea dimensiunii potrivite pentru encoder. Cu resursele disponibile am reușit să antrenez arhitectura folosind următoarele valori pentru dimensiunea encoding-ului: 16, 64, 128, 256, 1024. Aspectul pe care l-am observat este faptul că modelul antrenat cu self-supervised contrastive loss a obținut cele mai bune rezultate pe măsură ce am crescut dimensiunea vectorului de trăsături. Diferența dintre acuratețea obținută de modelul cu ecoding de 256 și cea obținută de modelul cu encoder de dimensiune 1024 este de 10%. Pe de altă parte, modelul supervised a oferit rezultate mai bune folosind encoding de 128, comparativ cu encoding de 1024. Aceste observații sunt susținute și de rezultatele menționate în experimentul anterior, care arată că modelul supervised este capabil să extragă informații cheie în formă mai compactă, în timp ce abordarea self-supervised generează unele erori.

6.2.2. Evaluarea modelului

În urma experimentelor detaliate anterior, am ajuns la concluzia că modelul DenseNet121 este arhitectura potrivită pentru encoder, iar dimensiunea vectorului de trăsături care a adus cele mai bune rezultate este 128.

Astfel, pentru problema de clasificare a setului Kermany am antrenat modelul pentru 22 de epoci, cu un batch size de 128, și regularizare de tip reduce learning rate on plateau cu factor patience de 5, iar valorile obținute se află în figurile 6.4 și 6.5.

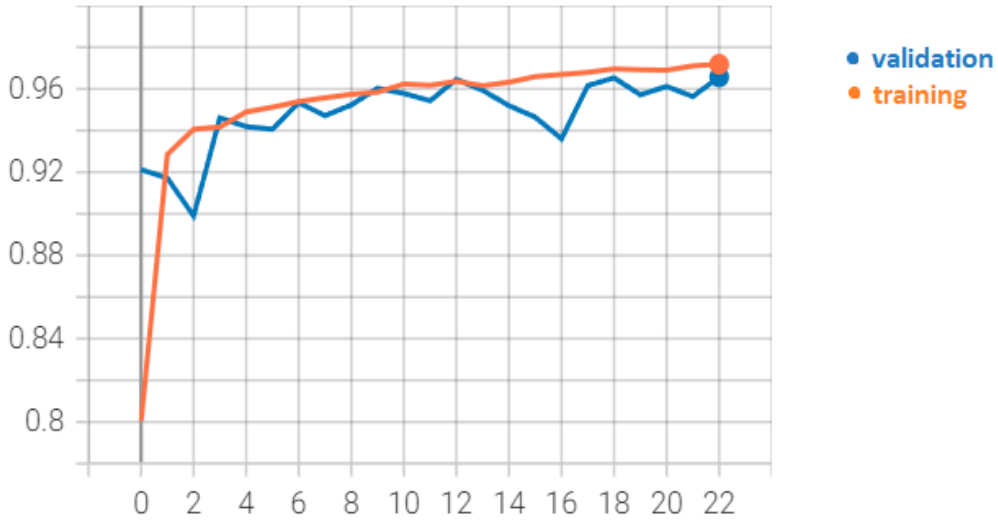

Figura 6.4: Acuratețea pe seturile de antrenare și validare obținute de modelul de clasificare

După cum se observă și în cele două figuri, acuratețea pe setul de antrenare crește destul de constant, în timp ce acuratețea pe setul de validare, care pornește cu o valoare mare din prima epocă, continuă să crească și să se îmbunătățească de la o epocă la alta, având anumite puncte în care tinde să scadă și să revină.

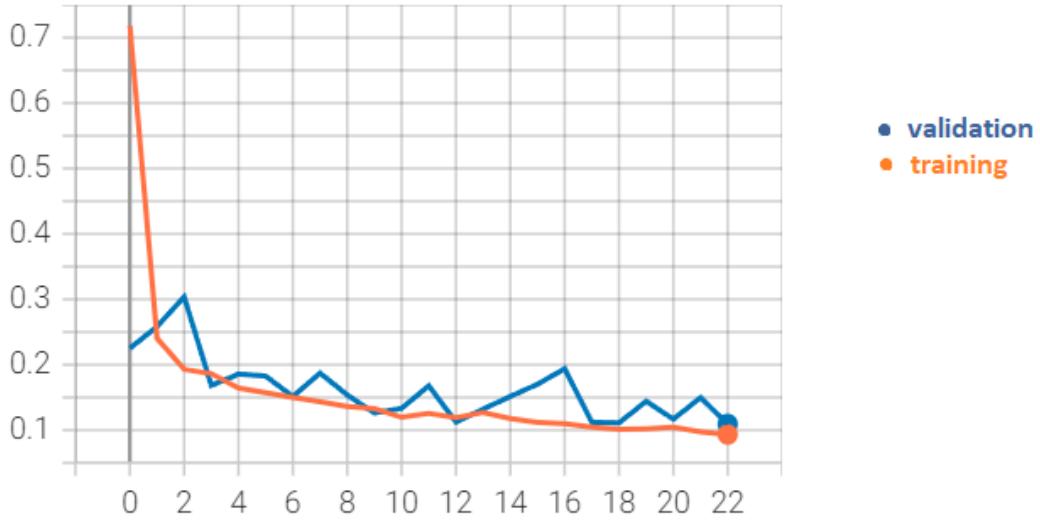

Figura 6.5: Loss-ul pe seturile de antrenare și validare obținute de modelul de clasificare

