

[Külső fekete borítólap formátuma]

Széchenyi István Egyetem  
Gépészmérnöki, Informatikai és Villamosmérnöki Kar  
Informatika Tanszék

# **SZAKDOLGOZAT**

**Török Kristóf**  
Mérnökinformatikus BSc szak

[beadás éve]

[Gerincen:] Hallgató Neve, Évszám {Titkosított}
---



**SZÉCHENYI  
EGYETEM**  
UNIVERSITY OF GYŐR



**INFORMATIKA  
TANSZÉK**  
DEPARTMENT OF COMPUTER SCIENCE

# **SZAKDOLGOZAT**

## **Fotóalbum szelektáló alkalmazás**

**Török Kristóf**

**Mérnökinformatikus BSc szak**

**[beadás éve]**

## Nyilatkozat

Alulírott, [Hallgató Neve (Neptun-kód), Hallgató szakja, képzési szintje] szakos hallgató kijelentem, hogy a [Szakdolgozat Címe] című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, [beadás dátuma]

---

hallgató

# Kivonat

[A dolgozat címe]

[1 oldalas, magyar nyelvű tartalmi kivonat]

## Abstract

[Title in English]

[1 oldalas, angol nyelvű kivonat]

# Tartalomjegyzék

1	Bevezetés.....	1
2	Elméleti háttér.....	2
2.1	Képközpontosítás hibák .....	2
2.1.1	Exponálás .....	2
2.1.2	Elmosódások.....	3
2.1.3	Purple Fringing Aberration.....	5
2.2	Probléma bemutatása .....	6
2.3	Kép transzformációk.....	8
2.3.1	Digitális képek hisztogramja, kiegyenlítése .....	8
2.3.2	Laplacian operátor .....	9
2.3.3	Canny éldetektor.....	13
2.4	Gépi tanulás .....	15
2.4.1	Áttekintés.....	15
2.4.2	Megközelítések.....	16
2.4.3	Előre tekintés .....	16
2.5	Neurális hálózatok .....	16
2.5.1	Felépítés.....	17
2.5.2	Egyrétegű vagy többrétegű előrecsatolt neurális hálózatok .....	17
2.5.3	Visszacsatolt neurális hálózatok .....	19
2.5.4	Tanítás.....	20
2.6	Konvolúciós neurális hálózatok.....	21
2.7	Hasonló munkák .....	23
2.7.1	Túlexponálás korrigálása és detektálása.....	23
2.7.2	Elmosódások korrigálása és detektálása.....	23
3	Tervezés.....	23
3.1	Fejlesztő környezet, programnyelv, programcsomagok .....	23
3.2	Funkcióterv .....	24
4	Irodalomjegyzék .....	24
5	Mellékletek .....	26

# 1 Bevezetés

A fotózás az elmúlt években egyre nagyobb népszerűsége tett szert. A digitális fényképezőgépek és az egyre jobb kamerákkal felszerelt okostelefonok széles körű megjelenése és sokoldalúsága vezetett ahhoz, hogy az emberek szeretnék életük minden emlékezetes pillanatát fotó formájában megörökíteni. Születésnap, nyaralás vagy más esemény megörökítése érdekében az emberek gyakran készítenek fotó sorozatokat, abban a reményben, hogy néhány jó fotót, amely megfelel az igényeknek, később kiválogathatnak szerkesztésre, közzétételre vagy megosztásra. A fotó sorozatok megjelenését tovább fokozzák az olyan megoldások, mint például az okostelefonokon elérhető sorozatfelvétel mód, amelynek segítségével a felhasználó több szinte azonos vagy hasonló képet készíthet. Ezen megoldásoknak köszönhetően a felhasználók rendkívül hatalmas fotógyűjteményekhez juthatnak, ahol ugyanazon pillanatról vagy látványvilágról több kép áll rendelkezésre. A hatalmas fotógyűjtemények szelektálása és kezelése, fárasztó és időigényes folyamat. Nehéz döntések sorozatát hozza magával, mint például, hogy melyik képet töröljük és melyiket tartjuk meg.

Jelenleg sok alkalmazás, mint például az iPhoto és a Picasa, lehetővé teszi a felhasználó számára, hogy rendszerezze a fotó sorozatokat idő vagy földrajzi információk alapján. Ezzel szemben sok online fotómegosztó közösségi oldal, például Facebook és a Google Fotók arcokat és más magasabb szintű tartalmakat vonnak ki címkézési célokra. Ezek az eszközök javítják az albumok elrendezését, viszont egyikük sem nyújt olyan lehetőséget a felhasználók számára, amely segítséget ad a rossz képek automatikus kiválogatására vagy a hasonló képeket tartalmazó gyűjtemények között a legjobb képek megtalálására. Jelenleg a felhasználóknak még mindig saját maguknak kell foglalkozniuk ezzel a problémával.

## 2 Elméleti háttér

### 2.1 Képalkotási hibák

Számos oka lehet annak, hogy egy fénykép rossz minőségű. Most áttekintem melyek azok a képalkotási hibák, amelyek miatt a képek sokszor nem olyan minőségűek, amilyeneket szerettünk volna készíteni. A leggyakoribb képalkotási hibák közé tartozik a rossz megvilágítás és a különféle elmosódások. [1]

A **rossz megvilágítás** olyan képekre vonatkozik, amelyeket a megfelelő fény elérése nélkül készítettek, Ez azt eredményezheti, hogy a képek tompának vagy sötétnek tűnnek. A rossz megvilágítással rendelkező képeket másnéven **túlexponálnak** vagy **alulexponálnak** is nevezzük. [1]

**Elmosódások** nagyon gyakoriak és különféle tényezők miatt keletkezhetnek fényképezés során. Ezek a tényezők lehetnek például a tárgy mozgása, a kamera remegése vagy a fényképezőgép lencséje nincs fókuszban. [2]

A rossz minőségű fényképeken gyakran találkozhatunk még **Purple Fringing Aberration** jelenséggel, amelyek általában az alacsony felszereltségű kamerák által készített fényképeken fordulnak elő. Ez az optikai torzulás általában a széles spektrumú megvilágítás fényes területei mellett lévő sötét szélek színeződéseként és világosításaként látható, mint például a nappali fény vagy a különböző típusú gázkisüléses lámpák. [11]

#### 2.1.1 Exponálás

Az exponálás szabályozza, hogy mennyi fény juthat a kamera érzékelőjére. Ebből adódóan a fénykép világosságát a megjelenített fény mennyisége határozza meg. A fotózásban, a „dinamikus tartomány” – „dynamic range” kifejezést használják a mérhető legsötétebb és legfényesebb fényintenzitások közötti arányának leírására. [3] [4]

Az átlagos fényképezőgépek dinamikus tartománya nagyon korlátozott, általában 100:1-hez, ami sokkal kevesebb mint a való életben. A nagy kontraszttal rendelkező helyszínek, mint például napfény alatti kültéri környezet, nagyon magas dinamikus tartománnyal rendelkezhetnek, 105-től 109-ig. Az ilyen helyszínek esetében nagyon nehéz minden területet jól megvilágítani. Emiatt a túlexponálás keletkezik. Túlexponálásnak nevezzük, ha a fénykép egyes világos részein a kiemelt részletek elvesznek. Ez a jelenség akkor fordul elő, amikor a kamera érzékelőjére hulló fény meghaladja azt, amit az érzékelő képes



rögzíteni. Túlexponálás történik nagyon gyakran a mindennapi fotózásban a helyszínek nagy dinamika tartománya (HDR) miatt. [3]

Előfordulhat, hogy a fényképeken fényes területek helyett sötét területekkel találkozunk, amelyek szinte megkülönböztethetetlenek a fekete színtől. Ennek oka, hogy nem jut elég fény a fényképezőgép érzékelőjére. Ezeket a területeket alulexponáltnak nevezzük. A következő ábrán két fényképet láthatunk, amelyeken túlexponált és alulexponált területek vannak:



1. ábra: Túlexponált (bal) és alulexponált (jobb) kép [4]

Gyakorlatban a fotósok néha megpróbálják csökkenteni az exponálási értéket azért, hogy megakadályozzák a túlexponálást. Azonban az exponálási érték túlzott csökkentése elhomályosítja a fényképet. [3]

