



Pázmány Péter Katolikus Egyetem  
Információs Technológiai és Bionikai Kar

Szakdolgozat

# **GPU alapú gépi tanulási módszerek az orvosi képfeldolgozásban**

Godó Ákos  
Molekuláris Bionika BSc

2015

Témavezető:  
dr. Cserey György

Alulírott Godó Ákos, a Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Karának hallgatója kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, és a szakdolgozatban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a szakdolgozatot más szakon még nem nyújtottam be.

---

Aláírás

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>6</b>
1.1. A feladatkiírás pontosítása . . . . .	7
<b>2. Az elméleti háttér bemutatása</b>	<b>8</b>
2.1. A diabetikus retinopátia és tünetei . . . . .	8
2.2. Képkalkotás a szemfenékről . . . . .	9
2.3. A felhasznált adathalmaz bemutatása . . . . .	13
2.4. Gépi tanulási módszerek . . . . .	14
2.5. Neurális hálózatok . . . . .	18
<b>3. Módszerek és munka bemutatása</b>	<b>27</b>
<b>4. Eredmények, értékelés</b>	<b>34</b>
4.1. Kitekintés . . . . .	37
<b>5. Összefoglalás</b>	<b>38</b>
<b>A. Mellékletek listája</b>	<b>42</b>

## Kivonat

A diabetikus retinopátia a látásvesztés legfőbb oka a munkaképes korú lakosságban, de megfelelő kezelésével nagy mértékben csökkenthető a látásvesztés esélye. Pontos diagnózis elengedhetetlen a kezeléséhez, bizonyos területeken viszont nem elérhető a megfelelő szaktudás és felszerelés a betegség felismerésére.

A betegség a tünetek alapján két fő kategóriára osztható, a proliferatív illetve a nem proliferatív retinopátiára, mely további három osztályra bontható a súlyossága alapján.

Jelen dolgozatomban bemutattam a diagnózisra használt szemfenék-fotográfia módszerét és a használatából fakadó lehetséges problémákat. Python és MATLAB szkriptekkel egy szabadon elérhető adathalmaz felhasználásával egy gépi tanulási módszer számára felhasználható tanuló- és tesztalmodrt hoztam létre, melynek lépéseit összefoglaltam. Megkíséréltem egy képfeldolgozási módszerrel a minták minőségét javítani.

Áttekintettem néhány, az orvosi képfeldolgozásban jelentős eredményeket elért módszer működését, mint a naiv bayesi klasszifikátor, a  $(k-)$ NN algoritmus, a tartóvektor-gép (SVM) és a neurális hálózatok. A választott módszerem, a neurális hálózatok működési elvét részletesebben is ismerttettem.

Egy ötosztályú klasszifikációs feladatot oldottam meg mély konvolúciós neurális hálózatok segítségével. Bemutattam a hálózat tervezésének elméleti és tapasztalati hátteret.

Caffe környezetben sikeresen létrehoztam egy 52,86% pontosságú hálózatot. Több hálózat együttműködésével pedig a maximális pontosságot megnöveltem 57,15%-ra. A dolgozat bemutatja az eredményekhez vezető tervezési folyamatot és elméleti megfontolásokat. Az eredmények alapján összefoglaltam a lehetséges továbbfejlesztési és felhasználási módokat.

## Abstract

Diabetic retinopathy is the leading cause of vision loss in working-age population but with proper treatment the chance of vision loss can be reduced considerably. The accurate diagnosis of the disease is essential for its treatment yet in several areas proper expertise and equipment is unavailable for the recognition of the disease.

Based on its symptoms the disease can be broken into two main categories, these being proliferative and non-proliferative diabetic retinopathy, which can be further classified into three separate categories based on its severity.

In this work I presented the method of fundus photography, a method used for the diagnosis of diabetic retinopathy and the possible issues stemming from its use. Using a publicly available dataset I created training and test sets for use with a machine learning method with Python and MATLAB scripts, the steps of which I have summarised. I attempted to increase the quality of the samples with an image processing method.

I surveyed several machine learning methods achieving substantial results in medical image processing such as the naive Bayesian classifier, the (k-)NN algorithm, the support vector machine (SVM) and neural networks. I presented the core principles of neural networks, my chosen method in more detail.

I solved a five-class classification problem using deep convolutional neural networks. I presented the theoretical and empirical background of the design process of such a network.

Using the Caffe environment I successfully created a network with an accuracy of 52,86%. With the co-operation of several such networks I increased the maximal accuracy to 57,15%. I showed the design process leading to these results and the necessary theoretical background. Based on the results I summarised the possible ways of upgrading and use of the system.

# 1. Bevezetés

A gépi tanulás fejlődésének mozgatórugói nagy mértékben az orvosi klasszifikációs problémák. A kórházak és egyéb egészségügyi intézmények modern eszközei roppant mennyiségű adatot termelnek viszonylag olcsón. Ennek az óriási adatmennyiségnek a kézzel történő feldolgozása nehézkes, szakértelmet és időt igénylő feladat. A gépi tanulási algoritmusok lehetővé teszik, hogy a szakértők által feldolgozott korlátozott mennyiségű és méretű adathalmazok alapján könnyedén és gyorsan feldolgozhassunk nagyságrendekkel több adatot. Mondhatni, hogy ezáltal a szaktudás válik hordozhatóvá és szabadon implementálhatóvá.

Bár a gépi tanulási algoritmusok még nem elfogadottak az orvosi diagnosztikában, napjainkban bizonyos alkalmazásokban képesek jobb eredményt elérni, mint egy ember. Ez természetesen nem jelenti azt, hogy bármilyen területen kiválthatják, esetleg felülmúlják az emberi ítélőképességet, de jelentős mennyiségű adat feldolgozásában nem elhanyagolható a teljesítményük.

Az orvosi adatokon tanított tanulási algoritmusok eredményére tekinthetünk úgy, mint az orvosi szaktudás sűrítményére, mely által könnyen kézbe adható a diagnosztika. Az orvosi képalkotási módszerek fejlődésének köszönhetően egyre könnyebben és költséghatékonyabban juthatunk hozzá a tanításra használható adathalmazokhoz. Továbbá ez a fejlődés lehetővé teszi azt, hogy egyre olcsóbban lehessen egy-egy személy diagnózisára használandó adatait megszerezni.

Ez motivált arra, hogy egy széles körben elterjedt és súlyos betegség tüneteinek felismerését próbáljam lehetővé tenni, hogy időben való kezelése könnyebben lehetségessé váljon. A diabetikus retinopátiára esett a választásom, mivel a diabéteszben szenvedők mind ki vannak téve a veszélyeinek. Non-invazív és olcsó diagnózisa olyan területeken is lehetővé teszi a betegség felismerését és kezelését, ahol eddig nem volt lehetséges.

Jelen dolgozat témája a diabetikus retinopátia súlyosságának klasszifikációja orvosi képalkotási adatokból. Az alábbiakban bemutatom a betegséget, tüneteit és egy módszert, mellyel a szemfenék struktúráiról alkothatunk képet. Ezek után áttekintek négy (naiv Bayes, (k-)NN, SVM és neurális hálózatok) releváns gépi tanulási módszert.

Az alábbiakban bemutatom továbbá, milyen adatokból építkezve hoztam létre a választott gépi tanulási módszeremhez, neurális hálózatokhoz használt tanító- és teszhalmazt, leírom a klasszifikáló rendszer tervezésének elméleti és tapasztalati megfontolásait. Nagy segítséget jelentett a modern grafikus processzorok, GPU-k könnyű elérhetősége, melyek lehetővé teszik a rendszer gyors működését. Az eredményeim összegzése és a munka során szerzett tapasztalataim alapján felvázolok lehetséges felhasználási területeket és továbbfejlesztési lehetőségeket.

## 1.1. A feladatkiírás pontosítása

- **Tekintse át az orvosi képfeldolgozásban kategorizálásra használt gépi tanulási módszereket:** ismerje meg és mutassa be a szakirodalomban orvosi képadatok klasszifikációjára használt és jó eredményeket elért módszerek működését. Válasszon egy, a feladat végrehajtására alkalmas módszert, melyet bővebben bemutat.
- **Válasszon egy releváns orvosi képfeldolgozást igénylő klasszifikációs problémát:** ismertesse a problémát, írja le a képkeltési módszer működését és mutasson be példákat a létrejött képekről.
- **Hozzon létre a rendelkezésre álló adatokból a tanuláshoz, a teszteléshez és a validációhoz szükséges adatszettet:** mutassa be a rendelkezésre álló adatokat és az adatszettek létrehozásának lépéseit.
- **Tervezzon és implementáljon gépi tanulási módszert használó osztályozó eljárást GPU-n:** az áttekintett gépi tanulási módszerek közül a legalkalmasabbnak választott eljárást használva ismertesse a tervezés lépéseit.
- **Mutassa be a létrehozott rendszeren egy tanítási folyamat működését és eredményeit. Összegezze tapasztalatait:** írja le a használt módszer eredményeit a beállítások függvényében. Vonjon le következtetéseket az eredményekből.
- **Foglalja össze a jövőbeli továbbfejlesztési és felhasználási lehetőségeket:** a kapott eredmények alapján szerzett tapasztalatokból fogalmazzon meg lehetséges javítási lehetőségeket a tervezett rendszerben. Vessen fel olyan alkalmazási területeket, ahol az eljárás felhasználása hasznos lehet.

## 2. Az elméleti háttér bemutatása

### 2.1. A diabetikus retinopátia és tünetei

A diabetes mellitus korunk egyik legelterjedtebb betegsége. A diabéteszben szenvedők száma világszerte növekszik, a 2013-ban ismert 382 millióról várhatóan 592 millióra növekszik majd az esetek száma 2035-re [1]. A cukorbetegség során fellépő hiperglikémia, a magas vércukorszint, következtében sérülnek mind a kis és nagy vérerek illetve az idegsejtek, így súlyos károk keletkeznek a belső szervekben, beleértve a veséket, a szívet, az agyat és a szemeket. A szemben fellépő komplikáció az úgynevezett diabetikus retinopátia. A diabetikus retinopátia a legfőbb oka a munkaképes korú népesség körében a vakságnak [2], de megfelelő kezelés mellett a súlyos látásvesztés valószínűsége 90 százalékkal csökkenthető.

Az állapotot a cukorbetegség következtében megsérülő vérerek okozzák. A magas vércukorszint miatt az érfalban levő periciták elhalnak, gyengül az érfal. A pericitákban a hiperglykémia egy intracelluláris jelátviteli útvonalat aktivál, melynek működése a sejtek apoptózisához vezet [3]. Ezáltal megnő az erek átteresztőképessége, a sérült érfalú vagy kifakadó erekből szivárgó vér és folyadék kárt okoz a retinában és látásromláshoz, látásvesztéshez vezet. A diabetikus retinopátiának két fő formája létezik: a nem proliferatív diabetikus retinopátia (NPDR) illetve a proliferatív diabetikus retinopátia (PDR) [4].

Az NPDR-nek három fokozata van, az enyhe NPDR során csak mikroaneurizmák, duzzanatok észrevehetőek a retina hajszálerein. Súlyos NPDR-ről beszélhetünk, ha az alábbiak közül akár egy feltétel teljesül (és nincsenek PDR-tünetek):

1. több, mint 20 intraretinális bevérzés mind a négy kvadránsban,
2. egyértelmű értágulat több, mint két kvadránsban,
3. feltűnő intraretinális elválkozás legalább egy kvadránsban.

A mérsékelt NPDR tünetei az enyhe és a súlyos NPDR között helyezkednek el, egyértelműen súlyosabbak az enyhe változaténál, de mérsékeltebbek, mint a súlyoséi [4]. NPDR esetén látásvesztést okozhat az úgynevezett makuláris ödéma, a szem sárgafoltjában jelentkező folyadékfelgyülemelés (melyet az erekből szivárgó vér és egyéb folyadék, köztük lipidek okoznak). A sárgafolt az éleslátás helye a szemben, az ide szivárgó folyadék kárt okoz a fotoreceptorokban, így okozva a látás elmosódását [5].



A PDR során neovaszkuarizáció, érképződés játszódik le, az új, a cukorbetegség miatt sérült erek kilépnek a retinából az üvegtestbe és ott történik a bevérzés. Súlyos NPDR-ről beszélhetünk ha fellép az alábbi tünetek legalább egyike [4]:

1. neovaszkuarizáció,
2. preretinális vagy üvegtestbeli bevérzés.

A PDR súlyos látásvesztést és vakságot is okozhat. A retina rossz vérellátása következtében történő abnormális neovaszkuarizáció során keletkező, a magas vércukorszint miatt elgyengült erek behatolhatnak az üvegtestbe. A már az üvegtestben lévő friss hajszálerekből szivárgó vér a bejövő fény útjába áll és kitakarja a látómező részeit. Az üvegtest a hegesedés miatt össze is húzódhat, mely súlyos esetben retinaleváláshoz is vezethet.

A diabetikus retinopátia következtében bekövetkező látásromlást a 2.1, 2.2 és 2.3 ábrák szemléltetik.



2.1. ábra. Tisztán látott kép. Forrás: [6].



2.2. ábra. A nem proliferatív diabetikus retinopátia következtében homályosan látott kép. Forrás: [6].

## 2.2. Képképzés a szemfenékről

A retináról közvetlen tudunk képet alkotni, mivel mind a beeső, mind a visszaverődő fénysugarak a pupillán keresztül haladnak. Kihívást jelent viszont a szemfenék és ese-



2.3. ábra. A proliferatív diabetikus retinopátia miatti látásvesztés. Forrás: [6].

tünkben a retina hajszálereiről megfelelő minőségű képek alkotása. Ideális esetben a megfigyelő szeme és a megfigyelt szem egymásnak szemben állnak és párhuzamos fénysugarak fókuszálódnak a megfigyelő retináján.

Viszont a szemfenék megfigyelése csak akkor lehetséges, ha az megfelelően megvilágított - a szem optikájának szerkezete miatt a teljes fundus nem lesz megvilágítva, csak az a része, amelyre az alkotott kép beeső fénysugarai esnek. Tehát ha a megfigyelő a megfigyelt szem fundusát szeretné látni, akkor a fényforrásnak (a visszatükröződés kiindulóhelyének) egy vonalba kell esnie a szemfenékkal. Tehát csak a megfigyelendő és a megvilágított terület metszete a valóban megfigyelhető. Általában ez a két terület nincs fedésben, ezért látjuk mások pupilláját feketének.