Valorile funcției de loss sunt optimizate folosind tehnica de regularizare reduce learning rate on plateau. Această tehnică urmărește tendințele funcției de loss și ajustează rata de învățare cu un anumit factor, dacă valoarea monitorizată nu se îmbunătățește. În scenariul de față, când cross-entropy loss nu scade într-un interval de 5 epoci va reduce rata de învățare cu un factor de 0.2.

Modelul a fost testat în clasificarea setului Kermany de 1000 de imagini și a obținut 99% acuratețe. Matricea de confuzie rezultată în urma predicției clasificării se află în figura 6.6.

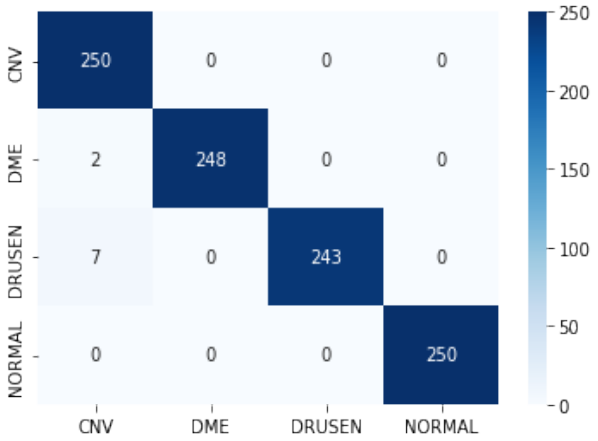

Figura 6.6: Matricea de confuzie rezultată în urma predicției pe setul Kermany

6.3. Evaluarea encoder-ului

6.3.1. Reprezentarea vizuală

În ceea ce privește reprezentarea de dimensionalitate redusă generată de encoder, am identificat anumite aspecte în imaginile obținute. Vectorii concatenați surprind niște

valori continue pe aceleași poziții din reprezentare, lucru care poate indica existența a două B-scans alăturate în volumul retinei, care prezintă aceleași trăsături. Mai mult decât atât, în zona vectorilor asociați imaginilor care prezintă afecțiune la nivelul retinei, valorile nu mai sunt continue, ci încep să surprindă alte trăsături, care pot fi reprezentative pentru afecțiunea respectivă. Aceste fluctuații în reprezentarea vizuală pot fi corelate și cu acuitatea vizuală asociată, deoarece o retină sănătoasă, cu o valoare mare a acuității nu prezintă schimbări semnificative în imaginile OCT la trecerea în volum de la un B-scan la altul. Câteva exemple relevante se află în figura 6.7.

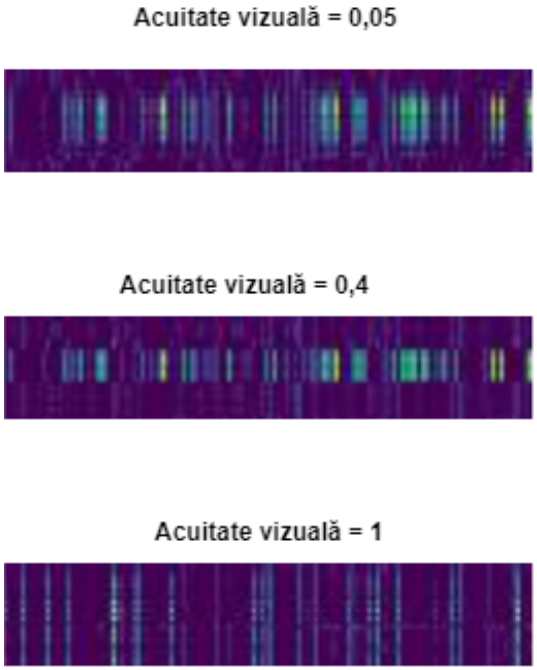

Figura 6.7: Reprezentări vizuale ale rezultatelor encoding-ului

Această analiză se poate efectua cu ajutorul metodelor puse la dispoziție în fișierul notebook destinat vizualizării imaginilor OCT, în care se pot trata comparativ volumele de imagini OCT corespunzătoare unei vizite a unui pacient, cu reprezentările rezultate de encoder.

6.3.2. **Predicția acuității**

O altă metodă de evaluare a calității encoder-ului este utilizarea reprezentărilor generate pentru predicția acuității vizuale. Cele două modele menționate în capitolul 4 (de clasificare și regresie) au fost testate folosind imaginile rezultate de encoder. Clasificatorul a fost antrenat pentru 100 de epoci cu un batch size de 32 de imagini, pentru a clasifica reprezentările imaginilor OCT în cele 12 clase corespondente valorilor acuității vizuale. Rezultatele antrenării sunt atașate în figura 6.8, iar matricea de confuzie se află în figura 6.10.