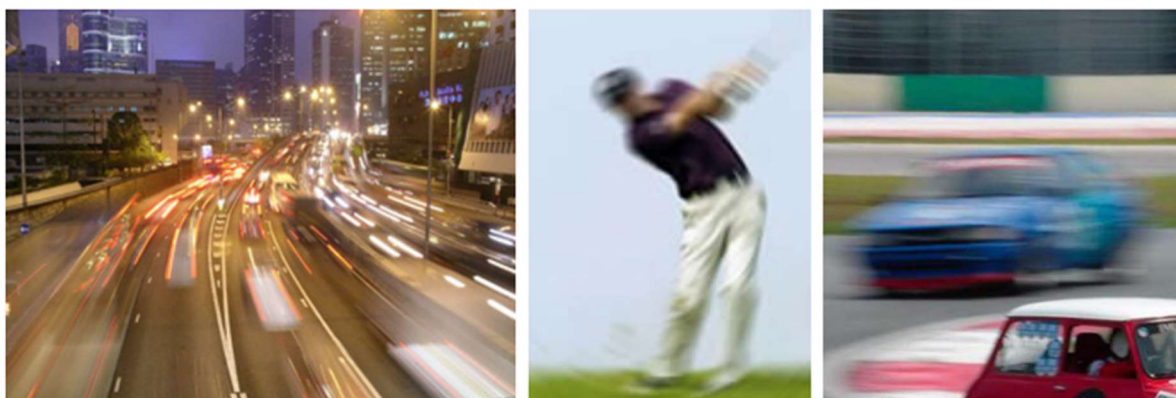
Túlexponálás javítása vagy elkerülése többféle módon lehetséges. Néhány nagy dinamika tartományú képrögzítéssel foglalkozó munkában, a teljes dinamikus tartomány rögzítését célozza meg. Tónustérképezési technikákkal az alacsony dinamika tartományú képeket az alacsony dinamika tartományú (LDR) képekhez hozzárendelik, elkerülve ezzel a túlexponálást. Ez azonban nagyon drága megoldás, a HDR kamerák magas árai miatt. Drága kamerák helyett vannak más létező HDR megoldások is, viszont általában ezek több felvételt igényelnek, különböző exponálási értékekkel. Az ilyen többszörös bemeneti módszerek korlátozóak, mert megkövetelik, hogy a helyszín statikus legyen. Ezenkívül a HDR rögzítés csak új fényképek esetében működik. Nem tudja kijavítani a túlexponálást a már meglévő fényképeken. [3]

### **2.1.2 Elmosódások**

Elmosódásokkal gyakran találkozhatunk a fényképeken, amelyeket a mozgás, kézremegés vagy a helytelen fókuszálás okoz. Azonban ezek az elmosódások lehetnek szándékosak

azért, hogy a fotós a fotó kifejezőerejét erősítse ezáltal. Ilyen esetekben az elmosódás nem csökkenti a kép minőségét, azonban ez az esetek kisebbik halmaza. [5]

A leggyakoribb elmosódások közé tartozik a **mozgási elmosódás** (motion blur). Egy expozíciós perióduson belül a kamera vagy tárgyak mozgása elmosódott képeket eredményez, mivel a megvilágítás változásai az idő múlásával integrálódnak és az élesség elmosódik. Másfelől viszont a mozgás miatt elmosódott kép megtartja a mozgásról szóló információt, amely paraméterezi az elmosódást. Így támpontokat ad ahhoz, hogy a mozgást visszanyerjük egyetlen képből. [6]



2. ábra: Példák térváltozó mozgási elmosódásra [6]

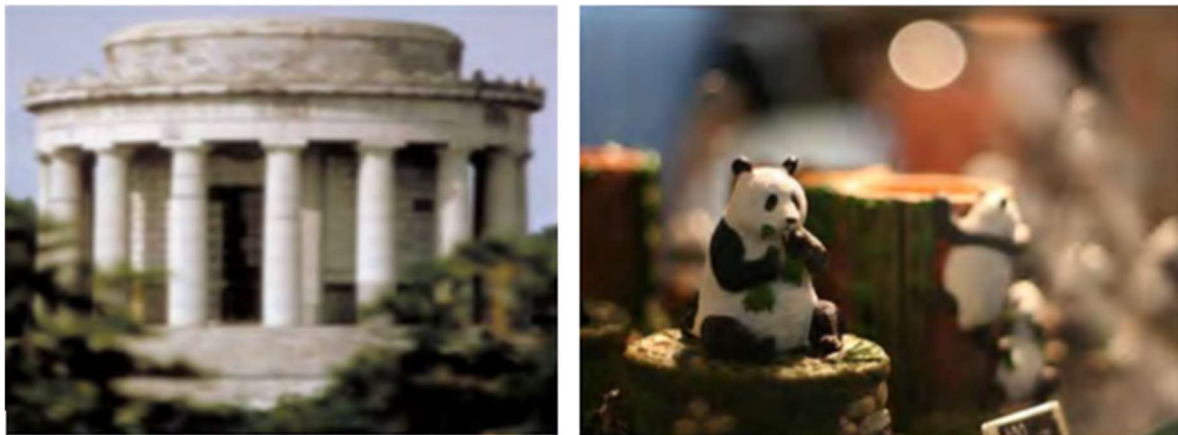
A mozgási elmosódást a pontterjedési függvény (Point Spread Function – PSF) jellemzi, amelynek paraméterei szorosan kapcsolódnak a mozgáshoz. A legegyszerűbb mozgási elmosódás a térinvariáns mozgási elmosódás. Ezt a fajta mozgást széles körben tanulmányozták. A gyakorlatban azonban, mivel a mozgás meglehetősen összetett lehet, a mozgásfoltok sokkal bonyolultabbak lehetnek ennél az egyszerű esetnél. Például az elmosódás lehet térváltozó, nemlineáris, lokális és többszörös. Több tanulmány is foglalkozik ezekkel az esetekkel, azonban a jelenlegi megoldások közül, azok a legjobbak, amelyek több bemeneti képre, felhasználói interakciókra vagy extra feltételezésekre támaszkodnak. [6]

A kamerarendszerek helytelen fókuszálása miatt létrejövő képi elmosódások elsősorban a geometriai képalkotás bizonyos problémájából és a kameraobjektívek véges mélységélességéből adódnak. Ezek a képelmosódások, amelyek a fókuszálási hiba vagy a nem megfelelő fókuszálással rendelkező képalkotó lencserendszer miatt következnek be, komoly képromlást okoznak. [7]



3. ábra: Példák lokális mozgási elmosódásra [7]

Ezen elmosódások kijavítására számos képfeldolgozási technikát fejlesztettek ki. Ezen megoldások többsége a képfelvételi rendszerhez kapcsolódó pontterjedési függvényt (Point-Spread Function - PSF) becsüli, amely a kép helyreállítása érdekében a becsült elmosódási függvény alapján térinvariáns dekonvolúciót végez. [7]



4. ábra: Teljesen elmosódott (bal) és részben defókuszált (jobb) kép [8]

### 2.1.3 Purple Fringing Aberration

**Purple Fringing Aberration** egy gyakran előforduló színeltérés a digitális fényképezőgépek által készített képeken, amely lila fátyolként jelenik meg az objektumok körül. A lila rojtos régió elveszíti az eredeti színét és a terület torzított színnel jelenik meg. Mivel napjainkban egyre kisebbek a digitális fényképezőgépek, amelyek támogatják a kiváló minőségű kimenetet, a lila rojtosodás komoly problémává vált. [9]

Általában a lila rojtosodás a kromatikus aberrációnak tulajdonítható, azonban ez nem minden esetben igaz. A fénypontok körüli hasonló rojtosodás a lencse becsillanásából is adódhat. A fénypontok vagy sötét területek körüli színes rojtosodás oka lehet, hogy a

különböző színek receptorainak dinamikatartománya vagy érzékenysége eltérő. A rojtosodás másik oka a kromatikus aberráció a CCD pixelenként több fény összegyűjtésére használt nagyon kicsi mikrolencsékben. Mivel ezek a lencsék a zöld fény helyes fókuszálására vannak beállítva, a vörös és kék fény helytelen fókuszálása lila rojtosodást eredményez a fénypontok körül. Ez a probléma egyenletes az egész képkockán és nagyobb problémát jelent a nagyon kis pixelosztású CCD-kenél. [9]



5. ábra: Példaképek Purple Fringing Aberration

A lila rojtosodás kezelésére számos módszert mutattak be. Hardveres megközelítésként a kromatikus aberráció kiküszöbölésére optimális lencseterveket javasoltak. Ezek azonban a kromatikus aberráció eltávolítására koncentrálnak és a lencsék tervezése magas költségekkel jár. A lila rojtos képek utólagos feldolgozására digitális képfeldolgozási módszereket is javasolnak. Amelyek általában két lépésből állnak. A lila területek detektálásából és az azt követő korrekciójából, amelyekben a korrekciós lépések korlátozott teljesítményt mutattak. [9]

## 2.2 Probléma bemutatása

A fotózás az elmúlt években egyre népszerűbb lett. Az emberek szeretnék életük minden emlékezetes pillanatát fotók formájában dokumentálni. Korábban a fényképezőgépek és a tárolási kapacitások elérhetősége miatt elképzelhetetlen volt, viszont napjainkban gyakran



készítenek több képet ugyanazon látványvilágról vagy pillanatról. Ezt tovább fokozzák az olyan megoldások, mint például több okostelefonon elérhető sorozatfelvétel mód, amely több tucat szinte azonos képet készít annak érdekében, hogy minél jobb minőségű képet állítsanak elő az adott pillanatról. Ennek eredményeként a felhasználók végül rendkívül nagy fotó gyűjteményhez juthatnak, ahol minden pillanatról vagy látványvilágról több kép áll rendelkezésre. Emiatt fotósok és egyszerű felhasználók sok időt töltenek el azzal, hogy kiválasszák a legjobb képet a sok hasonló közül. Ez a folyamat gyakran nehézkes és időigényes. [10]

Manapság egyre több kutatási erőfeszítés történik a képminőség esztétikai szempontból történő értékeléséről. Ezek a módszerek irányíthatják a képek kiválasztását, azonban ezeket alig használják fotósorozatokra, amelyek teljesen ugyanazt a látványvilágot ábrázolják. Általában ezek változatos tartalmú képek egy nagy általános korpuszára vannak kiképezve és szoros értékelést adnak hasonló képekhez. Ennek ellenére a vizuális különbségek a fotósorozatok között nagyon finomak, emiatt a létező módszerek a képek esztétikai értékelésére, kevésbé képesek fotósorozatok kiválasztására. [10]