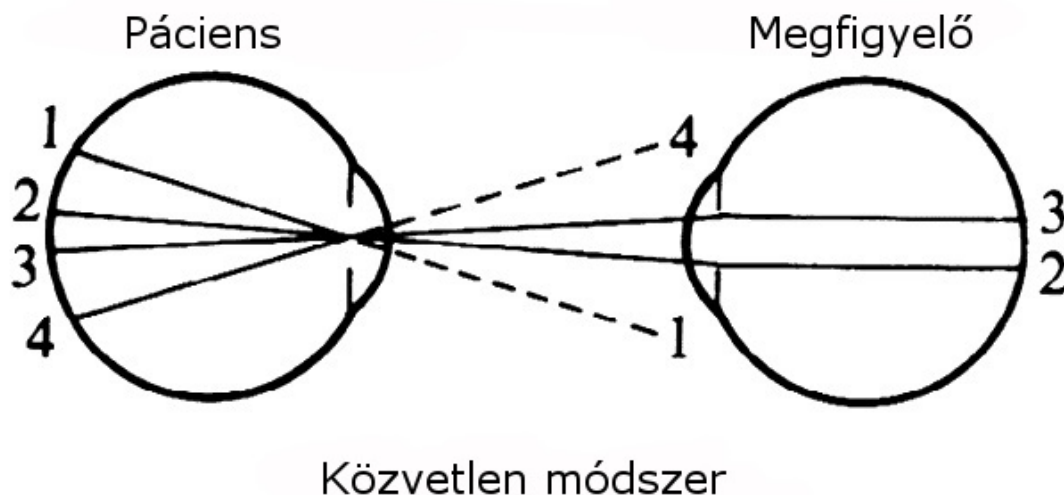
A szem belsejéről képet legegyszerűbben a *red reflex* felhasználásával kaphatunk. A szembe bemenő fény egy része elnyelődik, a visszatükröződő fény pedig vöröses-narancssárga színű, innen kapta a nevét a jelenség. Ahhoz, hogy megfelelő minőségű képet kapjunk szükséges egyrészt a fény szembe juttatása oly módon, hogy a valós forrása ne a megfigyelő fejével egy vonalban legyen, másrészt pedig a kép nagyítása, hogy azon a szemfenék kisméretű struktúrái is jól láthatóak legyenek [5].

Ennek a problémának a megoldására szolgál az ophtalmoszkóp. Az eszköz a német Hermann von Helmholtz nevéhez köthető. Egy tükrös és lencserendszer segítségével fókuszál fényt a megfigyelendő szembe oly módon, hogy az megfelelően megvilágításba kerüljön. A tükrök és lencsék segítségével a megfigyelő már diagnosztikai célokra megfelelő minőségű és nagyítású képet lát [7].

További problémát okozhat, ha megfigyelt szem nem emmetrop. Emmetrop egy szem akkor, ha a szemmozgató izmok nyugalmi állapotban vannak és a szemlencsén áthaladó fénysugarak a sárgafolton fókuszálódnak. Ametropia esetében a fénysugarak nem a sárgafolton kereszteződnek. Myopia esetén a retina előtt kereszteződnek, hypermetropia esetén pedig a retina mögött található az elképzelt találkozási pont. Fontos tehát a megfelelő minőségű képek létrejöttéhez a képalkotáshoz használt fénysugarak fókuszálása is, úgy a bemenő fénysugarakra, mint a visszatükrözőttekre, mivel a szét- vagy összetartó sugarak nagy mértékben ronthatják a kapott kép minőségét. Ha a kép minősége a diagnózis számára nem megfelelő a vizsgált személy közel- vagy távollátása miatt, a megoldás, hogy az ophtalmoszkópos vizsgálat során viselje a szemüvegét. A szemüveg viselése miatt a kapott képen megjelenhetnek további reflexek a fény útjába helyezett

lencsék miatt [8].

Az így megfigyelhető terület méretét az eszköz látószöge határozza meg. Ezt az  $\alpha$  szöget a szembe merőlegesen beeső fénysugárhoz tartozó ponttól legtávolabb levő, még megfigyelhető pont határozza meg. A közvetlen ophthalmoszkóp esetében fénysugarak közvetlenül, eltérülés nélkül haladnak, így a szöget a merőlegesen beeső fénysugár és a szemfenék belső felületének metszéspontjától legtávolabbi megfigyelhető pont beeső fénysugara zárják be. Ezt szemlélteti a 2.4 ábra.

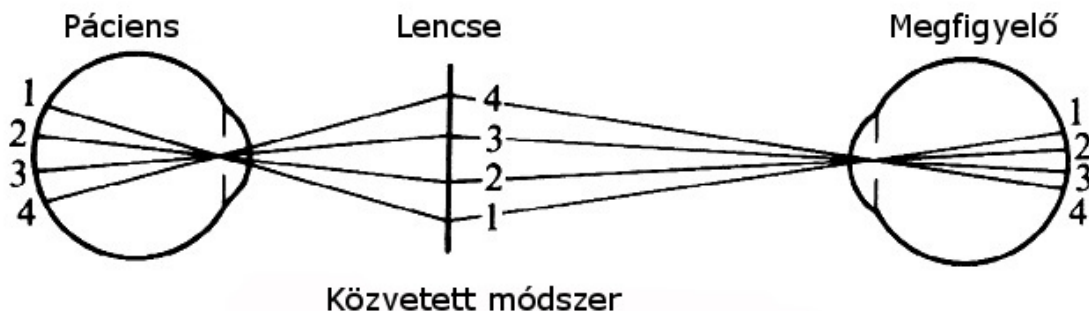


2.4. ábra. A fény útja a közvetlen ophthalmoszkóp használatakor. Bal oldalon a megfigyelt, jobb oldalon a megfigyelő szeme látható. Szerkesztve. Forrás: [9].

Közvetett ophthalmoszkóp használatakor a megfigyelt szemből kijövő széttartó fénysugarakat egy lencse segítségével újrafókuszáljuk úgy, hogy azok összetartva a megfigyelő szemébe jussanak. Ezáltal olyan fénysugarakat is összegyűjthetünk, melyek közvetlen ophthalmoszkópia esetében már a látómezőn kívüliek lettek volna, vagyis a megfigyelő és a megfigyelt szem pupillái által eltakart pontokhoz tartoztak volna. A lencse használatának következtében a kapott kép fejjel lefelé fordul. Ez a módszer látható a 2.5 ábrán. A kép a valósághoz képest kettő- és ötszörös nagyítású lesz [7].

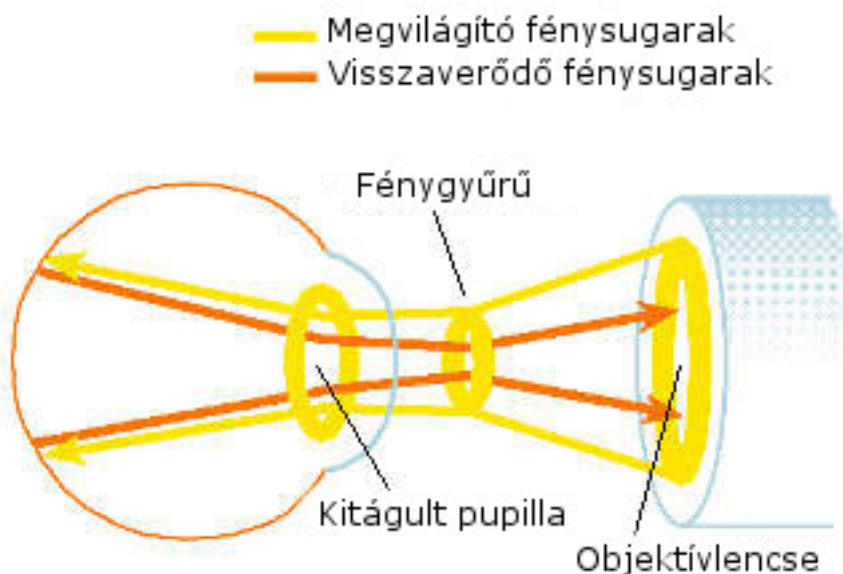
Az így kapott képek rögzítésére és ezáltal a fentebb leírt tünetek könnyebb diagnosztizálásra szolgál az úgynevezett szemfenék-fotográfia, mely a szemfenék, a fundus, félig átlátszó szöveteiről visszatükröződő fény segítségével alkot képet. A módszer a szemfenék háromdimenziós struktúrájáról készít kettő dimenziós képet. A létrejött kép intenzitása a visszaverődő fénysugarak intenzitásától függ.

A funduskamera, másik nevén réslámpás kamera, egy alacsony nagyítású mikroszkóp és egy fényképezőgép összessége, működési elve a közvetett ophthalmoszkópéra épül. A funduskamerák átlagos látószöge 30 fokos, a széles látószögűek 45 és 140 fok közötti látószöggel rendelkeznek, a szűk látószögűek pedig 20 fok alatti látószögűek. A szemfenéket megvilágító fény gyűrű alakú (mint azt a 2.6 ábra mutatja), a gépben szűrők során halad át, mielőtt kilép a réslámpás kamera lencséjén és a szaruhártyára vetül. A különböző



2.5. ábra. A fény útja a közvetett optalmoszkóp használatakor. Bal oldalon a megfigyelt, jobb oldalon a megfigyelő szeme látható, középen pedig egy gyűjtőlencse helyezkedik el. Szerkesztve. Forrás: [9].

hullámhosszú szűrők fontosak a különböző anatómiai struktúrák kihangsúlyozásában a szemfenékről alkotott képeken. Amennyiben a megvilágítás és a fókusz megfelelő, a visszatükröződő fény a bejövő fény gyűrűjén belül lép ki a szemből, majd (digitális képet rögzítő réslámpás kamera esetében) vetül a szenzorra [10].



2.6. ábra. A megvilágító és a visszatükrözőtt fénysugarak útja réslámpás kamera használatakor. Bal oldalon található a vizsgált szem, jobb oldalon pedig az objektív. Kívül haladnak a megvilágító sugarak, belül pedig az alkotott kép sugarai. Szerkesztve. Forrás: [11].

Monokromatikus képalkotáskor a megfelelő szűrő kiválasztásával hatékonyan emelhetünk ki elemeket a szemfenék alkotórészei közül.

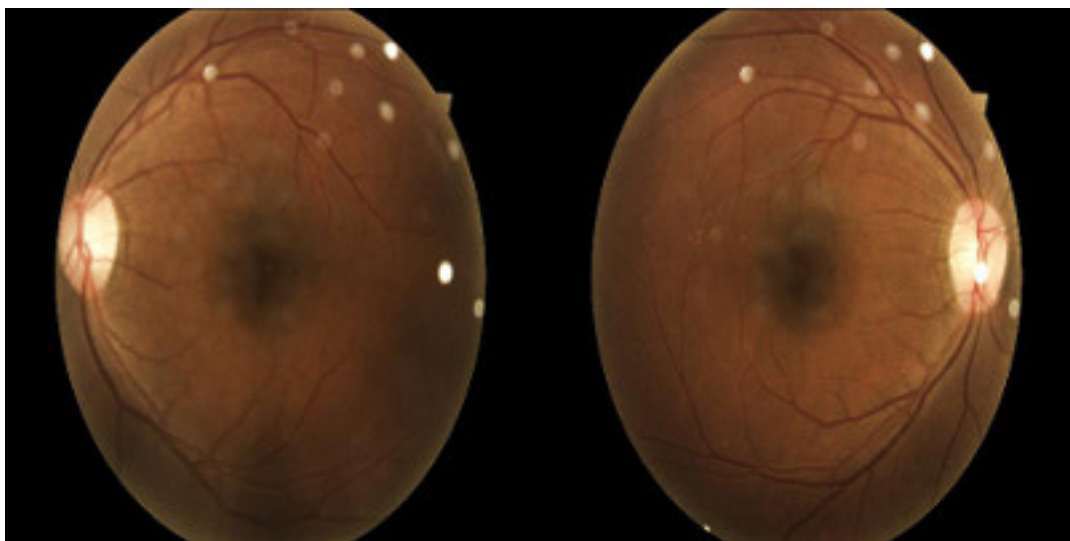
Kék fénnel a retina elülső rétegeit emelhetjük ki. Ez különösen hasznos, mivel fehér fénnel való megvilágításkor ezek a rétegek szinte teljesen átlátszóak. A kék fényt elnyeli a retina erezte és pigmentációja, így ezek sötétben jelennek meg a kapott képen. A belső határoló réteg, az idegrostok és egyéb részben átlátszó struktúrák, mint a ciszták kiemelődnek alacsony hullámhosszú megvilágítottságnál. A kép minőségét ronthatja,

hogy a szem belsejében az alacsony hullámhosszú fény jobban szóródik, mint a magasabb hullámhosszú fény, így a kapott kép elmosott lesz [12].

A zöld fényt szintén elnyelik a retina erei, de részben visszaverődik a pigmentekről. Ennek a ténynek köszönhetően a kapott képen a kontraszt magas. Könnyen megfigyelhetővé válik az erezet és a bevérvések is jól láthatóakká válnak. Ez a zöld fénnel való megvilágítás a legalkalmasabb a diabetikus retinopátia megfigyelésére, mivel pont azokat a struktúrákat emeli ki, melyek a tüneteket mutathatják [12]. A kép emberi szemmel való feldolgozásának könnyedségében az is szerepet játszik, hogy világosban az emberi szem spektrális érzékenysége maximuma 555 nanométernél van [13], mely a zöld fénynek felel meg.

Az érhártya kiemelésére alkalmas a vörös fénnel való képalkotás. A szemfenék kontrasztja jóval alacsonyabb így, mivel a szemfenék struktúráinak nagyrésze vörös színű [12].

Színes szemfenékképeket úgy kaphatunk, hogy három képet alkotunk, a szenzor beállításainak megfelelő frekvenciájú piros, zöld illetve kék fénnel, melyeket a gép a benne található szűrők segítségével állít elő, a kapott képeket pedig egy RGB színkódolású kép egy-egy csatornájának tekintjük. Egészséges szemekről készült képek láthatóak a 2.7 ábrán.



2.7. ábra. Egy egészséges szempárról készített szemfenékkép. Jól kivehetőek a jobb felső negyedekben található műtermékek. A műtermékek felismerését segíti, hogy az azonos kamerabeállítással készített képeken ugyanott helyezkednek el.

### 2.3. A felhasznált adathalmaz bemutatása

A cél a diabetikus retinopátia előrehaladásának klasszifikációja réslámpás kamerával készült képek alapján. Ehhez szükséges megfelelő mennyiségű és minőségű felcímkézett mintát találni, hogy egy osztályozó rendszer ezekből képes legyen még számára ismeretlen adatokat osztályozni. Megfelelő adathalmaz keresése során a Kaggle Diabetic Retinopathy Challenge-hez [14] biztosított adatokat használtam.

A diabetikus retinopátia fentebb leírt tünetei alapján a mintákat öt osztályba sorolhatjuk, melyek a betegség négy fázisa (enyhe, mérsékelt, súlyos NPDR, illetve PDR) továbbá amikor a szem nem mutatja a retinopátia jeleit. A minták címkéi az adathalmazban az alábbiaknak felelnek meg:

- 0, ha nincsenek diabetikus retinopátiára utaló jelek a képen;
- 1, ha az enyhe NPDR jelei vannak jelen;
- 2, ha mérsékelt NPDR-ra utaló jeleket mutat;
- 3, ha súlyos NPDR jeleit mutatja;
- 4, ha PDR-ban szenved.

