***Arcfelismerés éles***

1. **Bevezetés**
   1. Az arcfelismerés jelentősége, alkalmazási területei.
   2. A szakdolgozat célja.
2. **Irodalmi áttekintés**
   1. Klasszikus arcfelismerő módszerek (pl. OpenCV Haar Cascade).
   2. Mélytanulási megközelítések TensorFlow és PyTorch segítségével.
3. **Módszertan**
   1. A projekt megvalósítási lépései (adatgyűjtés, előfeldolgozás, modellek építése).
   2. Az alkalmazott technológiák ismertetése.
4. **Megvalósítás**
   1. OpenCV alapú arcdetektálás.
   2. TensorFlow vagy PyTorch modellépítés és integrálás.
   3. Modell validálása és finomítása.
5. **Eredmények és összehasonlítás**
   1. Klasszikus és mélytanulási technikák eredményeinek bemutatása.
   2. Teljesítménymutatók, pl. pontosság, futási idő.

**1/a 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.*

**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/b A szakdolgozat célja:**

*A szakdolgozat célja egy arcfelismerő rendszer megvalósítása Python programozási nyelven, amely az OpenCV könyvtárat*

*használja az arcok detektálására, és a PyTorch könyvtárat 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.*

*2/a* ***Klasszikus arcfelismerő módszerek:***

***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. 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/b* ***Mélytanulásos módszerek Pytorch-al:***

*A mélytanulási megközelítések az arcfelismerés területén jelentős előrelépést hoztak, különösen olyan keretrendszerek, mint a TensorFlow és a PyTorch használatával. Ezek a technikák képesek pontosabb és robusztusabb arcfelismerési rendszereket létrehozni. Az alábbiakban néhány elterjedt mélytanulási módszert mutatok be:*

***1. Konvolúciós Neurális Hálók (CNN-ek)***

* ***Leírás****: A CNN-ek az arcfelismerési modellek alapját képezik, mivel hatékonyan képesek az arcok vizuális jellemzőinek automatikus kinyerésére. A szakdolgozatomban ezt a módszer fogom alkalmazni.*

***2. Szemantikus Beágyazások (Face Embeddings)***

* ***Leírás****: A mélytanulás-alapú arcfelismerés legmodernebb megközelítése az, hogy az arcokat nem egyszerűen osztályozzuk, hanem egy alacsony dimenziós vektorba ("embedding") ágyazzuk be, amely az arc jellemzőit reprezentálja.*

***3. Rekurzív Neurális Hálók (RNN-ek) időbeli adatokhoz***

* ***Leírás****: Ha az arcfelismerés videó alapú, az RNN-ek vagy LSTM-ek használhatók az időbeli adatok (pl. arckifejezések, mozgások) feldolgozására.*

***4. Előre betanított modellek használata***

* ***Leírás:*** *Sok esetben előre betanított modellek használhatók, mint például:*
  + *MTCNN (Multi-task Cascaded Convolutional Networks) az arcok detektálására.*
  + *Dlib arcfelismerő könyvtárának integrációja.*
  + *TensorFlow-ban és PyTorch-ban: előre betanított ResNet vagy EfficientNet modellek finomhangolása arcfelismerésre.*

*3/a* ***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/b*  ***Az alkalmazott technológiák ismertetése***

***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, valamint van egy külön "anchor" mappa (****337*** *elemmel) amelyben a pozitív képekhez hasonló arcképeim vannak.*
* ***Pozitív képek:*** *A kamerával készített* ***389*** *elemű pozitív mappa tartalmazza az adott személy arcait (jelen esetben az enyémet) különböző helyzetekben.*
* ***Apa mappa:*** *Egy alternatív mappa* ***439*** *képpel, amely szükség esetén további arcokat tartalmazhat az edzéshez.*
* ***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.***