**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Programozáselmélet és Szoftvertechnológia Tanszék**

Arcfelismerő   
ResNET használatával

Szerző: Témavezető:

Skultéty Áron Bencsik Gergely

Programtervező informatikus BSc. egyetemi docens

**Szombathely, 2022**

Ide kerül a hivatalos témabejelentő lap.

Tartalomjegyzék

[1. Bevezetés 1](#_Toc87521353)

[2. Irodalmi áttekintés 1](#_Toc87521353)

[3. Felhasználói dokumentáció 2](#_Toc87521354)

[4. Fejlesztői dokumentáció 3](#_Toc87521355)

[5. Tesztelés 1](#_Toc87521353)

[5. Összefoglalás és további fejlesztési lehetőségek 4](#_Toc87521356)

[6. Eredmények összehasonlítása 5](#_Toc87521357)

[7. Összefoglalás és továbbfejlesztési lehetőségek 6](#_Toc87521358)

[8. Irodalomjegyzék 1](#_Toc87521353)

[8. Melléklet 1](#_Toc87521353)

# Bevezetés

Az arcfelismerés egy mesterséges intelligencián alapuló technológia, amely lehetővé teszi az emberek automatikus felismerését és azonosítását képek vagy videók alapján. A dolgozat célja egy Python alapú arcfelismerő rendszer fejlesztése, amely az OpenCV könyvtárat az arcdetektáláshoz és a PyTorch-ot a mélytanulás alapú felismeréshez alkalmazza. Vizsgáljuk a klasszikus és modern mélytanulási megoldásokat, összehasonlítva azok hatékonyságát és megbízhatóságát.

A képen poszter, képernyőkép, Emberi arc, szöveg látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. ábra: Nyitókép

**1.1 Az arcfelismerés jelentősége, alkalmazási területei:**

*Az arcfelismerés egy mesterséges intelligencián alapuló technológia, amely képes automatikusan felismerni és azonosítani emberek arcát képeken vagy videókon. Főbb jellemzői:*

1. ***Automatizált működés****: Az emberi beavatkozás minimalizálása érdekében az arcfelismerés képes valós idejű vagy előzetesen rögzített képek feldolgozására.*
2. ***Pontosság és megbízhatóság****: A modern algoritmusok, különösen a mélytanulási alapúak, rendkívül pontosan képesek megkülönböztetni az egyéneket, még hasonló arcok esetén is.*
3. ***Skálázhatóság****: Az arcfelismerő rendszerek alkalmazhatók kis léptékben (pl. személyes eszközökön) és nagy léptékben is (pl. tömeges megfigyelés).*
4. ***Robusztusság különböző körülmények között****: Az újabb rendszerek képesek alkalmazkodni eltérő fényviszonyokhoz, arckifejezésekhez, szögekhez és részleges eltakarásokhoz.*
5. ***Nem invazív technológia****: Nem szükséges fizikai érintkezés, így kényelmes és gyors az alkalmazása.*

**1.2 Az arcfelismerés alkalmazási területei**

*Az arcfelismerés széles körben alkalmazható különböző iparágakban és mindennapi élethelyzetekben. Főbb alkalmazási területei közé tartoznak:*

1. **Biztonság és megfigyelés**
   * Beléptető rendszerek: Hozzáférés korlátozása érzékeny területeken (pl. irodák, laboratóriumok).
   * Rendőrségi és katonai célok: Keresett személyek azonosítása tömegekben vagy videofelvételeken.
2. **Mobiltechnológia**
   * Eszközök feloldása: Arcfelismerés alapú képernyőzár (pl. Face ID).
   * Biztonságos tranzakciók: Banki műveletek jóváhagyása arcfelismeréssel.
3. **Egészségügy**
   * Pszichológiai diagnózis: Arckifejezések elemzése stressz, depresszió vagy más pszichés problémák észlelésére.
   * Ritka betegségek azonosítása: Ritka genetikai szindrómák felismerése arcvonások alapján.

**1.3 A szakdolgozat célja:**

A szakdolgozat célja egy arcfelismerő rendszer megvalósítása Python programozási nyelven, amely az OpenCV könyvtárat használja az arcok detektálására, és a TensorFlow vagy PyTorch könyvtárakat az arcfelismerés finomítására és deeplearning algoritmusok integrálására. A dolgozat keretein belül vizsgálom az OpenCV hatékonyságát, továbbá implementálok fejlettebb megoldásokat, amelyek gépi tanulást használnak a feladat további optimalizálására.