Az adathalmaz 35126 felcímkézett képet tartalmaz, egy-egy páciens jobb és bal szeméről készített retinaképeket. A tünetek alapján kategorizált képek eloszlása a 2.1 táblázatban látható.

Kategória	Képek száma	Százalékos eloszlás
Nincs DR (0)	25810	73,48%
Enyhe NPDR (1)	2443	6,96%
Mérsékelt NPDR (2)	5292	15,07%
Súlyos NPDR (3)	873	2,48%
Proliferatív DR (4)	708	2,01%

2.1. táblázat. A Kaggle Diabetic Retinopathy Challenge-hez biztosított címkékkel ellátott adatszett tartalma.

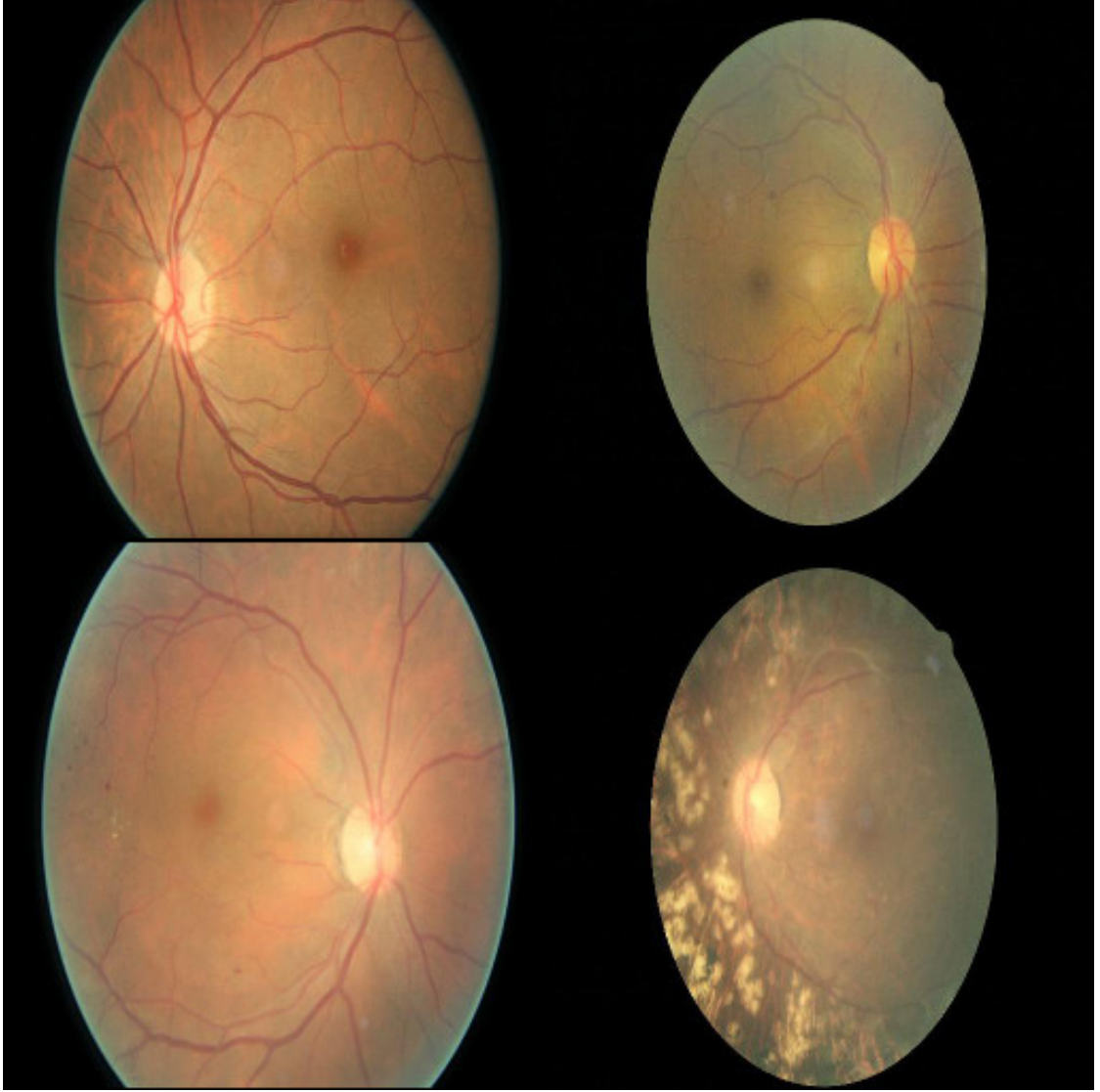
Elérhető volt további 53576 címke nélküli kép is, melyek osztályozása volt a feladat a Diabetic Retinopathy Challenge-ben. Ezek számomra a feladat szempontjából elhanyagolhatóak voltak. Az adathalmazban található képekre példaként szolgálnak a 2.7 és 2.8 ábrák képei.

## 2.4. Gépi tanulási módszerek

Mivel rendelkezésre áll egy címkékkel ellátott adathalmaz, így felügyelt gépi tanulási módszereket tekintettem át a probléma megoldására. A felügyelt gépi tanulási módszerek egy tanulási fázis után, melyek a címkézett minták megléte miatt „felügyeltek”, képessé válnak még nem látott adatok klasszifikációjára. Előnyös ezen módszerek használatakor egy tesztalmazt is létrehozni, mely szintén tartalmazza az elvárt kategóriák címkéit a benne levő mintákra. A tesztalmaznak köszönhetően könnyen ellenőrizhető a hálózat teljesítménye, akár tanítás közben is.

A gépi tanulási módszerek bizonyos esetekben képesek felülmúlni az emberi teljesítőképeséget. Egy 1971-es kutatásban [15] 472 hasi fájdalomra panaszkodó pácienszt diagnosztizált hat orvos, egy hatszáz páciens adataival tanított gépi tanulási módszer és egy további gépi tanulási módszer szakértői véleményekből és diagnózisokból álló hibrid





2.8. ábra. Példák az adathalmazban található képekre. Bal felül: enyhe NPDR (1). Jobb felül: mérsékelt NPDR (2). Bal alul: súlyos NPDR(3). Jobb alul: PDR (4).

adattázból tanítva. A legjobban teljesítő orvos általi diagnózisok pontossága 79,7% volt, a hibrid adattázzsal tanított rendszeré 82,2%, a 600 páciens adatain tanítotté pedig 91,1%. Ez alapján feltételezhető, hogy az emberek nem képesek pontosan ad hoc elemezni, valószínűségeket és eloszlásokat rendelni nagy adathalmazokhoz.

A feladat megoldásához egy erre alkalmas gépi tanulási módszert kellett választanom. Az első módszer, melynek használatát megfontoltam, a naiv bayesi klasszifikátor volt. Ez a megközelítés a minták előzetes eloszlásából alkotott hovatartozási valószínűségekkal dolgozik. A minta egy adott  $C$  osztályba tartozását, vagyis az osztály valószínűségét, az ahhoz tartozó leírók, *feature*-ök,  $(F_1, F_2, \dots, F_i)$  segítségével a Bayes-tétel alapján becsli meg [16]. Ez látható a 2.1 egyenletben.

$$p(C|F_1, F_2, \dots, F_i) = \frac{p(C) p(F_1, F_2, \dots, F_i|C)}{p(F_1, F_2, \dots, F_i)} \quad (2.1)$$

A naiv bayesi klasszifikátor a  $C$  osztályba tartozás feltételes valószínűségét  $(F_1, F_2, \dots, F_i)$

leíróktól függően a Bayes-tétellel becsli meg (*a posteriori* valószínűség). A  $p(C)$  valószínűség az előzetes, *a priori* valószínűség. Az *a posteriori* valószínűséget maximalizálja a módszer.

A tört nevezője elhanyagolható gyakorlati esetekben, mivel egy minta adott leíróra  $p(F_1, F_2, \dots, F_i)$  konstans. Tehát a bayesi klasszifikátorok  $p(C) p(F_1, F_2, \dots, F_i|C)$ -t maximalizálják a klasszifikálandó mintákra.

A módszer azért naiv, mert az egyes leírókat függetlennek tekinti. Kis mennyiségű mintával is képes nagy pontosságot elérni, viszont ehhez sok információ szükséges az adathalmaz elemeiről. Ezért ideális orvosi adatok klasszifikációjára, mivel a kórházak és egyéb egészségügyi intézményekben rengeteg adat halmozódik fel páciensenként, így akár ezekből, a magukra a mintákra nézve metaadatokból, is jelentős információtartalom nyerhető ki. Ilyen adat lehet például az életkor, vérnyomás, stb., melyek nem szerepelnek egy esetben egy szemfenékről alkotott képen. Épp ezért jelentős eredményeket ért el ez a módszer ezen a területen.

Mellrák túlélési ráta predikciójakor a naiv bayesi klasszifikátor  $84,5 \pm 1,4\%$ -os pontosságot ért el, közel azonosat, mint a többi megközelítés [17]. Egy másik kutatásban pedig  $67,3\%$ -os pontossággal diagnosztizált újszülöttekben szívbetegséget [18].

Egy másik lehetséges módszer a legközelebbi szomszéd, vagyis *nearest neighbour* (NN) algoritmus. Az NN-módszer eltárolja az egyes osztályokhoz tartozó leírókat és a klasszifikálandó mintát a rá vonatkozó leíróktól legkisebb távolságra levő tanítóminta osztályához rendeli. A módszer használatából eredő problémák közé tartozik az az eset, amikor az osztályok leírói közel vannak egymáshoz a Hamming-távolságuk alapján (vagyis csak néhány bitben térnek el). Ekkor az osztályok nehezen elkülöníthetők [19].

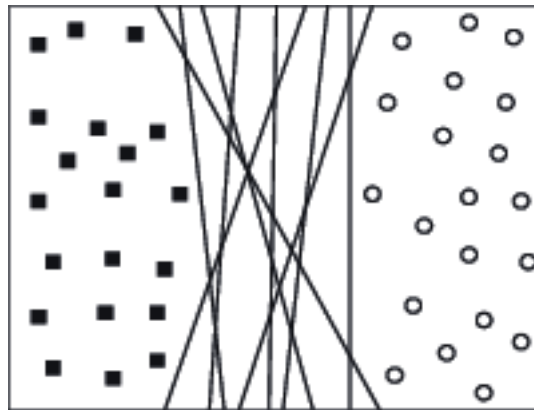
A hibaráta csökkentéséhez intuitívnak tűnhet az, hogy egyre több tanítómintát vagy leírót kell eltárolni. Ahogy egyre több leírót használunk, növekszik a döntési tér dimenziószáma, ezáltal az egyes adatpontok távolsága nő, a halmazok egyre ritkábbak lesznek. A dimenziók számával exponenciálisan nő a pontok távolsága, ez a jelenség a „dimenziónalitás átka”. Hogy egy NN-alapú osztályzó hatásosan működjön a szomszédos pontok távolságának egy, a problémától függő  $d$  értéknél kevesebbnek kell lennie. Példaként egy dimenzióban (egy leíróval) átlagosan  $\frac{n}{d}$  szükséges, hogy ez teljesüljön, ha a leíró a  $[0, 1]$  intervallumból vehet fel értékeket. Ezáltal az egyes pontok távolsága nem lesz nagyobb, mint  $\frac{1}{n}$ . Ha  $p$  leírónk van, akkor  $\frac{1}{d^p}$  pontra lesz szükségünk, hogy ezt a sűrűséget fenntartsuk. Amennyiben egy dimenzióban 10 ponttal fedtük le a  $[0, 1]$  intervallumot,  $p$  dimenzióban ennek a kiterjesztett térnek a lefedésére ugyanilyen sűrűséggel  $10^p$  pontra lesz szükségünk [20]. Kérdéses továbbá, hogy mekkora súllyal számítanak az egyes leírók. Egy lehetőség a sűrűségigény csökkentésére, hogy az osztályok középpontjához közel levő pontokat eltávolítjuk a halmazból. Ha sikerül megtalálnunk az adatok szempontjából fontos leírókat, nagyban csökkenthetjük az ideális működéshez szükséges pontok számát. A megfelelő metrika kiválasztásával és a leírók fontosságának súlyozásával számos probléma megoldására képes lehet az NN-módszer [19].

A nearest neighbour algoritmus teljesítménye javítható azáltal, hogy az egyetlen legközelebbi szomszéd osztályába való azonnali besorolás helyett több szomszédot vizsgálunk

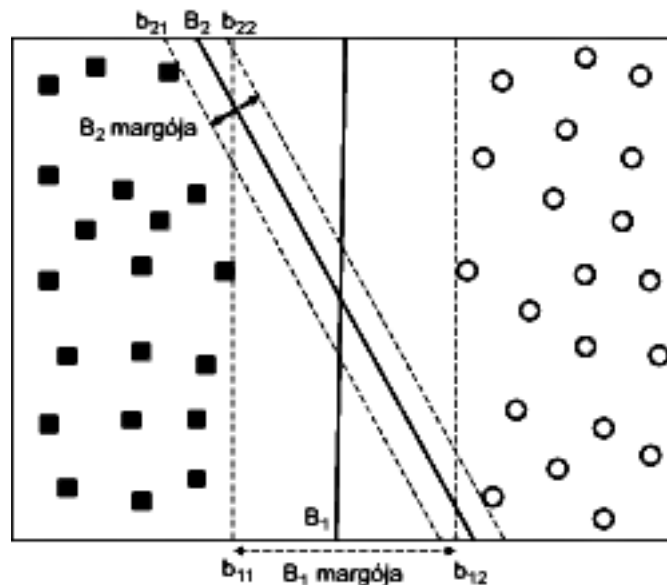


meg. Ez az úgynevezett *k-nearest neighbour*, vagyis a  $k$  legközelebbi szomszéd algoritmus. Ilyenkor egy új minta osztályozásakor a  $k$  legkisebb távolságra levő már osztályozott (eleve felcímkezett vagy elfogadott klasszifikációjú) pont határozza meg az osztályt.

Az úgynevezett tartóvektor-géppel, azaz *support vector machine*-nel, SVM-mel végzett klasszifikáció egy másik megközelítése a feladatnak. Az SVM lineáris szeparációra képes maximális margójú hipersík keresésével. A maximális margójú hipersík úgy szeparál mintákat, hogy a minták és a hipersík között a távolság a legnagyobb legyen. Mivel a tanítóhalmazon több hipersík is képes lesz megfelelő eredményt elérni (2.9 ábra), a legnagyobb távolságot tartó hipersík lesz az, amely teszteléskor és új minták klasszifikációjakor a legjobb teljesítményt nyújtja majd, mint az a 2.10 ábrán látható [21] [22].



2.9. ábra. Tanuláskor több hipersík is megfelelő lehet. Forrás: [23].