Fenomenul de overfit este prezent în acest scenariu. Acest lucru se poate observa cu ușurință și în graficul acurateței la antrenare, unde valorile obținute pe setul de validare stagnează în intervalul 35%-45%, în timp ce valorile pe setul de antrenare se apropie de 100%, iar valoarea loss-ului tinde să crească. Un principal motiv al apariției acestui fenomen este dezechilibrul în ceea ce privește numărul de instanțe asociate fiecărei clase.

Numărul redus de exemple îngreunează modelul deep learning, care pentru a învăța are nevoie de o cantitate mare de date. De asemenea, am amintit faptul că în acest set de date există fluctuații în valorile acuității vizuale, mai ales în rândul pacienților cărora li s-a administrat tratamentul injectabil pentru încetinirea evoluției DMLV.

Pentru a reduce apariția overfit-ului, am grupat setul de date în două clase, în scopul de a echilibra numărul de instanțe din fiecare clasă. Clasele alese au fost: acuitate cu valoare $\leq$0.1, respectiv >0.1. Această împărțire ar putea scoate în evidență pacienții cu formă avansată DMLV. Am păstrat hiper-parametrii de antrenare, iar rezultatele sunt ilustrate în figura 6.9. Overfit-ul s-a redus semnificativ, acuratețea pe setul de validare atingând valoari apropiate de 90%. Cu toate acestea, acuratețea maximă pe setul de test este de doar 53%, lucru care evidențiază faptul că predicția acuității folosind acest set de date nu este tratată eficient ca o problemă de clasificare.

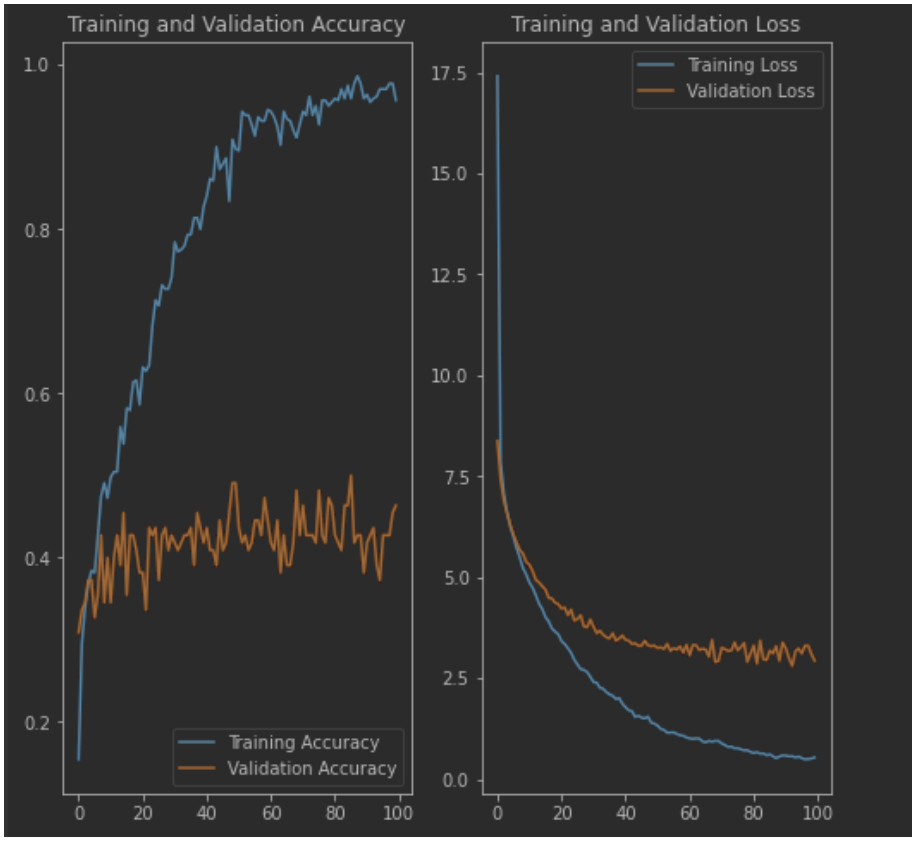

Figura 6.8: Acuratețea și loss-ul la antrenarea clasificatorului cu 12 clase

În matricea de confuzie asociată clasificării în 12 clase se poate observa cum pentru unele instanțe se prezice o valoare apropiată celei adevărate, motiv pentru care acest task poate fi adresat ca o problemă de regresie.

Algoritmul SVR a fost testat folosind datele OCT. Am ales să elimin vizitele pacienților ale căror acuitate vizuală a fost măsurată prin metoda cps - cu punct stenopeic, deoarece medici menționează faptul că în anumite cazuri îmbunătățește valoarea. Astfel, am ales din cei 80 de pacienți rămași 70 pentru a antrena modelul și 10 pentru test. Această metodă de împărțire asigură faptul că modelul nu primește în setul de test date similare cu cele pe care le-a întâlnit la antrenare.