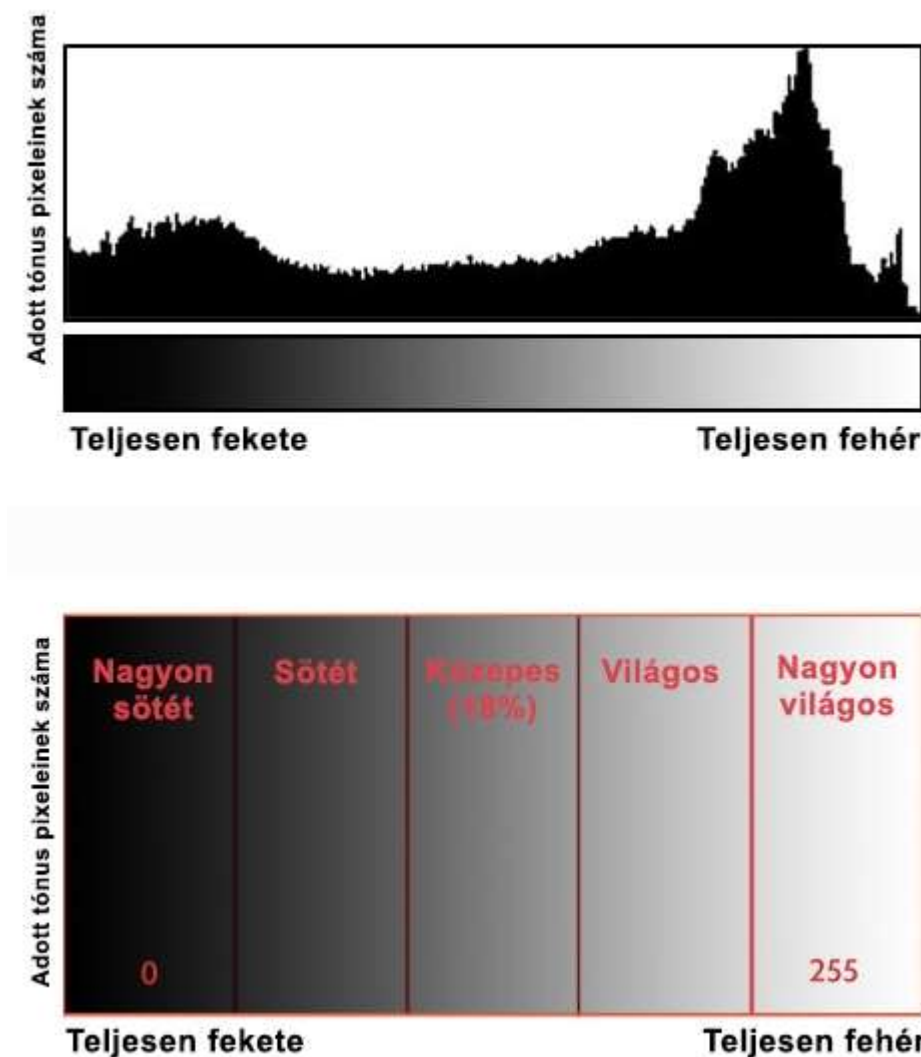
Máig több módszert javasoltak fotósorozatok kiválasztásának megkönnyítésére. Kuzovkin és társai [11] [12] meghatározták az egyes fotók többszintű összefüggéseit hierarchikus csoportosítással. Chang és társai [13] összegyűjtötték az első nyilvános adatkészletet, amely személyes fotóalbumok fotósorozatait tartalmazza és bemutatott egy végpontok közötti mélytanulás módszert Siamase hálózattal. Az előbb említett tanulmányokban elért kezdeti áttörés ellenére továbbra is nagy kihívást jelent a megkülönböztető jellemzők megállapítása a finom különbségek azonosítása érdekében a fotósorozatok képei között. [10]

Ezért a célom egy olyan szoftver megvalósítása, amely alkalmas lehet fotósorozatok automatizált feldolgozására. Amely képes felismerni ugyanazon látványvilágról készült több képet és ezek közül kiválasztani a legjobb felvételeket. Amennyiben szükség és lehetőség van rá akkor több kép felhasználásával állítson elő jobb felvételt. Felismeri a képalkotási hibákat (elmosódás, túlexponálás) és ha lehetséges javít ezek minőségén, illetve ahol a képminőség egy meghatározott szint alatti akkor javaslatot tesz a felhasználó számára, hogy törölje a képet. [10]

## 2.3 Kép transzformációk

### 2.3.1 Digitális képek hisztogramja, kiegyenlítése

A kép hisztogramja, más hisztogramokhoz hasonlóan, szintén a gyakoriságot mutatja. A képhisztogram azonban a pixelek intenzitásértékeinek gyakoriságát mutatja. A képhisztogramban megfigyelhető a szürkeségi intenzitások mértéke és ezen intenzitások gyakorisága. [14]



6. ábra: Egy kép hisztogramja

<https://fotoalap.hu/a-histogram-ertelmezese-es-hasznalata/>

A hisztogram vízszintes tengelye a pixel értékek tartományát mutatja. Mivel a kép 8 bpp-es (bits per pixel), ez azt jelenti, hogy 256 szürkeárnyalatot tartalmaz. Ezért az vízszintes tengely tartománya 0-tól indul és 255-nél ér véget. Míg az függőleges tengelyen ezeknek az

intenzitásoknak a száma látható. [14]

Ha a hisztogramban néhány képpont a minimum vagy maximum szint közelében van, akkor azt a kép tartósan lefokozott régiójának nevezhetjük. Így azt mondhatjuk, hogy ha a képen sok nagyon világos pixel van (másnéven fehér), akkor a kép túlexponált és hasonlóképpen, ha nagy mennyiségű sötét pixel van (fekete), akkor a kép alulexponált. [14]

A szakirodalomban gyakran feltételezik, hogy az ideális hisztogram diagramjának hasonlónak kell lennie a normális eloszlás diagramjához. Ez azt jelenti, hogy nem tartalmazhat "sávdiagramokat", sem a balszélén (alulexponált), sem a jobb szélén (túlexponált). Az ábrán egy ilyen hisztogram példája látható. A kutatások rámutatnak arra, hogy a hisztogram területe három régióra osztható: világos, sötét és egyéb szürkeárnyalatok. Analóg módon az exponálás is három kategóriába sorolható: alulexponálás (sötét pixelek túlsúlya), túlexponálás (világos pixelek túlsúlya) és a megfelelő exponálás. Természetesen hangsúlyozni kell, hogy a hisztogramdiagramnak nem kell mindig ideális mutatónak lennie a kép minőségének értékelésére. [14]

A hisztogramokat sokféleképpen használják a képfeldolgozásban. Az első felhasználási mód, ahogy azt fentebb is tárgyaltuk, a kép elemzése. Egy képről már a hisztogram megnézésével is jósolhatunk. Ez olyan, mintha egy test csontjának röntgenfelvételét néznénk. Második felhasználási módja a fényerősség. A hisztogramokat széles körben alkalmazzák a kép fényerejében. Nemcsak a fényerősség, hanem a hisztogramokat a kép kontrasztjának beállítására is használják. Ezenkívül másik fontos felhasználási módja a kép kiegyenlítése. Végül, de nem utolsósorban a hisztogramot széles körben használják a küszöbértékek meghatározásánál. Ezt leginkább a számítógépes látásban használják. [14]

### **2.3.2 Laplacian operátor**

A Laplacian-operátor egy derivált operátor, amelyet a kép éleinek megtalálására használnak. A Laplacian és más operátorok, mint például a Prewitt, Sobel, Robinson és Kirsch operátorok között az a fő különbség, hogy ezek mind első rendű derivált maszkok, de a Laplacian egy másodrendű derivált maszk. Ebben a maszkban két további osztályozás van, az egyik a pozitív Laplacian-operátor, a másik pedig a negatív Laplacian-operátor. [15]

Egy másik különbség a Laplacian és a többi operátor között az, hogy a többi operátortól eltérően a Laplacian nem veszi ki az éleket egy adott irányba, hanem a következő osztályozásban veszi ki az éleket. [15]

- Befelé irányuló élek
- Kifelé irányuló élek

### **Pozitív Laplacian operátor**

A pozitív laplacian operátorban van egy standard maszk, amelyben a maszk középső elemének negatívnak kell lennie, és a maszk sarokelemeinek nullának kell lennie. [15]

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (1)$$

A pozitív Laplacian operátort arra használják, hogy kivegyék a külső éleket a képből. [15]

### **Negatív Laplacian operátor**

A negatív Laplacian operátorban is van egy standard maszk, amelyben a középső elemnek pozitívnak kell lennie. A sarokban lévő összes elemnek nullának, a maszk többi elemének pedig -1-nek kell lennie. [15]