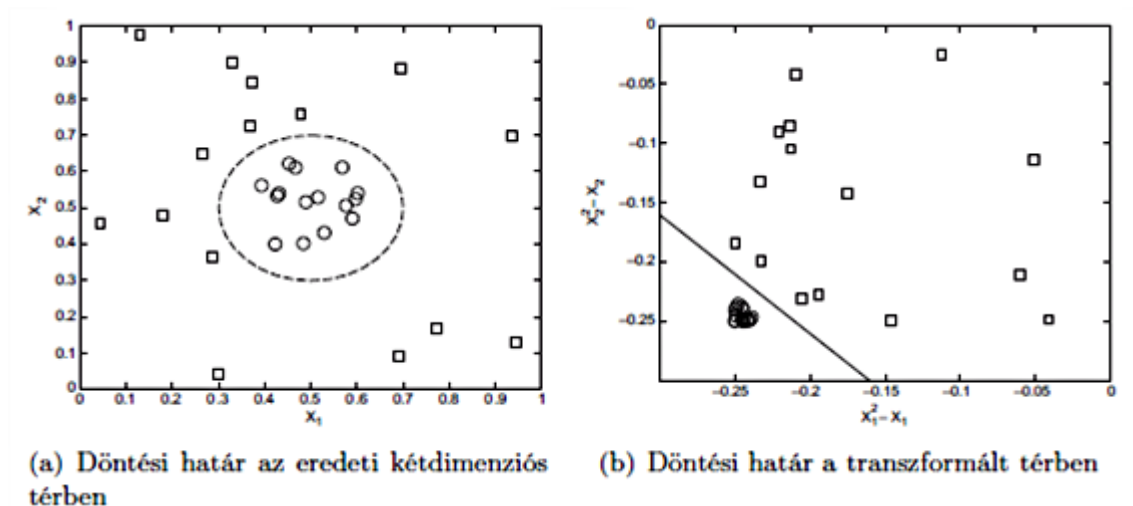
2.10. ábra. Teszteléskor és klasszifikációkor a maximális margójú hipersík a legideálisabb. Szerkesztve. Forrás: [23].

Kilenc különböző leíróval meltrák diagnosztizálásra használva egy SVM alapú rendszer 92,13%-os pontosságot ért el [24].

Amennyiben a mintáink nem szeparálhatóak lineárisan, megoldást jelenthet a döntési tér transzformációja, vagy olyan leírók kiválasztása, melyek lineárisan szeparálhatóak.

Egy másik megoldás a dimenziók számának növelése, mely által egy magasabb dimenziójú hipersíkkal lineárisan szétválaszthatóvá válnak a minták [24].

A fentebb leírt módszerek használatakor fontos a megfelelő leírók kiválasztása. Olyan magas dimenziójú és lehetségesen zajos adatoknál, mint a réslámpás kamerákkal készített szemfenékfotók, ez meglehetősen nehéz. Ez motivált arra, hogy egy olyan gépi tanulási módszert találjak, mely robusztus és zajtűrő, továbbá képes közel nyers adatokkal dolgozni.



2.11. ábra. Lineárisan nem szeparálható minták esetében segíthet a döntési tér transzformációja. Szerkesztve. Forrás: [23].

## 2.5. Neurális hálózatok

A fentebb leírt szempontok megfontolása alapján neurális hálózatok használata mellett döntöttem. A neurális hálózat olyan rendszer, mely azonos vagy hasonló műveleti egységekből áll, úgynevezett mesterséges neuronokból. Képes minták alapján tanulni a tanulási algoritmusán keresztül, a megtanult információt pedig képes felhasználni az előhívási (*recall*) algoritmusán keresztül [25].

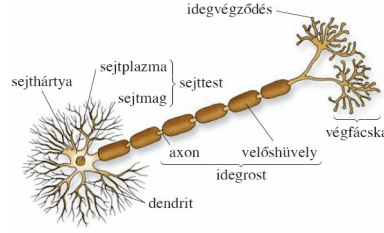
A neurális hálózatok alapvető építőeleme a mesterséges neuron, mely egy egy vagy több bemenettel rendelkező, egy kimenetű egység. A kimenetet, a választ az úgynevezett aktivációs függvény,  $\Phi$  határozza meg a bemenetek, az ingerek súlyozott összege alapján. Tehát a kimenet,  $y$ , a 2.2 egyenlet alapján kapható meg.

$$y = \Phi(\bar{w}^T \bar{x}) \quad (2.2)$$

A neuron kimenete,  $y$ .  $\Phi$  az aktivációs függvény,  $\bar{x} = (x_0, x_1, \dots, x_n)$  a bemeneti vektor,  $\bar{w} = (w_0, w_1, \dots, w_n)$  pedig a bemenetekhez tartozó súlyok vektora.

Ez a modell a valódi, az élőlényekben megtalálható idegsejtek működésén alapul. Az idegsejtekben a dendriteken kapott ingerek a sejtestben, a szómában összegződnek. Ha ez a kumulatív inger átlép egy küszöböt, azaz eléri az akciós potenciált, a neuron tüzel és az axon mentén végighaladó inger a szinapszisokon keresztül átadódik további idegsej-

teknek. Egy általános neuron illusztrációja található a 2.12 ábrán. A posztszinaptikus potenciálok (Postsynaptic Potential - PSP) befolyásolhatók úgy pozitív, gerjesztő irányba (Excitatory Postsynaptic Potential - EPSP), mint negatív, gátló irányba (Inhibitory Postsynaptic Potential - IPSP). Míg a bejövő ingerek analógok, számít az intenzitásuk, a létrejövő válasz digitális, a neuron vagy tüzel vagy nem. Ebben a modellben a kapcsolatoknak a szinapszisok, a súlyoknak a PSP-k felelnek meg, az akciós potenciált pedig az aktivációs függvény modellezi.



2.12. ábra. Egy idegsejt. Forrás: [26].

A  $w_0x_0$  szorzat az úgynevezett bias,  $b$ . Lineáris szeparáció esetén a tengelymetszetet határozza meg és küszöbértékként hat a neuron kimenetére nézve. A biast kiemelve a 2.3 formát kapjuk az aktiváció egyenletére.

$$y = \Phi(\bar{w}^T \bar{x} + b) \quad (2.3)$$

A neuron kimenete,  $y$ .  $\Phi$  az aktivációs függvény,  $\bar{x} = (x_1, x_2, \dots, x_n)$  a bemeneti vektor,  $\bar{w} = (w_1, w_2, \dots, w_n)$  pedig a bemenetekhez tartozó súlyok vektora.

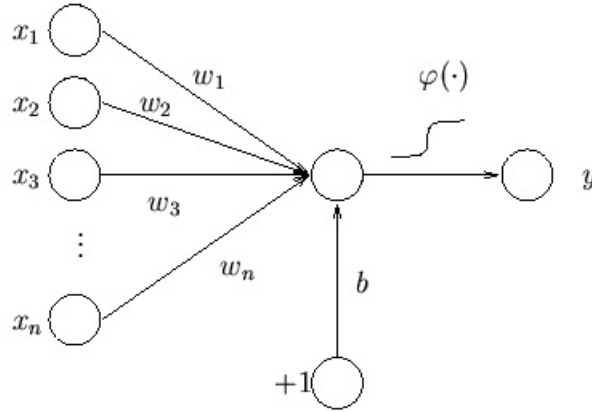
Egy ilyen mesterséges neuron Rosenblatt perceptronja [25]. Már egy perceptron is képes klasszifikációra és tanulásra. A perceptron aktivációs függvénye az előjelfüggvény,  $\text{sgn}(x)$ , ezáltal két osztályba tud osztályozni: a -1, illetve a +1 osztályba. A klasszifikáció során a súlyok alapján egy hipersíkkal szeparálja az osztályozandó pontokat. Egy perceptron csak lineáris klasszifikációt képes végrehajtani, két dimenzióban a  $\bar{w}^T x$  irányított egyenes mentén. Az egyenes normálvektora által kijelölt irányban lesznek a pozitív, az ellentétes oldalon pedig a negatív minták. A perceptron tanítása egy tanítóhalmazon keresztül valósítható meg. A tanítóhalmaz,  $T$ , bizonyos bemenetek,  $\bar{x}$ , és a rájuk rendszertől elvárt válaszok,  $d$ , párosait tartalmazza, ahogy azt a 2.4 egyenlet mutatja.

$$T = \{(\bar{x}, d) | d \in \{-1, 1\}\} \quad (2.4)$$

Az adott válasz és az elvárt kimenet különbsége hibát generál, a súlyok pedig ez alapján módosulnak. A megváltozás mértékét egy  $\eta$  tanulási paraméter határozza meg. A tanulási szabály a 2.5 egyenletben látható.

$$\bar{w}_{i+1} = \bar{w}_i + \eta(d(k) - y(k))\bar{x}(k), \text{ ahol } (\bar{x}(k), d(k)) \in L \quad (2.5)$$

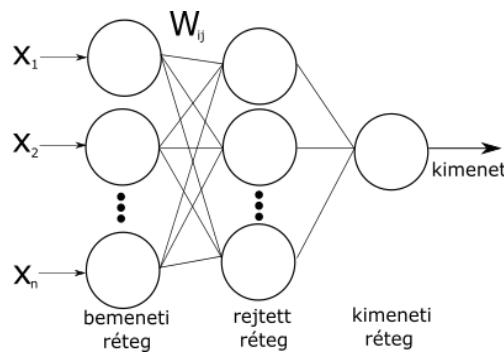
A perceptron súlyainak ( $\bar{w}_i$ ) módosulása.  $(d(k) - y(k))$  az elvárt kimenet és a neuron válaszáinak különbsége a tanulóhalmaz  $k$ -adik elemére.



2.13. ábra. Egy mesterséges neuron modellje. A bemenetek a bal oldalon,  $w_0$ , azaz  $b$  és  $x_0$  (ebben az esetben  $+1$ ) alul, a kimenet pedig a jobb oldalon találhatóak. Forrás: [27].

Ha a perceptron az ingerekre ugyanazt a választ adja, mint az elvárt kimenet, a tanítási lépésre nincs szükség, mivel a hiba 0 lesz. A perceptron tanulási algoritmus során a súlyok véges idő alatt konvergálnak [28], így ez egy a gyakorlatban is jól használható módszer.

Jóval komplexebb feladatokat képes megoldani a mesterséges neuronok hálózatba kapcsolása. Ebben az esetben az elemi számítási egységek rétegekbe rendeződnek, egy réteg neuronjainak kimenete a következő réteg bemenete lehet, a válasz előrecsatolódik - ez egy úgynevezett *feed-forward neural network*, előrecsatoló neurális hálózat. Ilyen hálózatokban a legelső, bemeneti réteg és a legutolsó, kimeneti réteg között úgynevezett rejtett rétegek találhatóak.



2.14. ábra. Egy általános előrecsatoló neurális hálózat architektúrája.

Mivel az egyes neuronok kimenete továbbadódik a következő rétegbe, a neuron által generált hiba is átadódik, a hálózat kimenetén a végső hiba a hálóban jelentkező hibákból áll össze. Így egy olyan tanulási algoritmusra van szükség, mely figyelembe veszi az eltérő mértékű hibákat és a hálózat súlyait ennek megfelelően módosítja.

Ha a hálózatunk  $L$  rétegből áll, az  $l$ . ( $l \in \{1, 2, \dots, L\}$ ) rétegben pedig  $N$  neuron található, az  $i$ . ( $i \in \{1, 2, \dots, N\}$ ) neuron kimenete  $y_i^{(l)}$  lesz. A hálózatban az  $(l - 1)$ . réteg  $i$ . neuronját és az  $l$ . réteg  $j$ . neuronját összekötő súlyt  $w_{ij}^{(l)}$ -vel jelölöm majd. Az  $l$ .

réteg súlyait a  $\overline{W}^{(l)}$  súlymátrix tartalmazza, biasait a  $\overline{b}^{(l)}$  vektor tartalmazza, kimenetei pedig az  $\overline{y}^{(l)}$  vektorban vannak.

Így jelölve,  $\overline{y}^{(l)}$ -t az előző réteg aktivációja alapján a fenti jelöléseket használva a 2.6 egyenlet alapján kaphatjuk meg.

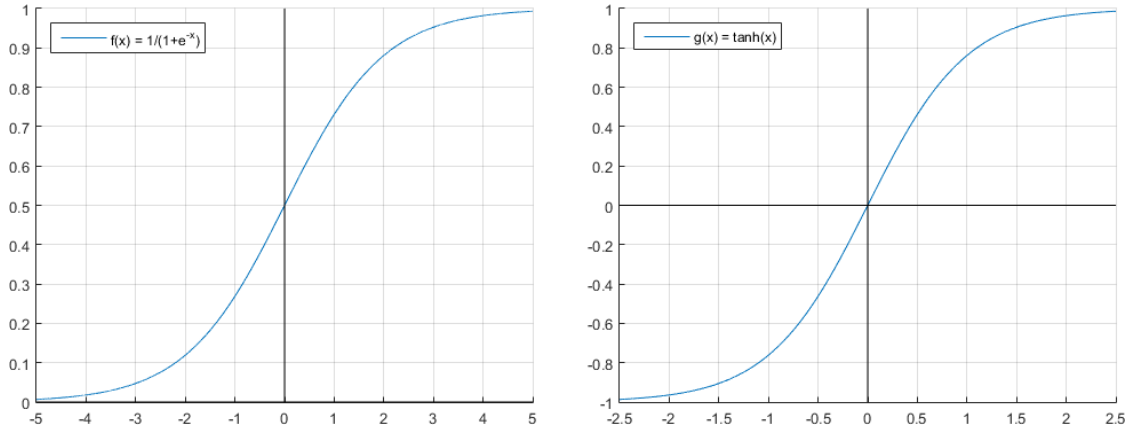
$$\overline{y} = \Phi(\overline{W}^{(l)}\overline{y}^{(l-1)} + \overline{b}^{(l)}) \quad (2.6)$$

Tanításhoz a hálózat hibáinak visszaterjesztésére van szükség, erre szolgál az úgynevezett *backpropagation* algoritmus. A neuronok  $\Phi$  aktivációs függvényeinek deriválhatóknak kell lenniük, mert a hiba-visszaterjesztés egy gradiens mentén minimalizálja a hálózat hibáját, melynek kiszámításához ez egy szükséges feltétel. A tanítás során a cél a gradiens felülete mentén ereszkedve a minimumpont megtalálása.

A  $\text{sgn}(x)$  függvény lecserélésre lehetséges jelölt lehet a logisztikus és a hiperbolikus tangens függvény, melyek a 2.7 és 2.8 egyenletekben illetve a 2.15 ábrán láthatóak.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.8)$$



2.15. ábra. A 2.7 és 2.8 egyenletekben leírt aktivációs függvények.