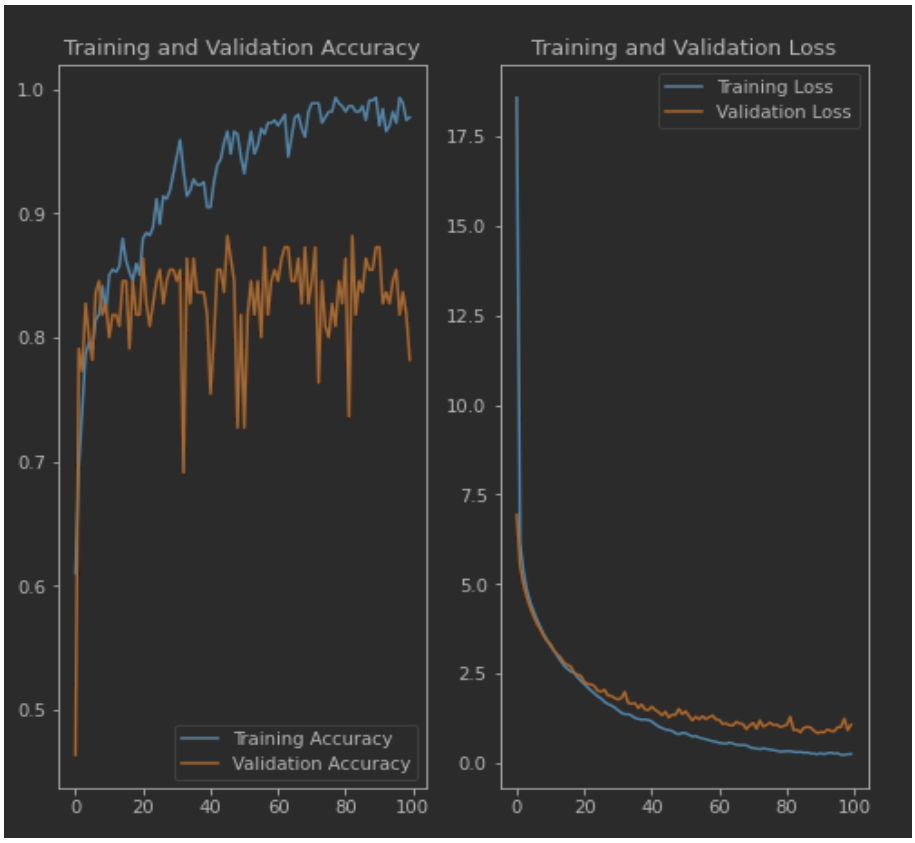

Figura 6.9: Acuratețea și loss-ul la antrenarea clasificatorului cu 2 clase

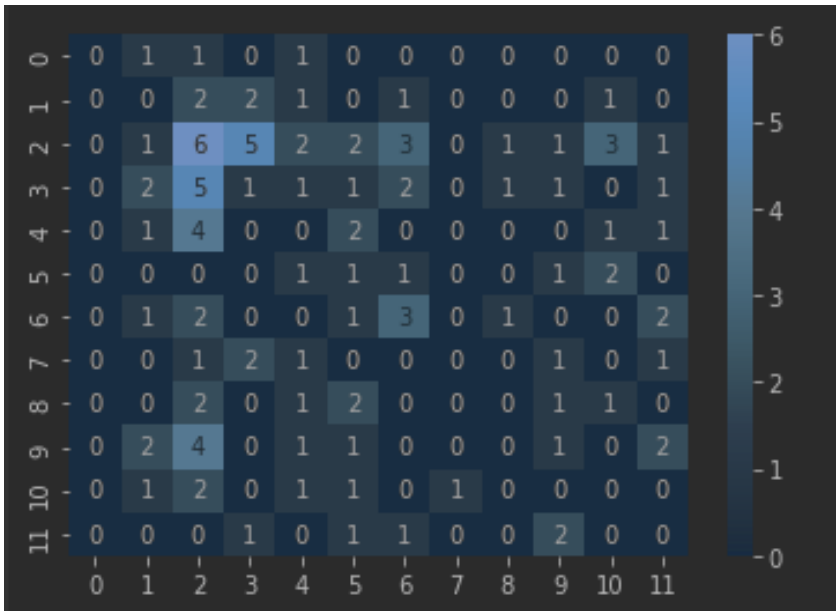

Figura 6.10: Matricea de confuzie a clasificării în cele 12 clase de acuitate - în ordine crescătoare

În urma mai multor teste, depinzând de alegerea la întâmplare a acestor pacienți, cele mai bune metrici obținute, care au valori impresionante, deoarece acuitatea se află în intervalul [0, 1], au fost:

- Root mean squared error: 0.18
- Mean absolute error: 0.12

O altă abordare care a dat rezultate mult mai bune a fost utilizarea modelului de regresie pentru a prezice valoarea acuității și folosirea rezultatului pentru a extrage clasa cea mai apropiată valorii reale. În urma calculului rezultatului regresiei, am considerat ca fiind clasa prezisă, valoarea cea mai apropiată a acuității de valoarea întreagă prezisă. Clasa a fost obținută prin calculul diferenței absolute dintre valoarea prezisă și cele 12 valori ale acuității. Am selectat valoarea cea mai mică a diferenței, iar clasa corespondentă a fost luată în considerare. De exemplu, pentru valoarea predicției 0.22, clasa aferentă este 0.2.