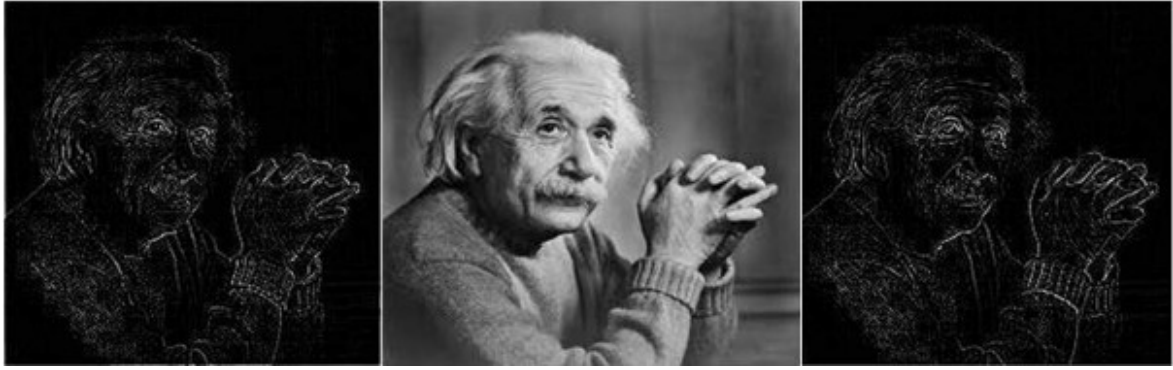
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (2)$$

A negatív Laplacian operátort pedig arra használjuk, hogy kivegyük a befelé irányuló éleket a képből. [15]

Fontos dolog, hogy hogyan alkalmazzuk ezeket a szűrőket a képre. Ugyanarra a képre nem alkalmazhatjuk a pozitív és a negatív Laplacian operátort. Egy képre csak az egyiket szabad alkalmazni és azt kell megjegyeznünk, hogy ha a pozitív operátort alkalmazzuk a képre, akkor az eredményül kapott képet kivonjuk az eredeti képből, hogy megkapjuk az élesített képet. Hasonlóképpen, ha a negatív operátort alkalmazzuk, akkor az eredményül kapott képet hozzá kell adnunk az eredeti képhez, hogy megkapjuk az élesített képet. [15]

A következő ábrán látható, hogy a szűrők alkalmazása után, hogyan kapunk befelé és kifelé irányuló éleket egy képből:





7. ábra: Példakép (közép) a pozitív operátor alkalmazása után kapott kép (bal) és a negatív operátor alkalmazása után kapott kép (jobb) [16]

### Működése

A Laplacian egy derivált operátor, használata kiemeli a kép szürkességi szintjeinek megszakadását, és megpróbálja a lassan változó szürkességi szintekkel rendelkező régiókat hangsúlytalanabbá tenni. Ez a művelet olyan képeket eredményez, amelyeken szürkés színű élvonalak és egyéb diszkontinuitások vannak sötét háttéren. Ezáltal a képen befelé és kifelé irányuló élek keletkeznek. [15]

Tulajdonképpen bizonyos értelemben a Laplacian operátor egyfajta mérőszám, hogy mekkora a minimum pont  $x, y$ . Ez úgy viselkedik, mint a második derivált többváltozós függvény skalárértékkel. [15]

A második derivált használható egy függvény helyi szélsőértékeinek meghatározására. Ha egy függvénynek van egy olyan kritikus pontja  $f'(x) = 0$ , amelyen a második derivált pontja  $f''(x) < 0$ , azaz negatív, akkor  $f$ -nek a pontban van a lokális minimuma. [15]

A Laplacian operátor a következőképpen adható meg:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} \quad (3)$$

Az  $x$  irányú parciális derivált pedig a következőképpen adható meg:

$$\frac{\partial^2 f}{\partial^2 x} = f(x + 1, y) + f(x - 1, y) - 2f(x, y) \quad (4)$$

Hasonlóképpen, az  $y$  irányú parciális deriváltat a következőképpen adhatjuk meg:

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad (5)$$

És kombinálva ezeket az egyenleteket a parciális deriváltakra, megkaphatjuk:

$$\nabla^2 f = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] - 4f(x, y) \quad (6)$$

Most tekintsük meg, hogyan néz ki egy szűrő. Egy kép 3x3 szűrője ábrázolható az (x,y) koordinátákhoz, és a következőképpen ábrázolható:

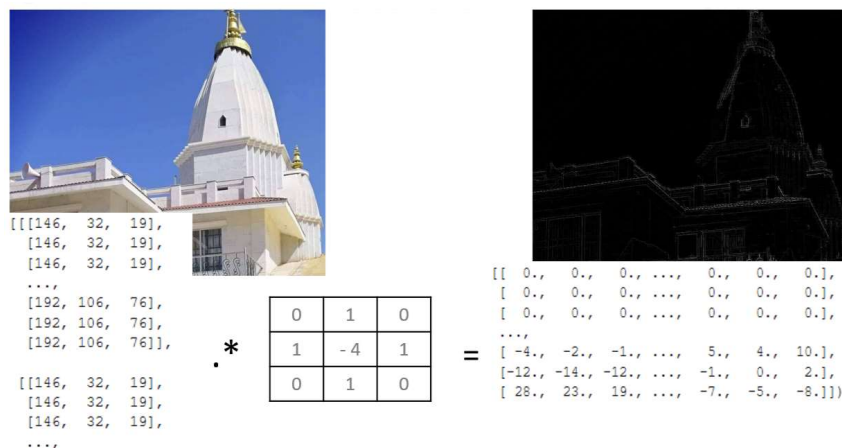
$x - 1, y - 1$	$x, y - 1$	$x + 1, y - 1$
$x - 1, y$	$x, y$	$x, y + 1$
$x - 1, y + 1$	$x, y + 1$	$x + 1, y + 1$

Ha az egyenlet együtthatóit a képbe helyezzük, a már korábban ismertetett mátrixot(szűrőt) kapjuk:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (7)$$

Ezt a mátrixot a képpel összevonva a Laplacian alapján transzformált képet ad. Ezt a mátrixot arra használjuk, hogy megtaláljuk a gyors változások területét a képeken. Alapvetően az operátor a kép második deriváltját veszi egy magasabb dimenziós síkon. Ha a kép alapvetően egységes, az eredmény nulla lesz. Ahol változás következik be, ott az eredő mátrix pozitív elemeket tartalmaz a sötétebb oldalon, és negatív elemeket a világosabb oldalon. [15]

A kép konvolúciós művelete a következő eredményt adja:



The diagram illustrates the convolution operation. On the left, a 3x3 image patch is shown with pixel values:  $\begin{bmatrix} 146 & 32 & 19 \\ 146 & 32 & 19 \\ 146 & 32 & 19 \end{bmatrix}$ . In the middle, the 3x3 Laplacian kernel is shown:  $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ . On the right, the resulting 3x3 output is shown:  $\begin{bmatrix} 0. & 0. & 0. & \dots & 0. & 0. & 0. \\ 0. & 0. & 0. & \dots & 0. & 0. & 0. \\ 0. & 0. & 0. & \dots & 0. & 0. & 0. \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ -4. & -2. & -1. & \dots & 5. & 4. & 10. \\ -12. & -14. & -12. & \dots & -1. & 0. & 2. \\ 28. & 23. & 19. & \dots & -7. & -5. & -8. \end{bmatrix}$ .

8. ábra: Konvolúciós művelet Laplace operátorral egy képen [15]

### 2.3.3 Canny éldetektor

A Canny-élességérzékelő egy olyan érzékelő operátor, amely egy többlépcsős algoritmust használ a képek széles körű érzékelésére. Ezt az érzékelést J. Canny 1986-ban fejlesztette ki. A Canny érzékelés egy olyan technika, amellyel a különböző látási objektumokból hasznos szerkezeti információkat nyerhetünk ki, és drámaian csökkenthetjük a feldolgozandó adatmennyiséget. Széles körben alkalmazzák különböző számítógépes látórendszerekben. [16]

Canny megállapította, hogy az érzékelés alkalmazásának követelményei a különböző látórendszerekben viszonylag hasonlóak. Így egy, az ezen követelményeket kielégítő érzékelési megoldás a legkülönbözőbb helyzetekben alkalmazható. Az érzékelés általános kritériumai a következők: [16]

1. Az élek alacsony hibaarányal történő felismerése, ami azt jelenti, hogy a felismerésnek a lehető legtöbb, a képen látható él kell pontosan érzékelnie.
2. A kezelő által észlelt élpontnak pontosan az él középpontjára kell lokalizálnia.
3. Egy adott él a képen csak egyszer legyen megjelölve, és ahol lehetséges, a kép zaj ne hozzon létre hamis éleket.

E követelmények teljesítéséhez Canny a variációs számítást használta - egy olyan technikát, amely megtalálja azt a függvényt, amely optimalizál egy adott függvényt. Az optimális függvényt Canny detektorában négy exponenciális tag összege írja le, de megközelíthető egy Gauss első deriváltjával. [16]

Az eddig kifejlesztett érzékelési módszerek közül a Canny-féle érzékelési algoritmus az egyik legszigorúbban meghatározott módszer, amely jó és megbízható érzékelést biztosít. Az érzékelés három kritériumának való optimális megfelelésének és a megvalósítási folyamat egyszerűségének köszönhetően az egyik legnépszerűbb érzékelési algoritmus lett. [16]

#### Canny algoritmus

A Canny algoritmus első lépése a kép simítása. Canny a Gauss függvény első deriváltját vezette le, amely az optimális érzékelési operátor legjobb közelítése. Kiválasztjuk a megfelelő 1D Gauss funkciót a kép simításához sor és oszlop szerint, azaz végrehajtjuk a konvolúciós műveletet a képmátrixra. Mivel a konvolúciós művelet kielégíti a kommutatív és asszociatív törvényt, a Canny algoritmus általában kétdimenziós Gauss függvényt (a következő ábra alapján) használ kép simítására és a zaj eltávolítására. [16]

ahol  $\sigma$  a Gauss szűrő paraméterét jelöli és a kép simításának kiterjedését szabályozza.