A hálózat visszaterjesztett hibáit az  $l$ . rétegre vonatkozóan a  $\overline{\delta}^{(l)}$  vektorok tartalmazzák. A súlyok módosulását a 2.9 egyenlet mutatja.

$$\overline{W}^{(l)}(k+1) = \overline{W}^{(l)}(l) + \eta \overline{\delta}^{(l)} \overline{x}^{(l)}(k) , \text{ ahol} \quad (2.9)$$

$$\delta_i^{(l)} = (\overline{\delta}^{(l+1)T} \overline{w}_i^{(l+1)}) y_i^{(l)} \quad (2.10)$$

A súlyok megváltozása backpropagation során. A pontos levezetés megtalálható [25] 4.2 fejezetében.

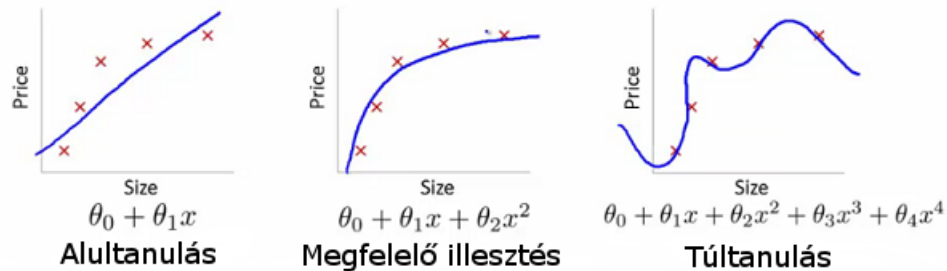
Már egy egyetlen rejtett rétegből álló feed-forward neurális hálózat is képes univerzális függvényapproximációra. Ez a képessége tetszőleges aktivációs függvények esetében ( $\Phi$ -

nek mindössze korlátosnak és nem konstansnak kell lennie) fenn áll, csak elegendő számú paraméter, melyek a súlyokban tárolódnak, szükséges a rejtett rétegben [30].

Egy hálózat megtervezésekor felmerülő kérdések egyike, hogy milyen legyen a hálózat topológiája. Hány rétegből álljon, illetve az egyes rétegekben hány neuron legyen? Bár a hálózat méretének növelése a rétegek számának vagy a bennük levő neuronok számának növelésével a hálózat teljesítményének növekedéséhez vezet (összetettebb asszociációkra válik képessé a hálózat), egy jelentős hátrányt is magában hordoz. Túl kevés szabad paraméter esetén a hálózat képtelenné válik az adatokból a megfelelő jellemzők kivonására és azokból a megfelelő asszociációk végrehajtására. Túl sok szabad paraméter esetén pedig a hálózat bár képes a tanult adatok megfelelő klasszifikációjára, képtelen lesz még nem látott adatok helyes osztályozására. Előbbi helyzetet hívjuk *underfitting*nek, utóbbit pedig *overfitting*nek.

Underfitting, azaz alultanulás esetén nincs elegendő kapacitása a hálózatnak a mintapontok feldolgozására. Ha a pontok egy  $e^x + \gamma(x)$  görbe mentén helyezkednek el, ahol  $\gamma(x)$  véletlen zaj, és a célunk, hogy a rendszer felismerje, hogy  $e^x$  mentén vannak elszórva a pontok és ez alapján klasszifikáljon, túl kevés szabad paraméterrel rossz esetben csak egy érintőegyenest lesz képes visszaadni nekünk. Ez azt is jelenti, hogy nem lesz képes megfelelően klasszifikálni sem a tanulandó pontokat, sem a még látatlan adatokat.

Overfitting, azaz túltanulás esetén a túl sok tanulható paraméter következtében szélsőséges esetben a hálózat egy az összes mintaponton áthaladó görbét tekint majd a legjobb eredménynek. Ezáltal a tanulóhalmazon kapott eredmény megfelelő lesz, de ismeretlen adatokra jóval rosszabb eredményeket fog produkálni a rendszer. Ezekre a jelenségekre példa a 2.16 ábrán látható.



2.16. ábra. Bal oldalon: túl kevés paraméter áll rendelkezésre megfelelő görbe létrejöttéhez, underfitting áll elő. Középen: pontosan elegendő mennyiségű paraméter elérhető. Jobb oldalon: túl sok szabad paraméter esetén nem általánosít a hálózat, overfitting áll elő. Szerkesztve. Forrás: [31].

A rétegek számának növelésével előtérbe kerül egy újabb probléma: az eltűnő gradiensek problémája (*vanishing gradients problem*). Nagy mélységű neurális hálózatokban a backpropagation során a kimeneti rétegtől távolodva a tanulás sebessége egyre csökken [32]. A megfelelő sebességű tanuláshoz a lehetséges megoldások közé a hálózat rétegeinek számával arányosan növekvő számítási teljesítmény növekedés illetve az általában használt szigmoid nemlinearitások helyett más, nem telítődő nemlinearitások használata.

Ilyen lehetséges nemtelítő nemlinearitás a ReL, azaz a *rectifying nonlinearity*, mely

a bemenete és a 0 maximumát veszi (2.11 egyenlet). Így a kimenete nem a korlátos  $[-1, 1]$  tartományba esik, hanem a  $[0, \infty)$  nem korlátos tartományba. A backpropagation működése szempontjából fontos, hogy differenciálható legyen a nemlinearitás, így a 2.12 egyenletben leírt, úgynevezett *softplus* függvény egy jelölt.

$$f(x) = \max(0, x) \quad (2.11)$$

$$g(x) = \ln(1 + e^x) \quad (2.12)$$

A ReL ( $f(x)$ ) nemlinearitás differenciálható közelítése a  $g(x)$  softplus függvény.

A hálózat teljesítményét nagyban befolyásolja a tanítóhalmaz minősége. A jó eredmények érdekében egy olyan halmazra van szükség, melyben elegendő mennyiségű, megfelelően felcímkézett adat található. Fontos, hogy az osztályok elemszáma kiegyensúlyozott legyen. Nem szükséges, hogy pontosan ugyanannyi kép legyen egy-egy osztályban, de nagyságrendi különbségek esetén a hálózat jobb teljesítményt mutat majd azokra a kategóriákra, melyekben több kép található, ugyanis tanulás során ezekkel a kategóriákkal gyakrabban találkozunk, így a tanult paraméterek nagyobb súllyal hajlanak majd ezeknek a kategóriáknak a megfelelő osztályozására.

Kérdéses továbbá maguknak a tanítóhalmaz elemeinek mérete és minősége. Nagyméretű adatok esetén megnő a feladat végrehajtásához szükséges tárolási kapacitás és számítási teljesítmény. A minták lehetnek zajosak, mely a rendszer számára lehetetlenné teszi a megfelelő klasszifikációt. Nem kizárható az sem, hogy maga a besorolása a képnek hibás. Kis számú hibás tanulandó minta esetén ez nem okoz gondot, akár még előnyt is jelenthet, mivel elvárható egy klasszifikátortól, hogy egy bizonyos mértékig hibátűrő legyen.

A kiindulási paraméterek megválasztása sem triviális. Meg kell választani a súlyok kezdeti értékét, egy megoldás ezek konstans értékre állítása, például 0-ra. Több megfontolást igényel az  $\eta$  tanulási paraméter inicializálása. Túl alacsony érték esetén a tanulás meglehetősen lassú lesz, túl nagy érték esetén pedig lehetséges, hogy a hálózat nem konvergál, elszáll [25]. Egy megoldás erre a paraméter csökkentése a tanítás alatt. Bizonyos lépésszám után, például amikor végigérünk a tanítóhalmazon, valamilyen szabály alapján csökkentjük  $\eta$ -t.

A tanítás során többféle módon változtathatjuk a hálózat súlyait. Tehetjük lépésenként, minden egyes iteráció hibáját visszaterjesztve, vagy egy bizonyos mennyiségű lépés után. Utóbbi esetben gyorsul a tanulás, így csak az egy tanítási ciklus alatti hibák átlaga alapján módosulnak a súlyok, nem lesz annyira reszponzív a rendszer egy-egy kiugróan rosszul klasszifikált mintára. Lépésenkénti frissítés mellett az szól, hogy a minden egyes elemre elvégzett súlymódosítás zajt visz a rendszerbe, így elkerülhető a lokális minimumokba ragadás a tanítás során.

Az adatok klasszifikációját nagyban segíti a megfelelő reprezentáció megválasztása. Példaképpen, lineárisan nem szeparálható egy olyan mintahalmaz, amelyben a két osztály koncentrikus körökként reprezentált Descartes-koordinátákban. Viszont ha polár-

koordinátákban fejezzük ki az egyes mintapontok elhelyezkedését, a probléma nemcsak megoldhatóvá válik, hanem jóval egyszerűbbé is, mint azt a 2.11 ábra mutatta.

Ezt használják ki a mély, sok rétegből álló neurális hálózatok. Az egyes rétegek egyre összetettebb reprezentációkkal dolgoznak, melyek az előző rétegek egyszerűbb reprezentációiból állnak össze. Mivel a bemeneti adatok ebben az esetben színes képek, egy olyan hálózati architektúrát választottam, mely képfeldolgozási feladatokban state-of-the-art eredményeket ért el megmérettetésekben világszerte [33][34] - konvolúciós mély neurális hálózatokat.

Az ilyen hálózatok az 1990-es években háttérbe szorultak az egyéb gépi tanulási módszerek mellett, mint az SVM-ek és a naiv bayesi klasszifikátorok, de már abban az időszakban is voltak próbálkozások célhardverrel gyorsítani a tanulást, mivel az akkor elérhető számítási teljesítményhez képest óriási igényeket támasztottak a mély neurális hálózatok. Napjainkban viszont újra nagy népszerűségnek örvendenek főként a szükséges számítási teljesítmény könnyű elérhetőségének és a már említett eredményeknek köszönhetően. A szükséges számítási teljesítményt a piacon széles körben elérhető grafikus processzorok, GPU-k nyújtják, melyek robbanásszerűen fejlődtek az elmúlt évtizedben. A GPU-k gyorsan elvégzett mátrixműveleteinek és párhuzamos számítási képességeinek köszönhetően egy hálózat tanítása akár ötvenszer gyorsabb lehet, mint CPU-n [35]. Ez a teljesítménynövekedés ellensúlyozza a fentebb említett eltűnő gradiens problémát és ezáltal teszi vonzóvá ezt a gépi tanulási módszert.

Egy általános konvolúciós hálózat rejtett rétegeiben találhatóak konvolúciós, alul-mintavételezési, teljesen csatolt, nemlinearitás és *fine-tuning* rétegek. A konvolúciós rétegek a lokális mezőkben levő bemenetekre alkalmazzák a súlyaik által kijelölt szűrőt. A bemeneti adatot ezek a mezők lefedik, egy adott konvolúciós rétegben a neuronok súlyai közősek, tehát az egész bemeneten ugyanazzal a szűrővel ablakol végig. Az ablakoknak nem feltétlen kell átfedniük, ezt a réteg *stride*-ja írja le, meghatározza mekkora a távolság két ablak középpontja között. Esetünkben a képek egy  $M_x \times M_y$  méretű mátrixnak felelnek meg, egy konvolúciós réteg kimenetének mérete a 2.13 és 2.14 egyenletekben látható.

$$M_x^n = \frac{M_x^{n-1} - K_x^n}{S_x^n + 1} + 1 \quad (2.13)$$

$$M_y^n = \frac{M_y^{n-1} - K_y^n}{S_y^n + 1} + 1 \quad (2.14)$$

Az  $n$ . konvolúciós réteg kimenetének mérete.  $K_x^n$  és  $K_y^n$  az ablak mérete,  $S_x^n$  és  $S_y^n$  pedig a középpontok közötti távolság, a stride.

Az alul-mintavételezést az úgynevezett *pooling* rétegek valósítják meg. Csökkentik a bemenet méretét, a konvolúciós rétegekhez hasonlóan ablakolással. Ez történhet a mintavételezési ablak kiátlagolásával, az ablak maximális vagy minimális elemének kiválasztásával. Az alul-mintavételezésnek köszönhetően a hálózat nem lesz túlérzékeny a kis területen bekövetkező változásokra, például egy pixel megváltozására. A konvolúciós rétegeket pooling rétegek szokták követni, mely mögött a motiváció az, hogy így a



hálózat térbeli érzékenysége csökken, a képen bizonyos tulajdonságok megléte és kihangsúlyozottsága számít majd, nem pedig a pontos helyük.

A konvolúciós és pooling rétegek kapcsolatának biológiai motivációjának tekinthető Hubel és Wiesel [36] kísérlete a macska vizuális kortexével. A kísérletben a neuronok egy fajtája, a komplex receptív mezejű neuronok produkáltak alig eltérő intenzitású választ egy vonal térbeli elhelyezkedésére, viszont jelentősen csökkent a válasz intenzitása, ha a vonalat elforgatták akár  $5-10^\circ$ -kal.

A nemlinearitást megvalósító rétegek a hálózat elvárt működésének érdekében szükségesek. A fentebb vázolt gradiens eltűnés probléma [32] súlyosságát csökkenthetjük és így a tanulás sebességét növelhetjük, ha az általában használt szigmoid nemlinearitások helyett a ReL nemlinearitást megvalósító egységet használjuk. Tapasztalatok alapján [37] jobb eredményeket lehet elérni az úgynevezett *leaky ReLU*-val (leaky rectifying non-linearity unit), melyek a ReL nemlinearitása helyett (2.11 egyenlet) a bemenetnek és  $\alpha$ -szorosának maximumát veszi (2.15 egyenlet). Egy sikeres háló  $\alpha = 0,5$  értékkel dolgozott ezen a feladaton [37].

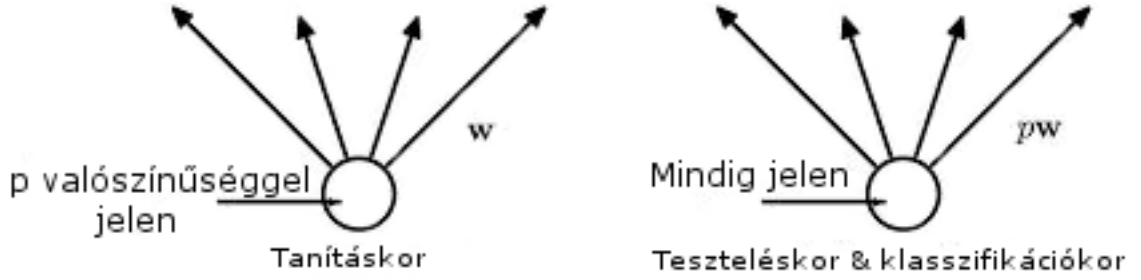
$$f(x) = \max(\alpha x, x) \quad (2.15)$$

A Leaky ReL nemlinearitás átengedi a bemenet egy  $\alpha$ -szorosát, ha az kisebb, mint 0.