# Felhasználói dokumentáció

***2.1 Klasszikus arcfelismerő módszerek:***

***2.1.1. Haar Cascade (Viola–Jones algoritmus)***

* ***Leírás****: A Haar Cascade az egyik legismertebb klasszikus arcdetektáló algoritmus, amelyet Paul Viola és Michael Jones fejlesztett ki. A módszer kiemelkedően hatékony, és valós időben képes arcokat detektálni.*
* ***Működés****:*
  + *Alapvető geometriai minták (Haar-jellemzők) keresése az arc különböző területein.*
  + *Többlépcsős döntési fa (cascade classifier) használata a számítási igény csökkentésére.*

***2.1.2. Eigenfaces (Főkomponens-analízis, PCA)***

* ***Leírás****: Az eigenfaces módszer a főkomponens-analízisen (Principal Component Analysis, PCA) alapul, amely az arcokat vektoros formában reprezentálja.*
* ***Működés****:*
  + *Az arcokat egy nagy dimenziós térben vektorokként ábrázolja.*
  + *A főkomponensek meghatározásával csökkenti a dimenziók számát.*
  + *Az azonosítást az arcok közötti euklideszi távolság alapján végzi.*

**2.2 Neurális Hálók – Irodalmi áttekintés**

*A mesterséges intelligencia fejlődésével a neurális hálók egyre meghatározóbb szerepet kaptak az adatfeldolgozásban, különösen a képfeldolgozás és az arcfelismerés területén. A neurális hálók inspirációja az emberi agy idegsejthálózata, ahol az információk rétegek között terjednek, súlyok és aktivációs függvények segítségével módosulva.*

***2.2.1. A neurális hálók alapjai***

*A neurális háló egy matematikai modell, amely többrétegű, összekapcsolt neuronokból épül fel. A leggyakrabban alkalmazott architektúra a* ***feedforward neurális hálózat****, amely bemeneti, rejtett és kimeneti rétegekből áll. A hálózat tanulása a* ***backpropagation algoritmuson*** *alapszik, amely gradiensalapú optimalizációval frissíti a súlyokat a hibák csökkentése érdekében.*

***2.2.2. Konvolúciós neurális hálók (CNN-ek)***

*A képfeldolgozási feladatokhoz speciálisan tervezett* ***konvolúciós neurális hálók (CNN-ek)*** *kiemelkedő hatékonyságot mutatnak. Ezek a modellek képesek felismerni és megtanulni különböző vizuális mintázatokat, például széleket, textúrákat és komplex formákat. A CNN-ek legfontosabb komponensei:*

* ***Konvolúciós rétegek****: Szűrők (filterek) segítségével emelik ki a képi információkat.*
* ***Pooling rétegek****: Az adatredukciót szolgálják, hogy csökkentsék a számítási igényt és a túlilleszkedést.*
* ***Teljesen összekapcsolt rétegek****: A végső osztályozásért felelnek.*

***2.2.3. ResNet és a mélyhálózatok problémái***

*A hálózatok mélyítésével új kihívások jelentek meg, mint például a* ***gradiens eltűnés****e és a* ***túlilleszkedés****. Erre a problémára kínált megoldást a* ***ResNet*** *(Residual Networks), amely maradék (residual) kapcsolatokat alkalmaz a rétegek között, lehetővé téve az információ közvetlen továbbítását mélyebb rétegekbe. A ResNet modellek nagy sikerrel alkalmazhatók arcfelismerési feladatokban is.*

***2.2.4. Neurális hálók az arcfelismerésben***

*A modern arcfelismerő rendszerek CNN-alapú modellekre épülnek, amelyek az arcokról* ***embedding vektorokat*** *állítanak elő. Az ilyen megoldások, mint a* ***FaceNet*** *vagy a* ***DeepFace****, a hagyományos osztályozási módszerek helyett a hasonlósági mérésekre helyezik a hangsúlyt, például* ***koszinusz-távolság*** *vagy* ***euklideszi távolság*** *alapján.*