A második lépés a kép gradiens nagyságának és irányának kiszámítása. Az algoritmus a kép gradiens értékének és irányának kiszámításához 22 szomszédos terület korlátozott különbségét alkalmazza. Az elsőfokú parciális derivált közelítése az X és Y irányokban a következő képletekből kapható [16]:

$$E_x[i, j] = \frac{I[i+1, j] - I[i, j] + I[i+1, j+1] - I[i, j+1]}{2} \quad (8)$$

$$E_y[i, j] = \frac{I[i, j+1] - I[i, j] + I[i+1, j+1] - I[i+1, j]}{2} \quad (9)$$

Ezért a kép gradiens számítási operátorának a sablonjai a következők:

$$G_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad (10)$$

$$G_y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \quad (11)$$

A gradiens nagysága és iránya kiszámítható. A kép gradiensének nagysága:

$$||M(i, j)|| = \sqrt{E_x[i, j]^2 + E_y[i, j]^2} \quad (12)$$

A kép gradiensének szöge:

$$\theta(i, j) = \arctan\left(\frac{E_y[i, j]}{E_x[i, j]}\right) \quad (13)$$

Miután megszereztük az  $M[i, j]$  gradiens nagyságú képet, az élék pontos pozicionálásához nem maximális elnyomást kell végezni a képen. Az NMS eljárása segíthet garantálni, hogy minden él egy pixel szélességű legyen. Az algoritmus 33 szomszédos területet használ, amely nyolc irányból áll, hogy interpolációt hajtson végre a gradiens nagyságához a gradiens iránya mentén. Ha az  $M[i, j]$  nagysága nagyobb, mint a gradiens irányában végzett két interpoláció eredménye, akkor az adott pont jelölt perempontként lesz megjelölve, ellenkező esetben nem perempontként lesz megjelölve. Ezért a folyamat révén megszületik a jelölt élkép. [16]

A nem maximális elnyomást követően az algoritmus kettős küszöbértékes módszert alkalmaz az élpontok kiválasztására. Azokat a képpontokat, amelyek gradiensének nagysága a magas küszöbérték felett van, élpontokként, azokat pedig, amelyek gradiensének nagysága az alacsony küszöbérték alatt van, nem élpontokként, a többit pedig jelölt élpontokként jelöli meg. Azokat a jelölt élpontokat, amelyek élpontokhoz kapcsolódnak, élpontokként jelöljük meg. Ez a módszer csökkenti a zaj hatását a végső élkép élére. [16]



9. ábra: Canny érzékelés  
<https://servforu.blogspot.com/>

Az egyik különbség a Canny-algoritmus és az előző szakaszban ismertetett egyszerűbb, Laplace-alapú algoritmus között az, hogy a Canny-algoritmusban az első deriváltakat  $x$  és  $y$  tartományban számítjuk ki, majd négy irányított deriváltként kombináljuk őket. Azokat a pontokat, ahol ezek az irányított deriváltak lokális maximumok, az élekké való összeállításra alkalmasak. [16]

## 2.4 Gépi tanulás

A gépi tanulás a modern informatikai megoldások közé sorolható, melynek célja az emberi gondolkodás és felismerés lemodellezése számítógépek segítségével. A gépi tanulás szorosan kapcsolódik a robotika tudományához is. Ez viszont természetesnek vehető, mivel emberi tulajdonságokkal próbálunk felruházni gépeket, és látható, hogy ők is képesek tévedésekre, mellélövésekre. [17]

### 2.4.1 Áttekintés

A gépi tanulás során a számítógépek felfedezik, hogyan tudnak feladatokat végrehajtani anélkül, hogy kifejezetten erre programoznák őket. Ez azt jelenti, hogy a számítógépek a megadott adatokból tanulnak, hogy bizonyos feladatokat elvégezzenek. A számítógépekre bízott egyszerű feladatok esetében lehetséges olyan algoritmusokat programozni, amelyek megmondják a gépnek, hogyan hajtsa végre az adott probléma megoldásához szükséges összes lépést; a számítógép részéről nincs szükség tanulásra. A bonyolultabb feladatok esetében az ember számára kihívást jelenthet a szükséges algoritmusok kézzel történő létrehozása. A gyakorlatban kiderülhet, hogy hatékonyabb, ha segítünk a gépnek kifejleszteni a saját algoritmusát, mintha az emberi programozóknak kellene minden szükséges lépést megadniuk. [18]

A gépi tanulás tudományága különböző megközelítéseket alkalmaz arra, hogy megtanítsa a számítógépeket olyan feladatok elvégzésére, amelyekhez nem áll rendelkezésre teljesen kielégítő algoritmus. Azokban az esetekben, amikor hatalmas számú lehetséges válasz létezik, az egyik megközelítés az, hogy a helyes válaszok közül néhányat érvényesnek címkéznek. Ez azután képzési adatként használható a számítógép számára a helyes válaszok meghatározására használt algoritmus(ok) fejlesztéséhez. [18]

### 2.4.2 Megközelítések

A gépi tanulási megközelítéseket hagyományosan három nagy kategóriába sorolják, attól függően, hogy milyen jellegű "jel" vagy "visszajelzés" áll a tanuló rendszer rendelkezésére. [18]

**Felügyelt tanulás (Supervised learning):** A számítógépnek bemeneti példákat és a kívánt kimeneteket mutatnak be, amelyeket egy úgynevezett "tanár" ad meg, és a cél egy olyan általános szabály megtanulása, amely a bemeneteket a kimenetekhez rendeli. [18]

**Felügyelet nélküli tanulás (Unsupervised learning):** A tanuló algoritmusnak nem adunk címkéket, így ráhagyva azt, hogy megtalálja a struktúrát a bemenetben. A felügyelet nélküli tanulás lehet öncélú (rejtett minták felfedezése az adatokban) vagy eszköz egy cél eléréséhez (jellemzőtanulás). [18]

**Erősítéssel tanulás (Reinforcement learning):** Egy számítógépes program kölcsönhatásba lép egy dinamikus környezettel, amelyben egy bizonyos célt kell teljesítenie (például járművet vezetni vagy egy ellenféllel játszani). Miközben navigál a problématerben, a program a jutalmakkal analóg visszajelzést kap, amelyet megpróbál maximalizálni. [18]

### 2.4.3 Előre tekintés

A jövőben a gépi látás fő iránya az lehet, hogy olyan gépeket hozzunk létre, amelyek az embereket segítik és önálló gondolkodással rendelkeznek, tehát önálló döntéseket legyenek képesek meghozni. Ez nagy segítség lehet az emberek számára például a háztartásokban vagy más területeken, ezért a témán belül fontos kérdésnek számít az ember és gép együttműködése. [17]

## 2.5 Neurális hálózatok

A neurális hálózat a mesterséges intelligencia megoldásokhoz szorosan kapcsolódó fogalom. A gépi tanulás során elengedhetetlenné vált napjainkban. A mesterséges neurális hálózat az idegrendszer felépítése és működése analógiájára kialakított számítási mechanizmus. Hiszen

a fő cél nem elvi, hanem ténylegesen működő modell létrehozása. Ezt pedig tipikusan valamilyen elektronikai eszközzel és valamilyen tudományos eljárással lehet elérni. Tehát a biológiai elvek alapján megalkottak bizonyos matematikai jellegű modelleket. Ezeket elméleti matematikai módszerekkel pontosították, alkalmazott matematikai módszerekkel számításokra alkalmassá tették, majd számítógépen realizálták. Azonban a matematikai módszerek mellett sokszor heurisztikus megfontolásokra és számítógépes kísérletezésre is szükség van. Egy neurális hálózatot érdemes úgy felfogni, hogy nem kívánja a jelenséget modellezni, arra törvényszerűségeket megállapítani, hanem a jelenséget fekete dobozként kezeli, csak a bemenő (input) és a kimenő (output) adatokat tekinti. A jó neurális hálózat olyan, hogy ugyanarra az inputra hasonló outputot ad, mint a vizsgált jelenség. De a fekete dobozban működő mechanizmust nem tárja fel, maga a neurális hálózat pedig nem hasonlít a jelenségre. A jelenségek fekete dobozként való kezelése a neurális hálózatoknak részben hátrányuk, de részben előnyük is. Hiszen működésükhöz csak adatokra van szükség. [19]