A teljesen kapcsolt rétegekben az összes bemenet az összes neuronnal kapcsolatban áll. Ezek a rétegek hasonlítanak legjobban egy klasszikus előrecsatolt neurális hálózat rejtett rétegeire. Az eddig kinyert reprezentációkból itt jönnek létre az asszociációk és tárolódnak el a súlyokban. Különösen fontos ezért a teljesen kapcsolt rétegek méretének körültekintő megválasztása, mivel az over- és underfitting problémára itt hajlamos igazán a hálózat.

A fine-tuning rétegek a hálózat teljesítményét szabályozzák. Egy gyakran használt réteg a *dropout* réteg, mellyel az overfitting hatásait lehet bizonyos mértékig csökkenteni. A dropout réteg  $p$  valószínűséggel a tanítás során az egyes neuronok ingereit meghagyja,  $1 - p$  valószínűséggel pedig nullára állítja. A hálózat tanulás utáni, klasszifikációra használt állapotában pedig a neuronok kimeneti súlyait  $p$ -vel megszorozza, így kompenzálva a tanításkori kinullázásokat (2.17 ábra). A réteg használata mögötti motiváció, hogy ez zajt tesz a rendszerbe, egyfajta regularizációként működik és így csökkenti az egyes neuronok közötti ko-adaptáció valószínűségét [38].

A hálózat kimeneti rétegében jön létre a döntés, hogy melyik kategóriába sorolódik majd az adott bemenet. A tervezéskor választott hibafüggvény itt számolja ki az eltérést a mintára elvárt kimenet és az aktuális, a tanulás során a mintára adott kimenet között. Ha a kimenetre úgy tekintünk, mint egy valószínűségeket tartalmazó vektorra (2.16 egyenlet), amely azt mondja meg, hogy melyik osztályba mekkora valószínűséggel tartozik a minta, akkor a kimeneti vektor legnagyobb eleme,  $\max(\{y_i | y_i \in \bar{y}\})$  lesz a hálózat által feltételezett osztálya a bemenetnek. Ezen elképzelés alapján, a tanulóhal-mazban a minta elvárt osztályát értelmezhetjük úgy, mint egy olyan vektort, melynek



2.17. ábra. A dropout működése. Bal oldalon: tanításkor  $p$  valószínűséggel maradnak jelen a bemenetek. Jobb oldalon: teszteléskor besúlyozódnak a kimenetek a tanításkori jelenlét valószínűségével. Szerkesztve. Forrás: [39].

minden eleme 0, kivéve a jelölt osztályt, melynél az érték 1.

$$\bar{y} = (y_0, y_1, \dots, y_n) , \text{ ahol } \sum_{i=0}^n y_i = 1 \text{ és } y_i \in [0, 1] \quad (2.16)$$

$$\bar{d} = (d_0, d_1, \dots, d_n) , \text{ ahol } d_i = \begin{cases} 0, & \text{ha } i \neq d \\ 1, & \text{ha } i = d \end{cases} \quad (2.17)$$

A hálózat egy  $\bar{x}$  bemenetre adott hovatarozási valószínűségeit  $\bar{y}$  tartalmazza. A  $\bar{d}$  elvárt kimeneti vektor csak egy koordinátájában 1, máshol 0. Ezt a koordinátát a tanítóhalmaz  $(\bar{x}, d)$  párosaiból  $d$  határozza meg.

Egy neurális hálózatot ábrázolhatunk irányított gráf formájában. A *mély* jelző abból ered, hogy egy mély hálózat gráf-reprezentációja sok rétegből áll. Lehet előrecsatolt, vagy rekurrens. Előrecsatolt (vagy feed-forward) esetben a hálózatot leíró irányított gráfban nem található kör, rekurrens esetben pedig tartalmaz kört.

### 3. Módszerek és munka bemutatása

A feladat elvégzéséhez szükséges volt a megfelelő előkészületek megtétele és ezek alapján egy megfelelő neurális hálózat létrehozása, tesztelése és az eredmények kiértékelése.

A munkához számos neurális hálózatok leírására és mély tanulásra alkalmas környezetből választhattam. A Berkeley Vision and Learning Center által kifejlesztett és karbantartott Convolutional Architecture for Fast Feature Embedding, másnéven Caffe környezet mellett döntöttem. A Caffe egy C++ könyvtár, mely MATLAB és Python front-endekkel rendelkezik. Előnyt jelentett, hogy hatékonyan képes kihasználni az Nvidia GPU-k CUDA magjait, ezáltal a már elérhető hardveren hatékonyan dolgozhattam. A feladatot egy Nvidia GeForce GTX Titan X videokártyával felszerelt, Ubuntu Linux 15.04 operációs rendszerű számítógépen végeztem.

A környezet kiválasztása mellett további nélkülözhetetlen előkészületi lépés volt egy megfelelő tanító- és teszhalmaz létrehozása. A fentebb már említett Kaggle Diabetic Retinopathy Detection Challenge-hez biztosított adathalmaz egy használható kiindulási pontot jelentett, de a minták eloszlása miatt nem volt ideális egy gépi tanulási feladat elvégzésére. Elsőként a halmazt 80%-20% arányban megbontottam, a nagyobb rész lett a tanítóhalmazom, a maradékot pedig a teszhalmazként kezeltem. A problémát az jelentette, hogy bizonyos kategóriák számottevően több mintát tartalmaztak a tanítóhalmazban, mint a többi kategória. Már az is torzította volna a hálózat eredményeit, hogy az utolsó két osztály mintái kevesebb, mint 5 százalékát tették ki a teljes mintaszámnak. A második kategória több, mint kétszer annyi képből állt, mint az első, és több képet tartalmazott, mint amennyi az első, harmadik, illetve negyedik kategóriába volt osztva együttesen. A helyzetet súlyosbítva az eredeti adathalmaz nulladik kategóriája, mely az egészséges szemeket tartalmazza, önmagában több képből állt, mint az összes többi kategória [14].

A probléma megoldására ki kellett egyensúlyoznom az egyes osztályok elemszámait. Ehhez olyan transzformációkat kellett alkalmaznom, melyek megtartják az egyes minták információtartalmát és besorolásukat is. Mivel a két utolsó osztály elemszáma volt a legkisebb, ezen két osztály elemszámának növelése mellett döntöttem, majd a többi osztály elemszámát az új, bővített negyedik és ötödik osztály méretéhez igazítottam. Ezt affin transzformációkon keresztül értem el, tükrözések által. A képeket tükröztem az  $x$  és az  $y$  tengelyre, majd az  $x$ , tengelyre tükrözött képeket még egyszer tükröztem az  $y$  tengelyre.

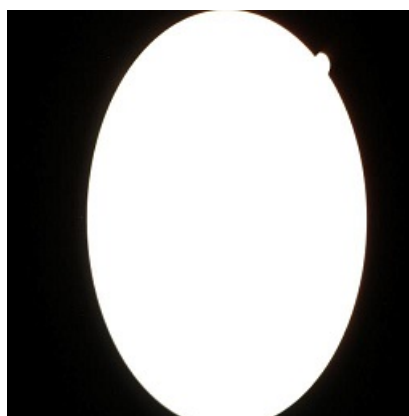
Eredményképp a legkisebb elemszámú osztályok mérete a négyszeresére nőtt: a súlyos NPDR kategóriába tartozó képek száma 698-ról 2792-re, a proliferatív DR kategória elemszáma pedig 566-ról 2832-re növekedett. A folyamatot egy Python-szkript segítségével hajtottam végre, mely egy bemeneti listafájlban levő fájlokat betölti, végrehajta rajtuk a transzformációt, majd egy kimeneti fájlba írja az eredeti fájl illetve az alapján

transzformált képek fájlneveit, továbbá megfelelően fel is címkézi őket. Ez szkript megtalálható a dolgozathoz csatolt CD-n. Ezekhez a számokhoz idomul az enyhe NPDR-ű minták száma 1954 képpel, de továbbra is kiugróan sok nulladik és második kategóriás képünk van. Végül 2928 képet tartottam meg a nulladik kategória mintáiból és 2419-et a másodikéiból. Ezáltal egy jóval kiegyensúlyozottabb, 12357 elemből álló tanulóhalmazhoz jutottam, melynek felépítése a 3.1 táblázatban látható.

Kategória neve	Régi elemszám	Régi eloszlás	Új elemszám	Új eloszlás
Nincs DR (0)	20648	73,48%	2928	23,69 %
Enyhe NPDR (1)	1954	6,96%	1954	15,81%
Mérsékelt NPDR (2)	4233	15,07%	2419	19,57%
Súlyos NPDR (3)	698	2,48%	2792	22,59%
Proliferatív DR (4)	566	2,01%	2264	18,32%
Összesen	35126	100%	12357	100%

3.1. táblázat. Az eredeti (a 80%-20% vágás utáni) és a kiegyensúlyozott adathalmazok elemszáma illetve a kategóriák elemszámának arányai.

Bár az egyes kategóriák elemszáma megfelelő volt, a képek minőségén és tartalmán a részhalmazok átméretezése nem változtatott. Bizonyos képeken a címke nem volt megfelelő, míg másokon a kép minősége (rosszul látható erezet, a képalkotási módszerből származó műtermékek jelenléte, elmosódott kép) volt elégtelen a tanításhoz. Egy szélsőséges esetet mutat a 3.1 ábra. Az ilyen képek a hálózat zajtűrőképességét növelik, ezért miután megbizonyosodtam a létezésükről benne hagytam őket a tanulóhalmazban mint zajos mintákat.



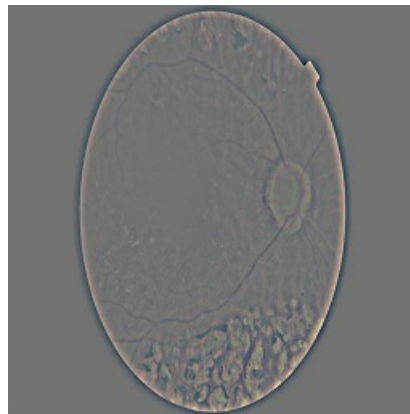
3.1. ábra. Egy hibásan 3-as kategóriába sorolt kép (súlyos NPDR). Félreklasszifikált és a tanulásra nézve zajt jelentő minta.

Az eredeti képek nagy mérete is problémát okozott a tanításban. A réslámpás kamerák képalkotási módszere miatt a létrejött kép kettő- és ötszörös nagyítású lesz a valósághoz képest, így a kapott képek mérete is ezt tükrözi. Az adathalmaz képei átlagosan  $3000 \times 2000$ -es felbontásúak voltak. Ekkora méretű képek feldolgozásához egy roppant méretű hálózatra lenne szükség és óriási számítási teljesítményre, hogy egyáltalán a háló-

zat teljesítményére a tanulás közben kapott átmeneti eredményekből becslést adhassunk, mennyire jól működik a rendszerünk. Ezért az új tanítóhalmaz és a teszhalmaz képeit átméreteztem  $512 \times 512$  méretűre.

Kérdéses volt továbbá, hogy a háromcsatornás, RGB színtérben levő képeken kell-e végrehajtsak valamilyen előfeldolgozási módszert. A szürkeskálásra való konverziót elvetettem, bár ez gyorsította volna a tanulást, mert háromcsatornásak helyett csak egycsatornásak lettek volna a képek. Ennek oka az volt, hogy az egyes csatornák fontos információt hordoznak, mivel az eltérő hullámhosszú fénnel más-más anatómiai struktúrákat figyelhetünk meg jól a szemfenéken. Mégis motivációt éreztem arra, hogy valamilyen módszerrel a megvilágításbeli különbségeket kiegyensúlyozzam, mivel a különböző funduskamerák eltérő hullámhosszokkal dolgoznak.

Létrehoztam egy alternatív tanító- és teszhalmazt melyben az előfeldolgozott képek voltak. Az úgynevezett *local colour average subtraction* módszert alkalmaztam az átméretezett eredeti képekre. A képek egyes csatornáin egy MATLAB-szkripttel gaussi elmosást végeztem 18-as ablakmérettel és 21-es szórással, majd a kapott eredményt kivontam az eredeti csatornákból. Ezután megnőveltem a képek kontrasztját. Bár az így kapott képeken valóban eltűntek a megvilágításbeli különbségek mint ahogy az a 3.2 ábrán látható, fennállt a lehetsége, hogy az egyes csatornában hordozott információ sérült vagy elveszett. Erre a lépésre tekinthetünk úgy, mint egy tudatos konvolúciós lépésre, mivel gaussi ablakok konvolúciójával végeztem az előfeldolgozást. A folyamatot végrehajtó MATLAB-szkript megtalálható a dolgozathoz csatolt CD-n.



3.2. ábra. Proliferatív diabetikus retinopátia jeleit mutató szemfenékfotó, az előfeldolgozás után  $256 \times 256$  pixeles képméretben.

A hálózat létrehozásához áttanulmányoztam a képfeldolgozási feladatokban jeles eredményeket elérő hálózatokat és az azokban levő megfontolások alapján terveztem meg a saját hálózatomat. Az alapokat a 2010-es *Imagenet Large Scale Visual Recognition Challenge*-en (ILSVRC) 37,5%-os hibarátát, vagyis 62,5% pontosságot elérő hálózatra építettem [40]. A hálózat maga a már fentebb említett konvolúciós réteg-nemlinearitás-alulmintavételezés blokkokból épül fel, a hálózat végén pedig két 4096 neuronból álló teljesen kapcsolt réteg található nemlinearitásokkal. Érdekessége, hogy a hálózat a bemeneti réteg után illetve a végső, kimeneti réteg előtt ketté lett bontva, és így két külön

GPU-n lett tanítva. Egy új elem, melyet a hálózatom tervezésekor felhasználtam az úgynevezett *local response normalizer*, mely a hálózat általánosítási képességét segíti, leírása [40] 3.3 fejezetében található.

A tervezett hálózataim legutolsó rétege egy ötelemű teljesen kapcsolt réteg, ebből képződik majd a fentebb leírt valószínűségeket tartalmazó vektor. Ezt úgy értem el, hogy a hiba kiszámítása előtt lenormáltam a kimeneti vektor elemeit a *softmax* függvénnyel. A softmax függvény segítségével egy  $K$  dimenziós valós értékű vektort normálhatunk egy  $K$  dimenziós  $[0, 1]$  intervallumba eső értékű vektorrá. A softmax függvény a 3.1 egyenletben látható.

$$\sigma(\bar{y})_j = \frac{e^{y_j}}{\sum_{k=0}^K e^{y_k}} \quad (3.1)$$

A softmaxból kapott  $\bar{\sigma}$  vektorra számoltam a hálózat hibáját. A gyors tanulás érdekében az volt a célom, hogy kis hibára is viszonylag nagy meredekségű gradiens jöjjön létre. Ez azért előnyös, mert így elkerülhető az az eset, amikor kvázi-lokális minimumban megakad a tanulás: a túlságosan kis gradiens miatt úgy tűnhet, hogy a hálózat konvergált, de valójában elegendő idő múlva túllendül ezen a kis meredekségű régió.