Astfel, se pot analiza și metricile specifice problemelor de clasificare, cum ar fi acuratețea și analiza matricii de confuzie (figura 6.11). Acuratețea maximă obținută este 23%, iar după cum se poate observa, în matricea de confuzie se regăsesc mult mai multe valori apropiate de cele reale, comparativ cu matricea obținută în urma clasificării cu rețea neuronală convoluțională.

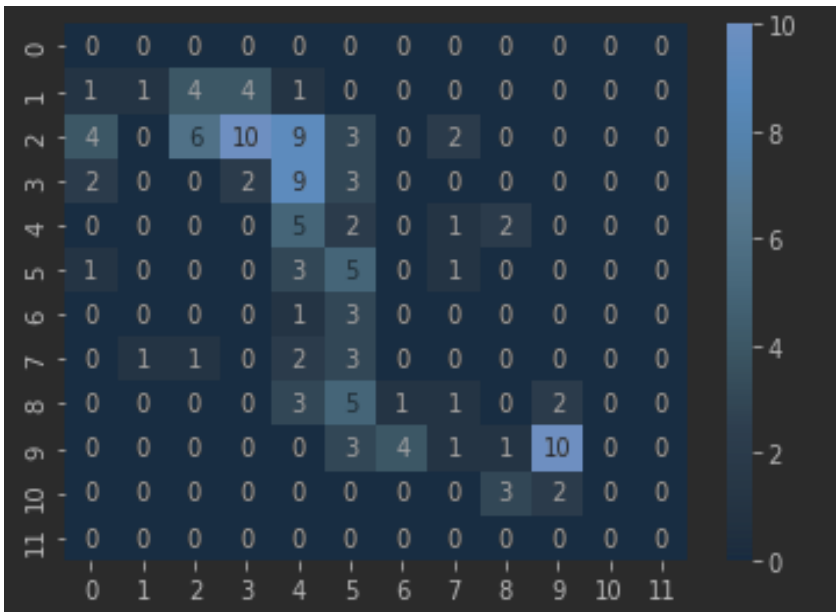

Figura 6.11: Matricea de confuzie obținută cu ajutorul modelului de regresie

Având aceste valori la dispoziție putem concluziona faptul că tratarea predicției ca o problemă de regresie aduce rezultate mult mai bune, comparativ cu problema de clasificare. Reprezentările obținute cu ajutorul encoder-ului contribuie la aceste rezultate, reușind să surprindă informații cheie din imaginile OCT. De asemenea, metodologia de antrenare supervised aduce un plus de performanță, ajutând modelul să extragă trăsături specifice domeniului problemei.

Capitolul 7. Manual de Instalare și Utilizare

Modelele detaliate în această lucrare au fost dezvoltate folosind limbajul Python, versiunea 3.8. Librăria Keras a fost utilizată pentru a antrena modelele, care reprezintă o interfață pentru biblioteca Tensorflow. Versiunea Tensorflow inclusă este tensorflow-gpu 2.8.0, care ajută la antrenarea modelelor folosind puterea computațională a unui GPU. Antrenarea a fost posibilă cu ajutorul platformei Docker, care ușurează dezvoltarea aplicațiilor, oferind resursele necesare fără a fi nevoie să fie instalate pe mașina locală.

Platforma pune la dispoziție sisteme software în variantă virtualizată, pe care programatorul le poate utiliza pe sistemul personal, prin niște pachete denumite containere. Container-ul pe care l-am creat include o imagine de tensorflow, mai exact **tensorflow:2.8.0-gpu**. Având la dispoziție această imagine, am reușit să realizez sistemul descris în capitolele anterioare.

7.1. Resurse necesare

Pentru a antrena modelele este nevoie de cele două seturi de date menționate în capitolul 5, setul public de imagini OCT Kermany, precum și setul de date de la Spitalul Clinic Județean de Urgență din Cluj-Napoca. Alte resurse includ:

- Resurse hardware:
  - Mașină care suportă limbajul de programare python și poate executa cod scris în acest limbaj, sau un mediu virtual cu aceste specificații
  - De preferat, sistemul să fie dotat cu unitate de procesare grafică sau GPU, pentru o mai bună paralelizare a operațiilor și o accelerare a procesului de antrenare

- Resurse software:
  - Versiunea 3.8 a limbajului **python**
  - Biblioteca publică **tensorflow**, care reprezintă platforma de antrenare a modelelor
  - Biblioteca publică **tensorflow-addons**, pentru a putea executa anumite funcții suplimentare folosite în antrenarea modelelor, care nu sunt incluse în librăria tensorflow
  - Biblioteca publică **matplotlib**, utilizată în reprezentarea grafică a rezultatelor antrenării și a evaluării modelelor
  - Biblioteca publică **pandas**, pentru vizualizarea metricilor rezultate în urma testării modelelor (ex. matricea de confuzie)
  - Biblioteca publică **sklearn**, care oferă funcții matematice utile și metrici de evaluare a modelelor (ex. confusion_matrix, classification_report, cosine_similarity)
  - Biblioteca publică **seaborn**, care permite reprezentarea vizuală a anumitor metrici (ex. confusion matrix)
  - Biblioteca publică **ipywidgets**, include anumite elemente grafice pentru vizualizarea volumelor de imagini OCT