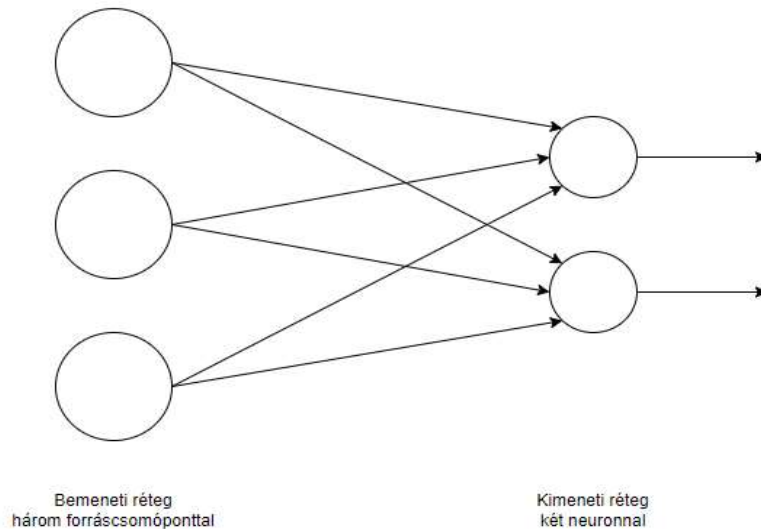
### **2.5.1 Felépítés**

A neurális hálózatok egymással összekapcsolt nagyszámú neuronokból álló egységek struktúrája. A hálózat mindegy egyes neuronja képes bemeneti jelet fogadni, feldolgozni és kimeneti jelet küldeni. Ezek a hálózatok lehetnek egyszerűek és bonyolultak is, amelyek számos rejtett réteggel rendelkezhetnek, ezek a rejtett rétegek nagyon fontosak lehetnek a gépi tanulásban. A hálózati architektúrák két különböző osztályba sorolhatók, az egyik a többrétegű előrecsatolt hálózatok (feedforward networks) és a másik a visszacsatolt hálózatok (recurrent networks). [19]

### **2.5.2 Egyrétegű vagy többrétegű előrecsatolt neurális hálózatok**

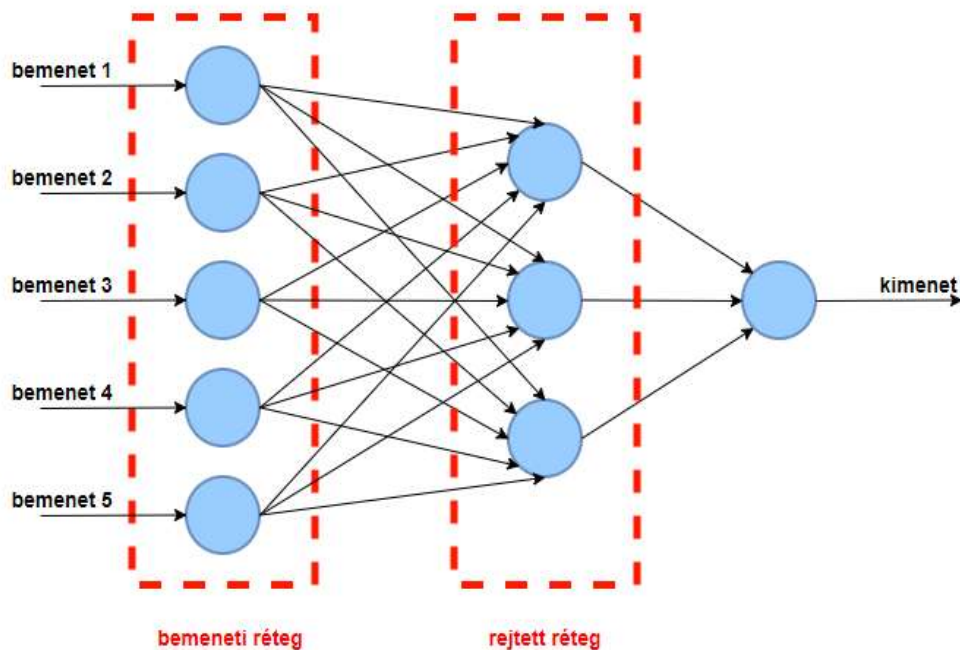
A legrégebbi és legegyszerűbb neurális hálózat az egyrétegű előrecsatolt neurális hálózatok. Ezek a hálózatok egy forráscsomagokból álló bemeneti rétegből és egy neuronokból álló kimeneti rétegből állnak. A hálózatnak csak egy rétege van, mivel a forráscsomópontok rétege nem számít, mert az nem végez számítást és az egy rétegre utal az „egyrétegű” szó is. [19]





10. ábra: Példa egyrétegű előrecsatolt neurális hálózat [19]

A többrétegű előrecsatolt hálózatok egy bemeneti rétegből, amely a forráscsomópontokat tartalmazza, egy vagy több rejtett rétegből és egy neuronokból álló kimeneti rétegből áll. A rejtett rétegek nem láthatók közvetlenül sem a hálózat bemeneti, sem a kimeneti rétegeből. Ezek a rétegek teszik lehetővé a neurális hálózat számára, hogy kivonja a magasabb rendű jellemzőket a bemenetéből. Ennek megfelelően a rejtett rétegekben található neuronokat rejtett neuronoknak nevezzük. Feladatuk, hogy beavatkozzanak a hálózat működésébe valamilyen hasznos módon. [19]



11. ábra: Példa többrétegű előcsatolt neurális hálózat [19]

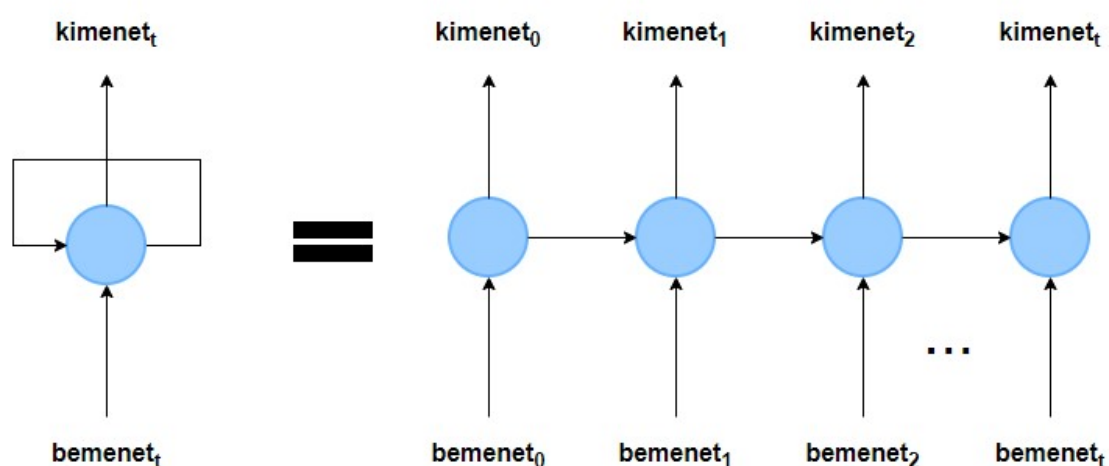


A bemeneti réteg forráscsomópontjai az aktivációs minta megfelelő elemeit szolgáltatják az első rejtett réteg neuronjaira alkalmazott bemeneti jelekhez. Az első rejtett réteg neuronjainak kimeneti jeleit csak a második rejtett réteg neuronjai használják bemeneti jelként. Általában az egyes rejtett rétegekben lévő neuronok kimeneti jeleit csak a szomszédos előre rejtett rétegben lévő neuronok bemeneti jeleként használják. Az azonos rétegben lévő neuronok között nincs kapcsolat. Végül az utolsó rejtett rétegben lévő neuronok kimeneti jeleit csak a kimeneti rétegben lévő neuronok használják bemeneti jelként. A hálózat kimeneti rétegében lévő neuronok kimeneti jeleinek összessége alkotja a hálózat teljes válaszát a hálózat bemeneti rétegében lévő forráscsomópontok által szolgáltatott aktivációs mintára. [19]

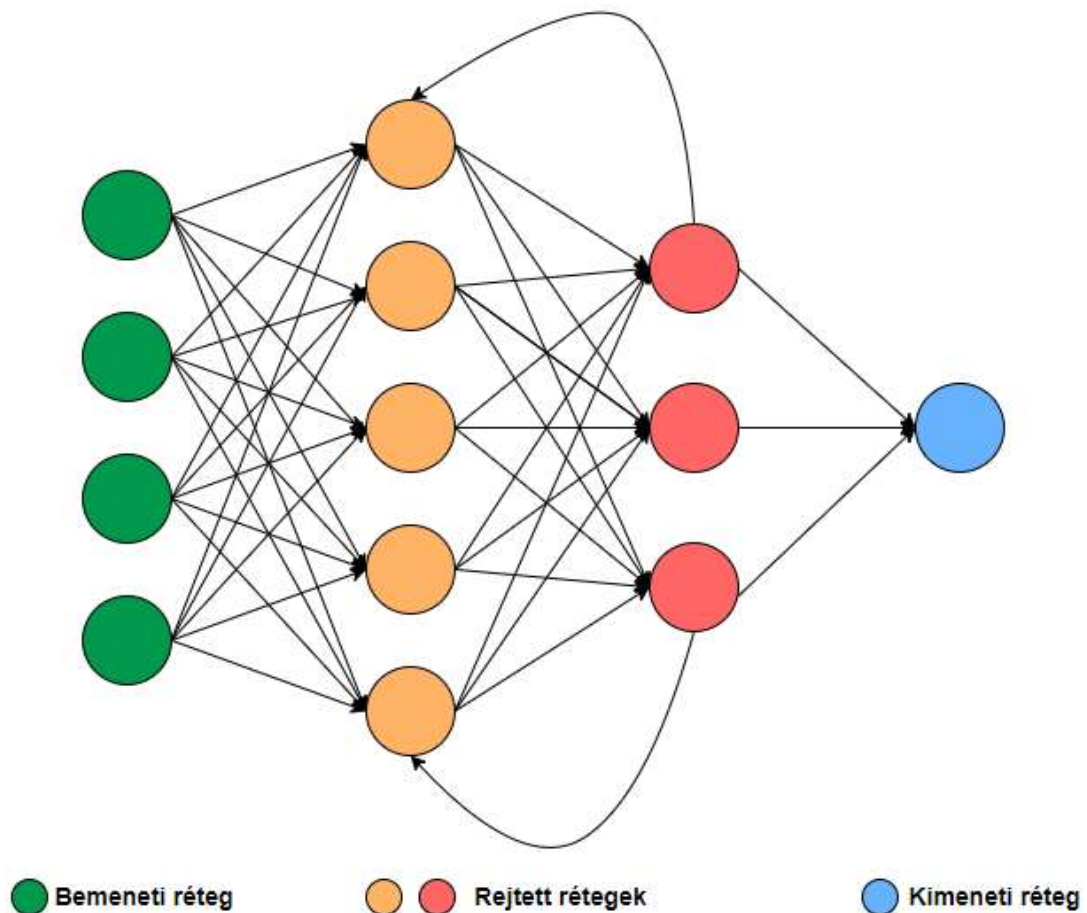
Teljesen összekapcsoltnak nevezzük azokat a többrétegű neurális hálózatot, amelyekben a hálózat mindegy egyes rétegének minden neuronja kapcsolatban áll a következő réteg minden neuronjával. [19]