Erre a megoldást a *cross-entropy loss* szolgáltatja (3.2 egyenlet). Ideális erre a feladatra, mert a logaritmus miatt a kis eltérések is nagyobb hibát generálnak, mint egy négyzetes hiba esetén. Továbbá, ha a softmax függvénnyel normált vektorral számoljuk az elvárttól való eltérést, visszakapjuk az eredeti  $\bar{y}$  kimeneti vektorunkat, vagyis a valós kimenetre számoljuk a hibát, mégis előnyösebb eredménnyel.

$$L = - \sum_{i=1}^k \ln(\sigma_i) d_i \quad (3.2)$$

Az így kapott hibát a *stochastic gradient descent* algoritmussal terjesztettem vissza a hálózatba. A Caffe környezetben a működése a 3.3 és 3.4 egyenletekben látható.

$$V_{t+1} = \mu V_t - \eta \nabla L(W_t) \quad (3.3)$$

$$W_{t+1} = W_t + V_{t+1} \quad (3.4)$$

A rendszer létrehozásakor az overfitting elkerülése érdekében fontos volt, hogy közben tartsam a szabad paraméterek számát. Ennek legegyszerűbb módja, hogy a végső teljesen kapcsolt rétegek méretét az egyes futtatások során szerzett adatok alapján módosítom. Itt érdemes megjegyezni, hogy bár a célunk a teszhalmazon mért pontosság maximalizálása, a tanulási algoritmus a tanítóhalmazon mért hibát minimalizálja. Előfordulhat, hogy a hiba a tanulóhalmazon számolva tovább csökken, mindeközben a tesztelés során a hiba nagyságrendekkel lassabban teszi ezt, akár növekedhet is. Az overfitting beszédes jele, hogy a tanulóhalmazon számolt hiba rohamosan csökken, míg a tesztelés során a hálózat pontossága is zuhan. Ilyenkor érdemes „elkapni” a hálózatot még mielőtt ez megtörténik és a legjobb elérhető pontossággal dolgozni. Ez nagy eséllyel fordulhat elő korlátozott mennyiségű tanítóminta esetében.

Réteg Típusa	Ablak Mérete	Jellemző paraméterek
Bemenet	-	256×256×3 - 100-as batch
Konvolúció	11×11//4	64 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Max Pool	3×3//2	-
LRN	5×5	$\alpha = 10^{-4}$ $\beta = 0,75$ [40]
Konvolúció	5×5//1	128 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Max Pool	3×3//2	-
LRN	5×5	$\alpha = 10^{-4}$ $\beta = 0,75$ [40]
Konvolúció	3×3//1	256 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Konvolúció	3×3//1	256 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Konvolúció	5×5//1	128 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Max Pool	3×3//2	-
Teljesen kapcsolt	2048	-
ReLU	-	$\alpha = 0,5$
Dropout	-	$p = 0,5$
Teljesen kapcsolt	1024	-
ReLU	-	$\alpha = 0,5$
Dropout	-	$p = 0,5$
Teljesen kapcsolt	512	-
ReLU	-	$\alpha = 0,5$
Dropout	-	$p = 0,5$
Teljesen kapcsolt	5	-
Softmax & Cross-entropy loss	-	-

3.2. táblázat. A feladat elvégzéséhez megtervezett első neurális hálózat.

Ezen megfontolások alapján terveztem meg a mély konvolúciós hálózataimat. Az első hálózatom leírása a 3.2 táblázatban található. Felhasználásra kerültek megtervezésekor az előrecsatolt neurális hálózatok tervezésére vonatkozó elméleti megfontolások és a fentebb említett tapasztalati tudás is.

A 3.2 táblázat hálózatának mintájára létrehoztam egy másik, kisebb teljesen kapcsolt rétegekből álló hálózatot arra az esetre, ha overfit állna elő az eredetiben. A kisebb hálózat leírása a 3.3 táblázatban látható. Egy másik megvizsgált módszerem a pontosság növelésére az agresszív méretcsökkentés mellett az 512×512 méretű képekkel való tanítás. A 3.2 és 3.3 táblázatokban ismertetett hálózatok tanítása után mind 256×256 mind pedig 512×512 méretű képekkel a legjobban teljesítő konfiguráción megvizsgáltam az előfeldolgozott képek hatását a rendszer pontosságára a megfelelő képmérettel.

A legvégső célom az volt, hogy a legjobban teljesítő hálózatot kiválasztva több hálózat együttes döntését megvizsgáljam. Erre a célra az úgynevezett *ensemble boosting* módszert használtam. Az ensemble boostinggal több hálózat együttműködve tud döntést hozni. Az eljárás során az első hálózat tanítása után a tanulóhalmaz elemeit sorba

rendeztem a hálózat által generált hiba szerint, majd egy bizonyos mennyiségüket egy újabb hálózattal újratanítottam. Ezután a két hálózat közös döntése alapján, melyben az egyes döntések megbízhatóságát a teszhalmazon külön-külön elért pontosság alapján súlyoztam, még egy hálózatot tanítottam. A végrehajtásra használt Python-fájl megtalálható a csatolt CD-n levő szkriptfájlok között. Majd a közös döntéshez hozzávettem a harmadik hálózatot, és így tovább. Ezt a folyamatot addig ideális folytatni, amíg a rendszer teljesítménye számottevően növekszik, vagy amíg a rendelkezésre álló képekből megfelelő mennyiség áll rendelkezésre (ezt a fentebb említett vágás méretével szabályozhatjuk).

Ezt a mennyiséget ebben az esetben a halmaz elemeinek 80%-ban határoztam meg, mivel a rendelkezésre álló képek mennyisége nem engedett volna meg számottevően kisebb százalékot. 80%-ot választva is a további hálózatok tanítására az újabb hálózatok tanítóhalmazában 9886 kép áll majd rendelkezésre. Ennél kisebb számot nem tartottam megfelelőnek a hálózatok teljesítményének biztosítására.

A fent leírt hálózatokat előfeldolgozott és előfeldolgozatlan képekkel is tanítottam, az eredmények és megfigyelések alább találhatóak az Eredmények, értékelés fejezetben.



Réteg Típusa	Ablak Mérete	Jellemző paraméterek
Bemenet	-	$256 \times 256 \times 3$ - 100-as batch
Konvolúció	$11 \times 11 // 4$	50 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Max Pool	$3 \times 3 // 2$	-
LRN	$5 \times 5$	$\alpha = 10^{-4}$ $\beta = 0,75$ [40]
Konvolúció	$5 \times 5 // 1$	100 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Max Pool	$3 \times 3 // 2$	-
LRN	$5 \times 5$	$\alpha = 10^{-4}$ $\beta = 0,75$ [40]
Konvolúció	$3 \times 3 // 1$	200 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Konvolúció	$3 \times 3 // 1$	200 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Konvolúció	$5 \times 5 // 1$	100 kimeneti csatorna
ReLU	-	$\alpha = 0,5$
Max Pool	$3 \times 3 // 2$	-
Teljesen kapcsolt	600	-
ReLU	-	$\alpha = 0,5$
Dropout	-	$p = 0,5$
Teljesen kapcsolt	300	-
ReLU	-	$\alpha = 0,5$
Dropout	-	$p = 0,5$
Teljesen kapcsolt	200	-
ReLU	-	$\alpha = 0,5$
Dropout	-	$p = 0,5$
Teljesen kapcsolt	5	-
Softmax & Cross-entropy loss	-	-

3.3. táblázat. A 3.2 táblázatban leírt konvolúciós neurális hálózat kisebb változata.

## 4. Eredmények, értékelés

A hálózataimat az alábbi közös paraméterekkel tanítottam:

- A kiindulási tanulási ráta,  $\eta_0 = 0,01$ .
- A bizonyos iterációnként (mely hálózatonként változó) frissített tanulási ráta  $\eta_{i+1} = \gamma\eta_i$ , ahol  $\gamma = 0,1$ .
- Az előző paramétermódosítás súlya,  $\mu = 0,9$ .
- 5000 iterációnként mentettem a hálózatok állapotát.
- 100-as kötegekben tanítottam a hálózatokat.

Ezek főként az SGD (3.3) algoritmusban megtalálható paraméterek, a tanulás sebességét illetve a gradiens stabilitását határozzák meg. Az ezen értékektől való eltérések az alábbiakban mindig feltüntetésre kerülnek majd, amennyiben ez nem történik meg, feltételezhető, hogy a hálózat tanítási paraméterek ezeknek felelnek meg.

Elsőként a legjobban teljesítő hálózatot kerestem meg. Ehhez a két hálózatomat (leírásuk a 3.2 és 3.3 táblázatokban található) tanítottam a két különböző méretű képeket tartalmazó tanítóhalmazzal. A képek méretén és a véletlen sorrendű beválogatásból adódó sorrendkülönbségen kívül a két halmaz tartalma azonos volt.

A nagyobb hálózat teljesítménye  $256 \times 256$  méretű bemeneti képekkel a 4.1 táblázatban látható. A kiindulási tanulási rátát 3000 iterációnként csökkentettem. A kiindulási, körülbelül 73%-os pontosság a hálózat súlyainak véletlenszerű feltöltésének köszönhető, egy hamis érték. Jól látható, hogy a hálózat pontossága 55,5% körülre konvergál.

Iterációk száma	Pontosság	Hiba
0	0,734648	1,53865
5000	0,609859	1,14415
10000	0,551549	1,23974
15000	0,55493	1,23605
20000	0,554507	1,2348
25000	0,555352	1,23617
30000	0,554648	1,23544

4.1. táblázat. A 3.2 táblázatban leírt hálózat pontossága,  $256 \times 256$  méretű előfeldolgozatlan bemeneti képekkel.

Kérdéses, hogy a hálózat méretének csökkentése milyen hatással van a teljesítményre. A kisebb hálózat teljesítménye  $256 \times 256$  pixel felbontású képekkel a 4.2 táblázatban

látható. Bár az eredmények meglepően jónak tűnnek, hamisak. A 73% körüli pontosság a teszhalmaz kiegyensúlyozatlanságának köszönhető. Egy további jel, hogy a hálózat rosszul teljesít, hogy míg a 4.1 táblázatban leírt 55,5%-os teljesítményhez hozzávetőlegesen 1,2 értékű hiba tartozott, itt egy látszólag jobb pontossághoz nagyobb hiba tartozik.

Iterációk száma	Pontosság	Hiba
0	0,0695775	1,60983
2000	0,734507	1,49123
4000	0,73507	1,48261
6000	0,733521	1,49524
8000	0,734789	1,51442
10000	0,734225	1,51191

4.2. táblázat. A 3.3 táblázatban leírt hálózat pontossága,  $256 \times 256$  méretű előfeldolgozatlan bemeneti képekkel.

A kisebb hálózat használatát így elvettem és a nagyobb (3.2 táblázat) hálózaton vizsgáltam meg a teljesítményt,  $512 \times 512$ -es méretű képeket használva. A képek nagyobb mérete miatt megfeleztam a bementi köteg méretét, így egyszerre 50 képpel tanítottam a hálózatot. Emiatt a tanulási paraméter frissítése 3000 iteráció helyett 6000 iterációra módosult. A 4.3 táblázatban látható, hogy a hálózat hibája felrobban 4000 iteráció után, így ezt a konfigurációt elvettem.

Iterációk száma	Pontosság	Hiba
0	0,15	1,56447
1000	0,735352	1,453
2000	0,177606	1,48261
3000	0,636479	1,22383
4000	0,02	87,3366
5000	0,02	87,3336

4.3. táblázat. A 3.2 táblázatban leírt hálózat pontossága,  $512 \times 512$  méretű előfeldolgozatlan bemeneti képekkel.

A fentebb felsorolt teljesítmények alapján a 4.1 táblázatban összegzett futtatást választottam ki, mint leghatékonyabbat. Megvizsgáltam, milyen hatása van az előfeldolgozott képek használatának a hálózat teljesítményére. A 3.2 táblázatban ebben a futtatásban leírt hálózat bemeneti adatai  $256 \times 256$  felbontású előfeldolgozott színes képek voltak. Nem jelentett különösebb teljesítménynövekedést az előfeldolgozás felhasználása, ezért nem használtam őket a továbbiakban.

Ezekből az adatokból látható, hogy a bemeneti képek mérete és az előfeldolgozási lépések nem számítanak akkora súllyal, mint a hálózat architektúrája és a tanulási paraméterek. A működő konfiguráció teljesítményét nem befolyásolta számottevően sem a bemeneti adatok dimenziójának megnövelése, sem egy olyan előfeldolgozás, mely az emberi szem számára sokkal jobb minőségű képet eredményez.

Végül a 3.2 táblázat hálózatát választottam az ensemble boosting módszer kiinduló-

Iterációk száma	Pontosság	Hiba
0	0,150423	1,62056
5000	0,442253	1,51196
10000	0,52662	1,48077
15000	0,529155	1,47939
20000	0,528028	1,47993
25000	0,527465	1,48005
30000	0,528873	1,47983

4.4. táblázat. A 3.2 táblázatban leírt hálózat pontossága,  $512 \times 512$  méretű előfeldolgozatlan bemeneti képekkel.

pontjának. Még mielőtt elkezdtem volna az újraklasszifikálandó minták kiválasztását, egy saját Python-szkripttel még egyszer validáltam az eredményeket. A szkript megszámlálja, mennyi képet talál el illetve téveszt el a tanulóhalmazból a hálózat, továbbá megjelöli az eltévesztetteket. Ez alapján a legjobban teljesítő hálózat teljesítménye 3712 találat a 7022 tesztképből, vagyis 52,86% 30000 iteráció után.

A tanulóhalmaz mintáira számolt hiba alapján sorba rendeztem a mintákat, majd a legrosszabb 80%-ot kiválasztottam közülük, 9886 képet. Fontosnak tartottam, hogy ne csak a félreklasszifikált képeket jelöljem meg, hanem azokat is, melyekben nem volt teljesen biztos a rendszer. Ezért az elvárt kimeneti vektor és a kapott predikcióvektorra euklideszi távolságot számoltam, ez alapján végeztem a kiválasztást.

Erre a 9886 képre futtattam újra a tanítást és teszteltem az így kapott hálózatot. A hálózat pontossága 25,16% volt a Caffe szerint 20000 iteráció után, ezzel a súllyal számítottam bele a közös véleménybe. A két hálózat közös döntésének pontossága 4002 helyesen osztályozott kép volt a 7022-ből, vagyis 57,15%. Ez mutatja, hogy a módszer képes javítani a hálózatok külön-külön vett pontosságán.

Egy harmadik hálózatot is tanítottam a fenti kiválasztással létrejövő új tanítóhalmazzal, annyi különbséggel, hogy itt az első két hálózat közös döntése alapján vettem a legrosszabb 80%-ot. Az ez alapján tanított hálózat pontossága 55,8% lett 20000 iterációt követően. A három hálózat közös pontossága 3924 találat volt a 7022-ből, ez 55,88%-nak felel meg (4.5 táblázat).