7.2. Manual de utilizare

Această aplicație include un jupyter notebook, cu ajutorul cărora se pot vizualiza imaginile OCT din setul de date DMLV la nivel de volum, precum și reprezentările trăsăturilor aferente. Pentru asta, se efectuează următorii pași:

1. se pornește o conexiune jupyter notebook
2. se deschide fișierul **visualize_oct.ipynb**
3. se rulează toate celulele existente în fișier
4. pentru a vizualiza un anumit volum de imagini se apelează metoda **show_octs_and_encodings(patient_nr, visit_nr, eye)**, unde patient_nr este o valoare de tip int care reprezintă indexul pacientului, visit_nr este o valoare de tip int care reprezintă indexul vizitei pacientului, iar eye este un string, care poate lua valorile "OS" (ochi stâng) sau "OD" (ochi drept); aceste valori reprezintă informațiile vizitei care se dorește a fi vizualizată
5. prin mișcarea slider-ului, se poate naviga prin imaginile OCT aferente volumului pacientului

Evaluarea modelelor detaliate în această lucrare se poate face din terminal, apelând comanda **python main.py**. Se selectează modelul dorit introducând numele corespunzător în consolă. Opțiunile sunt:

- "kermany_classifier" pentru a evalua modelul de clasificare contrastive pe setul de test Kermany
- "acuity_classifier" pentru a evalua modelul de clasificare a acuității vizuale
- "acuity_regressor" pentru a evalua modelul de regresie a acuității vizuale

După selectarea modelului dorit, în consolă vor apărea rezultatele evaluării.

Capitolul 8. Concluzii

8.1. Contribuții proprii

În această lucrare am propus utilizarea contrastive learning ca metodă de reducere a dimensionalității imaginilor OCT pentru analiza afecțiunilor oftalmologice. Am pornit de la încercarea de a aplica metodologia transfer learning în domeniul medical, folosind imagini OCT pentru a rezolva un task de clasificare. Ulterior, am încercat să includ această paradigmă în analiza setului de imagini OCT provenit de la Spitalul Clinic Județean de Urgență din Cluj-Napoca. Încercarea nu a adus rezultatele dorite, datorită contextului diferit pe care acest set îl impune, și anume prezența volumelor de imagini.

Într-un scenariu real, alegera unei reprezentări compacte cât mai bune la nivel de volum nu este o sarcină ușor de realizat, având la dispoziție rețele neuronale convoluționale de clasificare, care întâmpină dificultăți în extragerea unor reprezentări care să cuprindă informațiile cele mai relevante rezolvării problemei.

Am abordat metodologia contrastive learning în urma studiului soluțiilor deep learning, care au apărut cu scopul de a analiza imagini din domeniul medical. Arhitecturile din literatură, în principal cele menționate în articolul [15], au adus rezultate impresionante, motiv pentru care am ales să implementez o astfel de soluție, bazată pe tehnica de învățare contrastive.

Având la dispoziție setul de date neetichetat, am început cu dezvoltarea unui model bazat pe învățare stil self-supervised, care nu necesită adnotări, însă rezultatele nu au fost mulțumitoare. Trecerea la învățarea în stil supervised a venit în urma analizei articolului [16], care evidențiază performanța impresionantă de care a dat dovadă arhitectura antrenată folosind funcția de loss SupCon, în stil supervised. Pentru a putea evalua performanța encoder-ului, care să extragă reprezentările de nivel redus ale imaginilor OCT, am ales să îl includ într-un task de clasificare pe setul de date Kermany, pentru care am obținut rezultate foarte bune.

Cu toate acestea, pentru problema predicției acuității, reprezentările obținute cu ajutorul encoder-ului nu au fost suficiente pentru ca o rețea neuronală clasică să reușească să surprindă corelația dintre acuitatea vizuală și imaginile OCT volumetrice. Pe de altă parte, un algoritm machine learning de regresie a oferit rezultate mult mai bune.

Lucrarea [14] prezintă o abordare similară, în ceea ce privește folosirea unui encoder pentru a extrage reprezentări de dimensionalitate redusă a imaginilor OCT afectate de DMLV. Totuși, metoda descrisă diferă prin modalitatea de învățare, care urmărește eroarea de reconstrucție în arhitectura autoencoder. În schimb, encoder-ul contrastive a oferit reprezentări eficiente în clasificarea setului Kermany și rezultate promițătoare pe datele DMLV prin învățare supervizată de tip contrastive.

Așadar, în ceea ce privește predicția acuității există loc de îmbunătățiri, însă reprezentările extrase prin metodologia contrastive learning sunt un pas promițător spre rezultate mai bune. Un alt aspect pe care această lucrare îl subliniază este nevoia de cantități mari de date în contextul deep learning.