### 2.5.3 Visszacsatolt neurális hálózatok

A visszacsatolt hálózatok rendelkezhetnek önvisszacsatolási hurkokkal, ezekben a visszacsatolási hurkokban egységnyi késleltetési elemekből álló ágakat használnak és a visszacsatolt hálózatoknak lehetnek rejtett neuronjaik is. A visszacsatolt hálózat abban különbözik az előre csatolt hálózatoktól, hogy legalább egy visszacsatolási hurokkal rendelkezik. Ez nagy rugalmasságot biztosít és tetszőleges dinamikus rendszereket közelíthet meg tetszőleges pontossággal. [19]



12. ábra: Példa visszacsatolt neurális hálózat sematikus rajza [19]



13. ábra: Példa visszacsatolt neurális hálózat

Forrás: <https://dataaspirant.com/how-recurrent-neural-network-rnn-works/>

#### 2.5.4 Tanítás

A neurális hálózatok betanítására úgynevezett tanulási folyamatot (learning process) használnak. A tanulás során a súlyokat a kívánt értékhez igazítják. A tanulás két fő kategóriába sorolható, egyik a felügyelt tanulás a másik pedig a felügyelet nélküli tanulás. A felügyelt tanulás során egy képzési készletet (training set). A képzési készlet tartalmazza a bemeneti példákat és megfelelő célkimeneteket. A súlyokat úgy érdemes beállítani, hogy minimalizáljuk a hálózat kimenete és a helyes kimenet közötti hibát. A képzési készlet felépítéséhez különleges odafigyelésre van szükség. A készletnek reprezentatívnak kell lennie az alapmodellre. Ha egy készlet nem reprezentatív, akkor nem képes nagyon megbízható és általános modellt létrehozni. A felügyelt tanulást alkalmazó hálózatok esetében először a hálózatot kell betanítani. Amikor a hálózat egy sor bemenetre a kívánt kimenetet produkálja, a súlyok rögzítésre kerülnek és ezután a hálózat üzembe helyezhető. Ezzel szemben a felügyelet nélküli tanulás nem használ célkimeneti értékeket. A hálózat

kizárólag a bemeneti adatokban próbálja felfedezni az alapmintázatot. A különböző tanulási folyamatok különböző tanulási folyamatokat igényelnek. [20]

Az egyrétegű neurális hálózatban a képzési folyamat viszonylag egyszerű, mivel a veszteségi folyamat a súlyok működése alapján számítható, ami lehetővé teszi a gradiens egyszerű kiszámítását. A többrétegű hálózatok esetében a problémát az jelenti, hogy a veszteség a korábbi rétegek súlyainak bonyolult összetételű függvénye. [20]

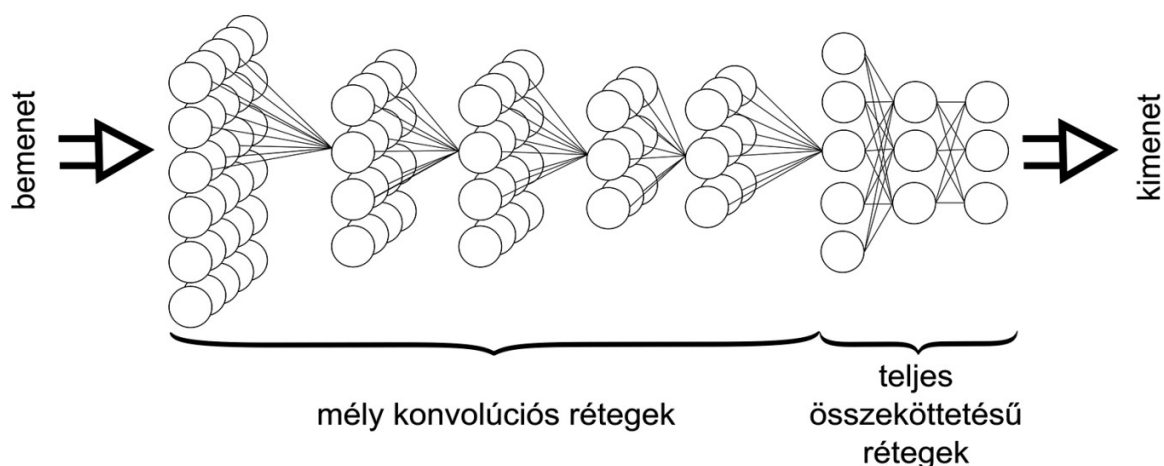
## 2.6 Konvolúciós neurális hálózatok

A klasszikus képfeldolgozásnál alapl művelet a konvolúció. Alkalmazható zajszűrésre, alacsony képi jellemzők kiemelésére, összetett objektumok kiemelésére. A konvolúciós neurális hálózatok előnye a teljesen összekötött hálókhoz képest, hogy jóval kevesebb a szabad paraméter. A konvolúciós rétegben a bementi kép egy megadott nagyságú részét kivesszük és számítást végzünk rajt. Ebben a rétegben egy neuron érzékenységi mezőjének a bemeneti kép azon részét nevezzük, melytől függ a kimeneti értéke. [27]

A képek felismerésére specializált neurális hálózat, amely 3 típusú rétegből tevődik össze [27]:

1. Konvolúciós réteg (convolution layer)
2. Összevonó réteg (pooling layer)
3. Teljesen összekötött réteg (fully connected layer)

Az első és az összevonó rétegek a jellemzők kinyerését végzik, míg a teljesen csatolt réteg a végső kimenetet alkotja meg, mint például classification (osztályozás). [27]



14. ábra: Konvolúciós neurális hálózat sematikus váza  
<https://www.hte.hu/>

A konvolúciós réteg teljes mértékben eltér az eddig ismertetett neurális hálózat rétegektől. Ez a réteg szűrőket tartalmaz, amelyeket a bemeneti képek feldolgozására használ. Ebben a rétegben kerneleket (numerikus tömböket) futtatunk végig a bemeneten. A bemeneti réteg szinten egy numerikus tömb. A kernel végigmegy a képen, és minden lépésnél, a bemenet aktuális része, valamint a kernel között kiszámítja a két tömb elemenkénti szorzatát, majd annak az összege lesz az érték a kimenet adott pozíciójában. Ez a kimenet a feature map(jellemző térkép). A folyamat folytatódik tovább mindaddig (különféle kerneleket alkalmazva), amíg nem érünk el megfelelő számú kimenetet. Új kerneleket alkalmazva a bemeneten más és más jellemzőket tudunk megállapítani. Két fontos jellemzője a konvolúciós rétegnek a kernelek száma, illetve a mérete. A kernel mérete lehet  $3 \times 3$ ,  $5 \times 5$  és  $7 \times 7$  is, ezek közül a leggyakoribb a  $3 \times 3$ , de a gyakorlatban találkozhatunk a többivel is. A kernelek száma tetszőleges, de meghatározza a vizsgálható jellemzők mennyiségét. [27]

A konvolúciós lépéseknél a kernel közepe sosem érinti a kép szélét, emiatt a kimenet kisebb méretű lesz. Erre alkalmazott megoldás a padding, amely közül a leggyakoribb a zero padding. Ezen megoldás segítségével 0 értéket fűzünk a bemenet minden oldalához, ezáltal megnövelve a méretét. Ennek köszönhetően a kernel már tud a kép szélén haladni, tehát a kimeneti kép ugyanakkora lesz, mint a bemeneti kép. A megoldás segítségével azt is elérjük, hogy több réteg alkalmazható, mivel nem kell számolni azzal, hogy a kép kisebb méretű lesz. [27]