***2.3 Mélytanulásos módszerek PyTorch-al***

*A mélytanulási megközelítések az arcfelismerés területén jelentős előrelépést hoztak, különösen olyan keretrendszerek, mint a TensorFlow és a PyTorch használatával. Az alábbiakban a legfontosabb mélytanulási módszereket ismertetem:*

1. ***Konvolúciós Neurális Hálók (CNN-ek)****: Az arcfelismerési modellek alapját képezik, mivel hatékonyan képesek az arcok vizuális jellemzőinek automatikus kinyerésére.*
2. ***Szemantikus Beágyazások (Face Embeddings)****: Az arcfelismerés egyik modern megközelítése, amely az arcokat alacsony dimenziós vektorokba ágyazza be a hatékony összehasonlítás érdekében.*
3. ***Előre betanított modellek alkalmazása****: A ResNet mellett más előtanított modellek is felhasználhatók, mint például az EfficientNet vagy az MTCNN az arcok detektálására és jellemzőik kinyerésére.*

*A dolgozatomban a ResNet-alapú megközelítést alkalmazom az arcfelismerési feladatokhoz, optimalizálva a felismerési pontosságot és a hálózat teljesítményét.*

# Fejlesztői dokumentáció

**3.1** **A projekt megvalósítási lépései**

A projekt főbb lépései a következők:

* **Adatgyűjtés és előfeldolgozás**: A saját készítésű képek és a nyilvánosan elérhető adatbázisok kombinálása.
* **Mélytanulási modell kialakítása és betanítása**: ResNet architektúra használata az arcfelismeréshez.
* **Validáció és finomhangolás**: A hálózat teljesítményének értékelése és optimalizálása.

Ebben a szakaszban részletesen bemutatom az adatgyűjtési stratégiát és a ResNet-alapú mélytanulási modellt.

**3.2 Az alkalmazott technológiák ismertetése**

**3.2.1. Adatgyűjtés és tárolás**

A rendszer tanításához és teszteléséhez különböző forrásokból származó adatok kerültek felhasználásra. Az adatokat a dataset\_create.py és dataset\_anc&pos.py szkriptek segítségével gyűjtöttem és dolgoztam fel. A képek a következő struktúrában kerülnek tárolásra:

* **Anchor képek**: Egy aktuális anchor kép mindig frissül, amelyet a modell klasszifikációhoz lehet használni.
* **Employees képek:** A kamerával készített Employees mappa tartalmazza az adott személyek (apa, áron, hanna) arcait különböző helyzetekben és személyenként külön mappában.
* **Negatív képek:** A Labeled Faces in the Wild (LFW) adatbázisból származó **7480** kép biztosítja a negatív mintákat az arcfelismerő rendszer számára.

**Az adatok előfeldolgozása során:**

* A képek méretezése és normalizálása történik, hogy megfelelő formátumban kerüljenek a mélytanulási modellbe.
* Az arcok automatikus felismerésére OpenCV vagy MTCNN használható.

**2. ResNet-alapú mélytanulási modell**

A ResNet (Residual Network) egy mély konvolúciós neurális hálózat, amely különösen hatékony képfeldolgozási és mintázatfelismerési feladatokban. A ResNet fő előnye az úgynevezett maradék (residual) kapcsolatok alkalmazása, amelyek segítenek a mély hálózatok hatékonyabb tanításában és elkerülik a gradiens eltűnésének problémáját.

**A jelenlegi implementációm egy testreszabott ResNet-50 variáns, amely az alábbi komponensekből áll:**

* Konvolúciós blokkok (convBlock): Ezek az alapvető építőelemek, amelyek konvolúciós rétegeken keresztül kiemelik a fontos vizuális mintázatokat.
* Maradék blokkok (residualBlock): Az eredeti bemenet és a feldolgozott rétegek összeadásával biztosítják a hatékony tanulást.
* Kimeneti osztályozó réteg: A háló végén egy teljesen összekapcsolt réteg biztosítja az arcok osztályozását vagy jellemzőik kinyerését (embeddingek formájában).