Setul de date DMLV este o unealtă bună pentru dezvoltarea unui sistem care să analizeze imagini OCT la nivel de volum, însă prezintă multe inconsistențe care îngreunează algoritmii în învățare. Această lucrare este încă o dovadă a puterii de învățare a modelelor de tip contrastive, dar și a capacității lor de a îmbunătăți semnificativ modalitatea de reprezentare a datelor cu care sunt antrenate. Modelele detaliate pot fi utilizate mai departe cu scopul de a ajuta medicii oftalmologi în analiza afecțiunilor oftalmologice precum DMLV.

8.2. Dezvoltări ulterioare

Prima îmbunătățire care se poate aduce acestui sistem este fine-tuning-ul modelelor incluse. Modelul contrastive poate fi antrenat astfel încât să extragă o reprezentare mai compactă decât cea curentă (128 de trăsături), pentru a obține o reprezentare generală mai redusă, care să includă cele mai importante caracteristici ale imaginilor OCT.

Pentru analiza vectorilor de trăsături se pot utiliza abordări de tip clustering sau tehnici de reducere a dimensionalității precum Umap, astfel încât să se valideze existența unor trăsături comune la nivel de acuitate pentru un volum B-scan.

Russakoff et al. au punctat în articolul [24] faptul că preprocesarea imaginilor, astfel încât să fie segmentate semantic în funcție de straturile distincte ale retinei, a îmbunătățit semnificativ performanța arhitecturii AMDnet, motiv pentru care este o tehnică ce merită abordată pentru a obține predicția acuității.

În scopul de a obține un encoder care să fie cât mai bine modelat pe problema afecțiunii DMLV, ar fi util ca modelul să fie antrenat folosind ambele tehnici (supervised și self-supervised), care să includă și imagini care prezintă retine afectate de această condiție. Pentru a realiza acest scenariu, folosirea unui set de date care să cuprindă mai multe volume de imagini ar fi un avantaj.

În cadrul unui volum de imagini B-scan se pot identifica exemplele care suprind retina fără nicio afecțiune și se pot elimina din volum, pentru a face ca reprezentările să cuprindă doar informațiile relevante afecțiunii care se dorește a fi analizată. În acest mod se reduce semnficativ puterea de procesare necesară, deoarece se folosesc mai puține date.

Nu în ultimul rând, în ceea ce privește predicția acuității vizuale la nivel de volum de imagini, se pot încerca abordări de tip recurrent neural network, datorită faptului că tratează date secvențiale, ceea ce le face foarte potrivite în cadrul vizitelor unui pacient, care se desfășoară la anumite intervale de timp.

Bibliografie

[1] S. Aumann, S. Donner, J. Fischer, and F. Müller, "Optical coherence tomography (oct): Principle and technical realization," in *High Resolution Imaging in Microscopy and Ophthalmology: New Frontiers in Biomedical Optics*, J. F. Bille, Ed. Cham: Springer International Publishing, 2019, pp. 59–85. [Online]. Available: https://doi.org/10.1007/978-3-030-16638-0_3

[2] D. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," vol. 172, no. 5, 2018, pp. 1122—-1131. [Online]. Available: https://doi.org/10.1016/j.cell.2018.02.010

[3] E. Elyan, P. Vuttipittayamongkol, P. Johnston, K. Martin, K. McPherson, F. Moreno-García, C, C. Jayne, and S. M. Mostafa, Kamal, "Computer vision and machine learning for medical image analysis: recent advances, challenges, and way forward," pp. 24–45, 2022. [Online]. Available: http://dx.doi.org/10.20517/ais.2021.15

[4] A. M. Ismael and A. Şengür, "Deep learning approaches for covid-19 detection based on chest x-ray images," *Expert Systems with Applications*, vol. 164, p. 114054, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420308198

[5] R. Ranjbarzadeh, A. Bagherian Kasgari, S. Jafarzadeh Ghoushchi, S. Anari, M. Naseri, and M. Bendechache, "Brain tumor segmentation based on deep learning and an attention mechanism using mri multi-modalities brain images," *Scientific Reports*, vol. 11, no. 1, p. 10930, May 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-90428-8

[6] R. Reiazi, R. Paydar, A. A. Ardakani, and M. Etedadialiabadi, "Mammography lesion detection using faster r-cnn detector," *Computer Science and Information Technology*, 2018.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *CoRR*, vol. abs/2002.05709, 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[8] J. Li, G. Zhao, Y. Tao, P. Zhai, H. Chen, H. He, and T. Cai, "Multi-task contrastive learning for automatic ct and x-ray diagnosis of covid-19," *Pattern recognition*, vol. 114, pp. 107 848–107 848, Jun 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33518812

[9] E. Boslaugh, Sarah, "Snellen chart," July 2018. [Online]. Available: https://www.britannica.com/science/Snellen-chart.