Egy másik jellemzője a konvolúciónak a stride(lépés), amely meghatározza, hogy a kernel mekkora lépéseket tegyen meg a bemeneten. Ennek az értéke legtöbbször 1, így az egész kép van vizsgálva, azonban, ha az értéket növeljük, akkor csökkentett mintavételezés történik. Ebben az esetben a kimenet kisebb lesz. [27]

Az összevonó (pooling) réteg segítségével csökkenthető a komplexitás a további rétegek számára. Ebben a rétegben nincs tanítható paraméter, azonban a szűrő mérete, a lépés nagysága, valamint a padding értéke megadható. Ez a réteg lényegében mintavételezi a képet. [27]

Különböző pooling technikákat különböztetünk meg. Az average pooling a gyakori minták kiemelését segíti elő. A max pooling, amely a leggyakoribb pooling technika olyan jellemzők kiemelését segíti, melyek csak kis számban fordulnak elő. A technika lényege, hogy a kép kis régiókra (téglalapokra) van felosztva, és minden régióból a maximum értéket

adja a kimenetre. A max pooling esetén használt leggyakoribb szűrő méret 2x2. Ebben az esetben az első 2x2 méretű régióból kimenetre adja a legnagyobb értéket, ezután tovább mozdul a szűrő. Átfedés nincsen a vizsgálatok között, ezáltal a kimenet kisebb lesz a bemenetnél. [27]

Pooling esetén még érdemes megjegyezni, hogy a folyamat által a pozíciós információk nem maradnak meg. Ennek következtében csak akkor alkalmazandó, amikor az információnak csak a jelenléte a fontos, a pozíciója nem. [27]

A pooling által generált kimenet általában le van lapítva (flattened). Ez az jelenti, hogy egy dimenziós szám tömbbe van átalakítva, és ezt egy vagy több teljesen csatolt (dense) réteghez kapcsolják. Ebben a rétegben vagy rétegekben dől el a végső kimenet. [27]

A legvégső teljesen csatolt rétegben általában annyi kimenet van, amennyi osztály van osztályozás esetén, és minden kimenet megadja, hogy mekkora esély van arra, hogy a vizsgált dolog az adott osztályba tartozik. [27]

## **2.7 Hasonló munkák**

Ebben a fejezetben olyan már meglévő módszereket szeretnék bemutatni, amelyek segítségével megvalósítható az általam kiválasztott projekt.

### **2.7.1 Túlexponálás korrigálása és detektálása**

Guo et. al [3] egy olyan módszert dolgoztak ki, amely a meglévő fényképen a szín és a világosság külön-külön történő helyreállításával korrigálja a túlexponálást. Először a jól exponált terület dinamika tartományát egy kissé összenyomják azért, hogy helyet teremtsenek a túlexponált terület visszanyert világosságának. Ezután a fényerő visszanyerése a túlvilágítás valószínűsége alapján történik. Az egyes képpontok színét a szomszédságterjedésen keresztül és az eredeti szín megbízhatósága alapján korrigálják. A módszer teljesen automatikus és csak egyetlen bemeneti fotót igényel. [3]

### **2.7.2 Elmosódások korrigálása és detektálása**

## **3 Tervezés**

### **3.1 Fejlesztő környezet, programnyelv, programcsomagok**

A projekt elkészítése során fejlesztő környezetként a Visual Studio Code alkalmazást használtam. Ennek az alkalmazásnak a használata rendkívül előnyös, mivel többféle

programozási nyelvet támogat, emellett az általa elfoglalt hely is nagyon alacsony, valamint könnyen, de nagy mértékben személyre szabható. További előnyként még megemlíthető, hogy platformfüggetlen. Ez a használat során nagyon nagy előnyt jelenthet a felhasználók számára.

A projekt Phyton nyelven íródott, elsősorban azért esett a választás erre a nyelvre, Phyton a legkedveltebb nyelv a gépi tanulós projektek készítőinek körében. Ezenkívül számomra az is előnyös volt, hogy a nyelv könnyen tanulható, szintaktikája és adatkezelése meglehetősen egyszerű.

A projekt elkészítése során többféle programcsomag használatára volt szükség. Először is szükség volt a tensorflow programcsomag használatára, mivel ez elengedhetetlen a gépi tanulás és a neurális hálózatok végett. Ahhoz, hogy adatokat tudjunk kezelni beolvasni és feldolgozni szükség volt a numpy, illetve pandas programcsomagok használatára. Ezeken kívül matplotlib programcsomag van még, ami azért van használva, hogy kiírassam a tanítás eredményét egy grafikon formában. A képek beolvasását, kezelését, valamint különféle képi transzformációk elvégzését OpenCV csomag segítette.

### 3.2 Funkcióterv

## 4 Irodalomjegyzék

- [1] A. Golchubian, O. Marques és M. Nojournian, „Photo quality classification using deep learning,” *Multimedia Tools and Applications*, 2021.
- [2] S. Zhang, X. Shen, Z. Lin, R. Měch, J. P. Costeira és J. M. F. Moura, „Learning to Understand Image Blur,” 2018.
- [3] D. Guo, Y. Cheng, S. Zhou és T. Sim, „Correcting over-exposure in photographs,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.
- [4] L. Hou, H. Ji és Z. Shen, „Recovering Over-/Underexposed Regions in Photographs,” *SIAM Journal on Imaging Sciences*, pp. 2213-2235, 2013.
- [5] B. Su, S. Lu és C. L. Tan, „Blurred image region detection and classification,” *19th ACM international conference on Multimedia*, pp. 1397-1400, 2011.
- [6] S. Dai és Y. Wu, „Motion from Blur,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [7] E. Kalalembang, K. Usman és I. P. Gunawan, „DCT-based local motion blur detection,” *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering*, 2009.
- [8] X. Wang, S. Zhang, X. Liang, H. Zhou, J. Zheng és M. Sun, „Accurate and Fast Blur Detection Using a Pyramid M-Shaped Deep Neural Network,” *IEEE Access*, 2019.
- [9] D.-K. Lee, B.-K. Kim és R.-H. Park, „Purple Fringing Correction Using Colorization



- in Yxy Color,” *IEEE International Conference on Consumer Electronics*, 2011.
- [10] J. Huang, C. Cui, C. Zhang, Z. Shen, J. Yu és Y. Yin, „Learning Multi-Scale Attentive Features for Series Photo Selection,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020., pp. 2742-2746.
- [11] D. Kuzovkin, T. Pouli, R. Cozot, O. Le Meur, J. Kervec és K. Bouatouch, „Context-aware clustering and assessment of photo collections,” in *Proceedings of the symposium on Computational Aesthetics*, ACM, 2017., p. 6.
- [12] D. Kuzovkin, T. Pouli, R. Cozot és O. Le Meur, „Image Selection in Photo Albums”.
- [13] H. Chang, F. Yu, J. Wang, D. Ashley és A. Finkelstein, „Automatic triage for a photo series,” in *ACM Transactions on Graphics*, ACM, 2016, pp. 1-10.
- [14] J. Bernacki, „Automatic exposure algorithms for digital photography,” *Multimedia Tools and Applications*, pp. 12751-12776, 2020..
- [15] R. Bansal, G. Raj és T. Choudhury, „Blur image detection using Laplacian operator and Open-CV,” *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, 2016..
- [16] G. Bradski és A. Kaehler, *Learning OpenCV*, United States of America: O'Reilly Media, 2008.
- [17] P. Gyarmati, „Gondolatok a mesterséges intelligencia, gépi tanulás kapcsán,” *Mesterséges intelligencia - interdiszciplináris folyóirat*, pp. 31-39, 2019.
- [18] E. Alpaydin, „Introduction to Machine Learning, Fourth Edition,” 2020..
- [19] Z. Zhang, „Artificial Neural Network,” *Multivariate Time Series Analysis in Climate and Environmental Research*, pp. 1-35, 2017..
- [20] J. Zou, Y. Han és S.-S. So, „Overview of Artificial Neural Networks,” in *Artificial Neural Networks*, 2008., pp. 14-22.
- [21] D. Hadházi, „Mély konvolúciós neurális hálózatok,” 2018..
- [22] Y. Deng, C. Change Loy és X. Tang, „Image aesthetic assessment: An experimental survey,” in *IEEE Signal Processing Magazine*, IEEE, 2017., pp. 80-106.
- [23] C. Cui, H. Liu, T. Lian, L. Nie, L. Zhu és Y. Yin, „Distribution-Oriented Aesthetics Assessment With Semantic-Aware Hybrid Network,” in *IEEE Transactions on Multimedia*, IEEE, 2018., pp. 1209-1220..
- [24] D. Kuzovkin, T. Pouli, R. Cozot, O. Le Meur, J. Kervec és K. Bouatouch, „Contextaware clustering and assessment of photo collections,” in *Proceedings of the Symposium on Computational*, ACM, 2017., p. 6.
- [25] D. Kuzovkin, T. Pouli, R. Cozot, O. Le Meur, J. Kervec és K. Bouatouch, „Image selection in photo albums”.
- [26] H. Chang, F. Yu, J. Wang, D. Ashley és A. Finkelstein, „Automatic triage for a photo,” *ACM Transactions on Graphics*, %1. kötet35, %1. szám4, p. 148., 2016..
- [27] Albawi, S., Mohammed, T. A., Al-Zawi, S.: Understanding of a convolutional neural network

## **5 Mellékletek**

[A dolgozat mellékletei, ha vannak]