Hálózatok	Pontosság
A	52,8%
A + B	57,15%
A + B + C	55,88%

4.5. táblázat. Az ensemble boosting módszer eredményei. Bár a teljesítménynövekedés marginális, látható, hogy a hálózatok közös döntése pontosabb, mint az eredetié.

## 4.1. Kitekintés

A végső lépéseknél megfontolás tárgyává tettem, hogy kevesebb képet választok ki, hiszen a 80%-20%-os kettévágásban az újratanításra jelölt képek a helyesen osztályozottakat is tartalmazza. Ezt azért nem tettem meg, mivel az így rendelkezésre álló 9886 kép mennyiségénél kevesebb már nem lenne ideális egy helyes tanítás lefutásához. Több, helyesen felcímkézett képhez való hozzáféréssel úgy érzem növelni tudtam volna az együttműködés teljesítményét.

A tervezési fázisban illetve az eredmények kiértékelésekor felvetődtek alternatív megközelítési lehetőségei is az egyes lépéseknek. Egy lehetséges előfeldolgozási lépésként felmerült a képek csak zöld csatornáinak felhasználása, mivel a zöld fénnel kapott kép a legalkalmasabb a retinopátia diagnosztizálására.

Az ensemble megvalósításakor lehetséges lett volna másképp is regularizálni az egyes hálózatok döntését:

- Közel azonos pontosságok esetén azonos súlyokkal számíthattak volna.
- Hálózatonként csökkenő súllyal, például  $\frac{1}{n}$  vagy  $2^{-n}$ , ahol  $n$  a hálózat sorszáma.

Egy lehetséges felhasználási területe a módszeremnek a diabetikus retinopátia automatizált felismerése fejlődő országokban vagy olyan helyeken, ahol nincs meg a szükséges infrastruktúra a kézzel történő diagnózishoz. A rendszer tanítása után a paraméterek szabadon integrálhatóak például egy okostelefonon futó szoftverbe, a tanítás teljesítményigényének töredéke a klasszifikációé.

Felhasználható lehet továbbá szemészek számára figyelmeztető rendszerként, lehetséges elváltozásokra felhívva az orvos figyelmét. Az orvost nem válthatja ki teljes mértékben a rendszer, de együttműködve gyorsabb és pontosabb diagnózist tesz lehetővé.

Egyértelmű továbbfejlesztési lehetőség az ensemble magjának tekinthető első hálózat teljesítményének növelése. Az NPDR három osztálya között sima az átmenet a tüneteket tekintve, ezt figyelembe véve javítható lenne a pontosság. További módja lehet a hibák csökkentésére a képalkotási módszerből fakadó műtermékek és reflexek kiszűrése. Bár láthattuk, hogy igen flexibilisek a konvolúciós neurális hálózatok (elhanyagolható volt a tudatos konvolúcióval végzett előfeldolgozás hatása a teljesítményre), mégis úgy gondolom, hogy ez sarkalatos pontja lehet a teljesítmény növelésére törekvésben.

## 5. Összefoglalás

Sikerrel jártam egy releváns gépi tanulási probléma megoldásában. A választott módszerem hasznosnak és kevés korábbi tapasztalattal rendelkezve is jól működőnek bizonyult. A végső legnagyobb pontosság, 57,15% jóval a véletlenszerű klasszifikáció pontossága felett áll, de úgy érzem a feladat megoldása közbeni tapasztalataim alapján tovább növelhető a rendszer teljesítménye.

Megoldandó problémának a diabétikus retinopátia súlyosságának klasszifikációját választottam szemfenékfényképek alapján. A diabétikus retinopátia gyors és olcsó diagnózisának relevanciáját mutatják azon becslések, miszerint a diabetes mellitusban szenvedők száma egyre növekszik [1], a cukorbetegség mind ki vannak téve a betegség veszélyeinek. A leírt tünetek alapján igen súlyos következményei lehetnek, ha nem kerül kezelésre, akár súlyos látásromláshoz, vaksághoz is vezethet.

A feladot neurális hálózatok segítségével oldottam meg. A választásomat nagyban befolyásolta a módszer előretörése a modern grafikus processzorok megjelenésének köszönhetően. Bemutattam a módszer használatához szükséges elméleti háttérrel. A helyes működéshez elegendő volt egy címkével ellátott adathalmaz megszerzése, melyet tanító- és teszhalmazokká alakítottam.

A létrehozott tanító- és teszhalmazban tartalmazott minták számát bár keveselltem, elégségesnek bizonyultak egy gépi tanulási probléma megoldásához. A konvolúciós neurális hálózatok erejét mutatja, hogy nyers képi adatokból képes volt jó pontosságot elérni egy általános hálózat. Modularitását pedig az támasztja alá, hogy kisebb tanító-halmazokkal tanított hálózattal az eredeti hálózat teljesítményét növelni lehetett.

Két különböző hálózati architektúra teljesítményét mértem fel eltérő méretű bemeneti képekkel. A legjobban teljesítő bemeneti halmazon egy, az emberi szem számára előnyös képfeldolgozási módszerrel kíséreltem meg javítani a teljesítményen. Végül pedig a legjobban teljesítő konfigurációkból három hálózat együttes működését vizsgáltam meg.

Az eredmények alapján felmértem a lehetséges továbbfejlesztési lehetőségeket és felvettem néhány alkalmazási területet is. A szerzett tapasztalatok és tudás hozzájárult a terület mélyebb megértéséhez.

A végrehajtandó feladatokat sikeresen teljesítettem.

## Referenciák

- [1] Yuankai Shi, Frank B Hu, “The global implications of diabetes and cancer”, *The Lancet*, 383. köt. 9933. sz., pp. 1947–1948, 2014. jún.
- [2] J. W. Yau, S. L. Rogers, R. Kawasaki, E. L. Lamoureux, J. W. Kowalski, T. Bek, S.-J. Chen, J. M. Dekker, A. Fletcher, J. Grauslund, et al., “Global prevalence and major risk factors of diabetic retinopathy”, *Diabetes care*, DC\_111909, 2012.
- [3] P. Geraldès, J. Hiraoka-Yamamoto, M. Matsumoto, A. Clermont, M. Leitges, A. Marette, L. P. Aiello, T. S. Kern, és G. L. King, “Activation of pkc-delta and shp-1 by hyperglycemia causes vascular cell apoptosis and diabetic retinopathy”, *Nature medicine*, 15. köt. 11. sz., pp. 1298–1306, 2009.
- [4] C. Wilkinson, F. L. Ferris, R. E. Klein, P. P. Lee, C. D. Agardh, M. Davis, D. Dills, A. Kampik, R. Pararajasegaram, J. T. Verdaguer, et al., “Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales”, *Ophthalmology*, 110. köt. 9. sz., pp. 1677–1682, 2003.
- [5] M. D. Abràmoff, M. K. Garvin, és M. Sonka, “Retinal imaging and image analysis”, *Biomedical Engineering, IEEE Reviews in*, 3. köt., pp. 169–208, 2010.
- [6] Elérhető: <https://nei.nih.gov/health/diabetic/retinopathy> (elérés dátuma 2015. 12. 1.).
- [7] A. Colenbrander, “Duane’s clinical ophthalmology”, Lippincott Williams & Wilkins, 2006., fejr. 63.
- [8] The basics: Direct and indirect ophthalmoscopy, Elérhető: <http://www.elsevier.com/samplechapters/9780702042973/9780702042973.pdf> (elérés dátuma 2015. 12. 1.).
- [9] Elérhető: <http://www.oculist.net/downat0502/prof/ebook/duanes/pages/v1/v1c063.html> (elérés dátuma 2015. 12. 1.).
- [10] Patrick J. Saine, Marshall E. Tyler, *Ophthalmic Photography: Retinal Photography, Angiography, and Electronic Imaging*, Second Edition. Butterworth-Heinemann Medical.
- [11] Elérhető: <http://www.opsweb.org/?page=fundusphotography> (elérés dátuma 2015. 12. 1.).
- [12] T. J. Bennett, *Monochromatic fundus photography*. Elérhető: <http://www.opsweb.org/?page=Monochromatic> (elérés dátuma 2015. 12. 1.).
- [13] Antal Ákos, Kály-Kullai Kristóf, Farkas Henrik, “A napsugárzás spektruma és az emberi szem érzékenysége”, *Fizikai szemle*, 6. köt., pp. 199–203, 2005.

- [14] Description - diabetic retinopathy detection | kaggle, Elérhető: <https://www.kaggle.com/c/diabetic-retinopathy-detection> (elérés dátuma 2015. 12. 1.).
- [15] R. W. Brause, “Medical analysis and diagnosis by neural networks”, pp. 1–13, 2001.
- [16] Naive bayes classifier, Elérhető: <http://www.ic.unicamp.br/~rocha/teaching/2011s2/mc906/aulas/naive-bayes-classifier.pdf> (elérés dátuma 2015. 12. 3.).
- [17] A. Endo, T. Shibata, és H. Tanaka, “Comparison of seven algorithms to predict breast cancer survival (< special issue> contribution to 21 century intelligent technologies and bioinformatics)”, *Biomedical fuzzy and human sciences: The official journal of the Biomedical Fuzzy Systems Association*, 13. köt. 2. sz., pp. 11–16, 2008.
- [18] D. J. Spiegelhalter, A. P. Dawid, S. L. Lauritzen, és R. G. Cowell, “Bayesian analysis in expert systems”, *Statistical science*, pp. 219–247, 1993.
- [19] E. R. Davies, *Machine Vision Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers, 2005.
- [20] Supervised learning: Predicting an output variable from high-dimensional observations, Elérhető: [http://scikit-learn.org/stable/tutorial/statistical\\_inference/supervised\\_learning.html#the-curse-of-dimensionality](http://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html#the-curse-of-dimensionality) (elérés dátuma 2015. 12. 3.).
- [21] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, *Bevezetés az adatbányászatba*. Panem Kft., 2011.
- [22] Carlsten Steger, Markus Ulrich, Christian Wiedermann, *Machine Vision Algorithms and Applications*. WILEY-VCH, 2008.
- [23] Elérhető: [http://www.tankonyvtar.hu/hu/tartalom/tamop425/0046\\_adatbanyaszat/ch05s05.html](http://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_adatbanyaszat/ch05s05.html) (elérés dátuma 2015. 12. 1.).
- [24] T. Subashini, V. Ramalingam, és S. Palanivel, “Breast mass classification based on cytological patterns using rbfn and svm”, *Expert Systems with Applications*, 36. köt. 3. sz., pp. 5284–5290, 2009.
- [25] Altrichter, Horváth, Pataki, Strausz, Takács, és Valyon, *Neurális hálózatok*. Panem Kiadó, 2006.
- [26] Elérhető: [https://www.mozaweb.hu/Lecke-BIO-Biologia\\_10-Az\\_allati\\_sejt\\_es\\_a\\_fobb\\_szovettipusok\\_jellemzoi\\_III-104960](https://www.mozaweb.hu/Lecke-BIO-Biologia_10-Az_allati_sejt_es_a_fobb_szovettipusok_jellemzoi_III-104960) (elérés dátuma 2015. 12. 1.).
- [27] Elérhető: <https://www.hiit.fi/u/ahonkela/dippa/node41.html> (elérés dátuma 2015. 12. 1.).
- [28] M. Collins. Convergence proof for the perceptron algorithm.
- [29] Elérhető: <http://www.nonlinearbiomedphys.com/content/2/1/3/figure/F2?highres=y> (elérés dátuma 2015. 12. 1.).



- [30] K. Hornik, “Approximation capabilities of multilayer feedforward networks”, *Neural Networks*, 4. köt., pp. 251–257, 1991.
- [31] Elérhető: [http://www.holehouse.org/mlclass/10\\_Advice\\_for\\_applying\\_machine\\_learning.html](http://www.holehouse.org/mlclass/10_Advice_for_applying_machine_learning.html) (elérés dátuma 2015. 12. 1.).
- [32] Y. Bengio, P. Simard, és P. Frasconi, “Learning long-term dependencies with gradient descent is difficult”, *Neural Networks, IEEE Transactions on*, 5. köt. 2. sz., pp. 157–166, 1994.
- [33] Y. LeCun, L. Bottou, Y. Bengio, és P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, 86. köt. 11. sz., pp. 2278–2324, 1998.
- [34] S. Dieleman. (2014. árp.). Winning the galaxy challenge with convnets: Sander dieleman, Elérhető: <http://blog.kaggle.com/2014/04/18/winning-the-galaxy-challenge-with-convnets/> (elérés dátuma 2015. 12. 1.).
- [35] J. Schmidhuber, “Deep learning in neural networks: An overview”, *Neural Networks*, 61. köt., pp. 85–117, 2015.
- [36] D. H. Hubel és T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”, *The Journal of physiology*, 160. köt. 1. sz., p. 106, 1962.
- [37] J. D. Fauw. (2015.). Detecting diabetic retinopathy in eye images, Elérhető: <http://blog.kaggle.com/2015/08/10/detecting-diabetic-retinopathy-in-eye-images/> (elérés dátuma 2015. 12. 1.).
- [38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, és R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting”, *The Journal of Machine Learning Research*, 15. köt. 1. sz., pp. 1929–1958, 2014.
- [39] Elérhető: <https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcS07M2F48TDRuxqxRDxRG9XJwG8C3Xc01s6W2xSy7f7E7x0Jm62> (elérés dátuma 2015. 12. 1.).
- [40] A. Krizhevsky, I. Sutskever, és G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, 2012., pp. 1097–1105.

# Függelék A.

## Mellékletek listája

A mellékelt CD tartalma az alábbi:

- jelen szakdolgozat elektronikus példánya.
- A *szkriptek* mappában a feladat végrehajtásához használt Python- és MATLAB-szkriptek. A szkriptek feladata:
  - *classifier.py*: egy, vagy több hálózat pontosságát vizsgálhatjuk meg vele a tesztalmazon.
  - *ensemblewloss.py*: egy, vagy több hálózattal klasszifikálja a tanulóhalmazt és a hiba alapján újratanulásra jelöli a minták egy részét.
  - *resizer.py*: az általunk megadott listafájlban levő képeket átméretezi  $512 \times 512$  pixel méretűre.
  - *extender3.py*: egy általunk megadott listafájlban levő képeket forgatja el  $90^\circ$ ,  $180^\circ$  és  $270^\circ$ -kal.
  - *mirrflip.py*: egy általunk megadott listafájlban levő képeket forgatja el  $270^\circ$ -kal.
  - *LCAS.m*: egy mappa tartalmán hajtja végre a *local colour average subtraction* transzformációt és átméretezi  $512 \times 512$  méretűre.
- A *listafajlok* mappában a kettébontott adatszett és létrehozott tanuló- és tesztalmaz listafájljai találhatóak meg.
- A *naplófajlok* mappa az egyes futtatások naplófájljait tartalmazza.