***A modell a következő szerkezettel épül fel (részlet a resnet.py fájlból​:***

***A képen szöveg, képernyőkép, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.***

***A hálózatot egy 224×224 méretű képen teszteltem, és az alábbi kimenetet generálta:***

***A képen szöveg, Betűtípus, képernyőkép, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.***

* 4.Összefoglalás és további fejlesztési lehetőségek
* **4.1 Arcfelismerő és Adatbázis Fejlesztési Lehetőségei**
* **4.1.1. Bevezetés**

Ebben a dokumentációban bemutatom a MySQL adatbázist, amelyet az arcfelismerő alkalmazásomhoz készítettem. Az adatbázis célja, hogy a későbbiekben lehetőséget biztosítson a belépési és kilépési események nyilvántartására. Ha az arcfelismerő felismer valakit, és az a személy szerepel az adatbázisban, akkor a rendszer automatikusan rögzíti a belépési és kilépési időpontokat.

A rendszer **fordított logikát** alkalmaz: **alapértelmezett állapotként minden személy inaktív (inactive)**, és csak akkor válik aktívvá (active), ha az arcfelismerő felismeri és rögzíti a belépést.

* **4.2. Az adatbázis szerkezete**

Az adatbázis két fő táblából áll:

* **users** – A nyilvántartott személyeket tárolja.
* **worklogs** – A belépési és kilépési eseményeket tartalmazza.
* **4.2.1. Felhasználók táblája (users)**
* **id** – Egyedi azonosító minden felhasználónak.
* **full\_name** – A személy teljes neve.
* **employee\_id** – Egyedi törzsszám, amelyet az arcfelismerő használhat a személyek azonosítására.
* **id\_card\_number** – Személyi igazolvány száma.
* **email** – A felhasználó email címe.
* **phone\_number** – Telefonszám.
* **role** – A személy szerepköre (admin, employee, guest).
* folder\_link – A személyhez kapcsolódó egyedi mappa elérési útvonala (pl. data/train/apa).
* **4.2.2. Belépések/Kilépések táblája (worklogs)**
* **id** – Egyedi azonosító minden eseményhez.
* **user\_id** – Kapcsolat a users táblához, így meg tudom határozni, hogy ki lépett be.
* **entry\_time** – Automatikusan felveszi az aktuális időpontot, amikor egy személy belép.
* **exit\_time** – Kilépési időpont, amely akkor töltődik ki, amikor a személy elhagyja az adott területet.
* **status** – A személy állapota, amely lehet:
  + inactive: Alapértelmezett érték, ha a személy nincs bent.
  + active: Ha a személy belépett.

A rendszer automatikusan kezeli a státuszváltásokat az arcbeolvasások alapján.

* **4.3. Adatok kezelése**
* **4.3.1. Új felhasználók hozzáadása**

Az alábbi SQL-paranccsal új személyeket adok hozzá az adatbázishoz.

* **4.3.2. Belépési esemény rögzítése**

Ha az arcfelismerő felismeri a személyt, és ő korábban kilépett (inactive állapotban van), akkor a legutóbbi inactive bejegyzés **active státuszra vált**, és frissíti az entry\_time mezőt.

A képen szöveg, Betűtípus, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Ha nincs korábbi inactive bejegyzése, akkor **új aktív bejegyzés** jön létre:

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**4.3.3. Kilépési esemény rögzítése**

Ha az arcfelismerő újra felismeri a személyt, és az illető már aktív (active állapotban van), akkor **új inactive bejegyzés jön létre az exit\_time rögzítésével**.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**4.4. Fejlesztési lehetőségek**

**4.4.1. Az arcfelismerő összekötése az adatbázissal**

Az **OpenCV** és **ResNet** modell segítségével azonosítani tudom a személyeket, és az adatbázisból lekérdezhetem, hogy szerepelnek-e benne.

Ha a személy ismert, **automatikusan** rögzíthetem a belépését.

**4.4.2. Automatikus belépés és kilépés követése**

Ha egy személyt **először** ismer fel a rendszer, akkor frissítjük az utolsó **inactive** státuszú bejegyzést **active**-ra, és mentjük az **entry\_time**-ot.

Ha a személy már **bent van** (active státusszal rendelkezik), akkor **új inactive** bejegyzést hozunk létre az **exit\_time** kitöltésével.

Ez a logika biztosítja, hogy minden egyes arcbeolvasáskor pontosan kövessük a dolgozók mozgását az adatbázisban.

# Irodalomjegyzék

# Melléklet