[10] "Amd or age-related macular degeneration," July 2020, https://www.thailandmedical.news/news/amd-or-age-related-macular-degeneration-researchers-identify-vitrone

[11] M. G. Kawczynski, T. Bengtsson, J. Dai, J. J. Hopkins, S. S. Gao, and J. R. Willis, "Development of Deep Learning Models to Predict Best-Corrected Visual Acuity from Optical Coherence Tomography," *Translational Vision Science Technology*, vol. 9, no. 2, pp. 51–51, 09 2020. [Online]. Available: https://doi.org/10.1167/tvst.9.2.51

[12] T. M. Aslam, H. R. Zaki, S. Mahmood, Z. C. Ali, N. A. Ahmad, M. R. Thorell, and K. Balaskas, "Use of a neural net to model the impact of optical coherence tomography abnormalities on vision in age-related macular degeneration," *American Journal of Ophthalmology*, vol. 185, pp. 94–100, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0002939417304464

[13] K. Park, J. Kim, and J. Lee, "A deep learning approach to predict visual field using optical coherence tomography," *PLOS ONE*, vol. 15, no. 7, pp. 1–19, 07 2020. [Online]. Available: https://doi.org/10.1371/journal.pone.0234902

[14] E. Kando, "Visual acuity prediction based on feature extracted from optical coherence tomography using convolutional autoencoders," Lucrare de licență, 2021.

[15] S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, and M. Norouzi, "Big self-supervised models advance medical image classification," 2021. [Online]. Available: https://arxiv.org/abs/2101.05224

[16] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 18 661–18 673. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf

[17] A. Dertat, "Applied deep learning - part 3: Autoencoders," October 2017. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

[18] S. I. Berchuck, S. Mukherjee, and F. A. Medeiros, "Estimating rates of progression and predicting future visual fields in glaucoma using a deep variational autoencoder," *Scientific Reports*, vol. 9, no. 1, p. 18113, Dec 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-54653-6

[19] S. M. Waldstein, P. Seeböck, R. Donner, A. Sadeghipour, H. Bogunović, A. Osborne, and U. Schmidt-Erfurth, "Unbiased identification of novel subclinical imaging biomarkers using unsupervised deep learning," *Scientific Reports*, vol. 10, no. 1, p. 12954, Jul 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-69814-1

[20] D. Mwiti, "Transfer learning guide: A practical tutorial with examples for images and text in keras," January 2022. [Online]. Available: https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras

[21] Q. Yan, D. E. Weeks, H. Xin, A. Swaroop, E. Y. Chew, H. Huang, Y. Ding, and W. Chen, "Deep-learning-based prediction of late age-related macular degeneration

progression," *Nature Machine Intelligence*, vol. 2, no. 2, pp. 141–150, Feb 2020. [Online]. Available: https://doi.org/10.1038/s42256-020-0154-9

[22] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[23] P. M. Burlina, N. Joshi, M. Pekala, K. D. Pacheco, D. E. Freund, and N. M. Bressler, "Automated Grading of Age-Related Macular Degeneration From Color Fundus Images Using Deep Convolutional Neural Networks," *JAMA Ophthalmology*, vol. 135, no. 11, pp. 1170–1176, 11 2017. [Online]. Available: https://doi.org/10.1001/jamaophthalmol.2017.3782

[24] D. B. Russakoff, A. Lamin, J. D. Oakley, A. M. Dubis, and S. Sivaprasad, "Deep Learning for Prediction of AMD Progression: A Pilot Study," *Investigative Ophthalmology Visual Science*, vol. 60, no. 2, pp. 712–722, 02 2019. [Online]. Available: https://doi.org/10.1167/iovs.18-25325

[25] F. G. Venhuizen, B. van Ginneken, F. van Asten, M. J. J. P. van Grinsven, S. Fauser, C. B. Hoyng, T. Theelen, and C. I. Sánchez, "Automated Staging of Age-Related Macular Degeneration Using Optical Coherence Tomography," *Investigative Ophthalmology Visual Science*, vol. 58, no. 4, pp. 2318–2328, 04 2017. [Online]. Available: https://doi.org/10.1167/iovs.16-20541

[26] H. Bogunović, A. Montuoro, M. Baratsits, M. G. Karantonis, S. M. Waldstein, F. Schlanitz, and U. Schmidt-Erfurth, "Machine Learning of the Progression of Intermediate Age-Related Macular Degeneration Based on OCT Imaging," *Investigative Ophthalmology Visual Science*, vol. 58, no. 6, pp. BIO141–BIO150, 06 2017. [Online]. Available: https://doi.org/10.1167/iovs.17-21789

[27] E. Bursztein and O. Vallis, "Boost your model's accuracy using self-supervised learning with tensorflow similarity," February 2022. [Online]. Available: https://blog.tensorflow.org/2022/02/boost-your-models-accuracy.html

[28] R. A. H., "Convolutional neural networks (cnns)," 2019. [Online]. Available: https://anhreynolds.com/blogs/cnn.html

[29] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[30] W. Brian, "Contrastive loss explained," March 2020. [Online]. Available: https://towardsdatascience.com/contrastive-loss-explaned-159f2d4a87ec

[31] A. Chaudhary, "The illustrated simclr framework," 2020, https://amitness.com/2020/03/illustrated-simclr.