

TDK-dolgozat

Bognár Lili

2025

Pécsi Tudományegyetem
Pollack Mihály Műszaki és Informatikai Kar

Műszaki informatika Tanszék

Deepfake detection

Készítette: Bognár Lili

Évfolyam: 2021

Belső konzulens: Dr. Storcz Tamás, adjunktus

Pécs, 2024

Tartalomjegyzék

1. Bevezetés és célkitűzések	4
1.1. Az arcmanipuláció	4
1.2. Deepfake videók generálásának módszerei	5
1.3. Megtörtént esetek	5
1.4. Célkitűzések és módszerek	6
2. Irodalmi áttekintés	7
2.1. Összegzés és kapcsolódás a saját munkához	9
3. Módszerek	10
3.1. Tanítási környezet	10
3.2. Dataset	10
3.3. Előfeldolgozás	11
3.4. Adataugmentáció	12
3.5. Early Stopping	12
3.6. Arcdetektálás	13
3.7. Modellek ismertetése	14
3.7.1. A modellek architektúrája	14
3.7.2. Veszteségfüggvény: BCEWithLogitsLoss	15
3.7.3. Optimalizációs algoritmus: AdamW	16
3.7.4. Konvolúciós mechanizmusainak elemzése	16
4. Eredmények	20
4.1. Módszertan és metrikák	20
4.2. Eredmények	21
4.3. Eredmények összehasonlítása	22

4.4. Valós idejű detekció tesztelése	23
5. Értékelés és következtetések	24
Összefoglalás	26
Köszönetnyilvánítás	28
Irodalomjegyzék	28

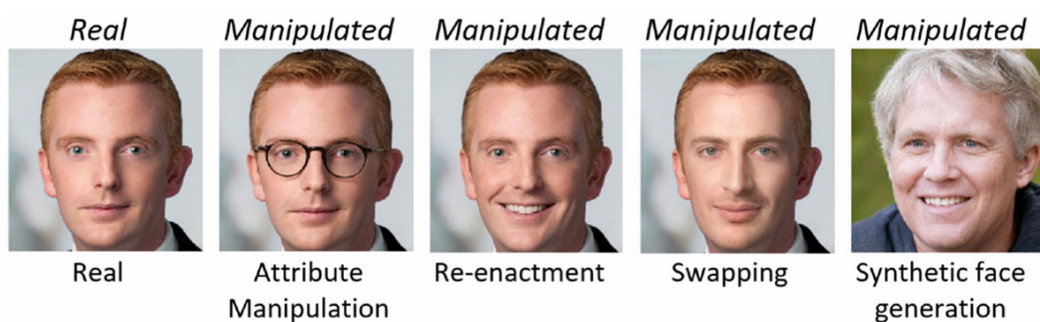
1. fejezet

Bevezetés és célkitűzések

1.1. Az arcmanipuláció

Az arcmanipuláció a deepfake technológia egyik leggyakrabban használt és leginkább hírvást jelentő formája. A deepfake videók és képek esetében az arcokat gyakran módosítják, hogy egy másik személy vonásait, kifejezéseit, vagy éppen mozgását tükrözzék. Az arc az egyik leggyakrabban alkalmazott azonosítási eszköz, ezért elengedhetetlen, hogy meg tudjuk állapítani, egy arc hamisított-e vagy sem.

A különféle arcmanipulációk felismerése és csoportosítása lehetővé teszi, hogy pontosabban meghatározzuk, milyen szintű beavatkozás történt. A az 1.1. ábrán láthatóak ezen manipuláció különböző típusai.



1.1. ábra. Az arcmanipuláció típusai
Forrás:[1]

Attribute Manipulation Csak az arc bizonyos jellemzőit változtatják meg, például a hajszínt vagy a bőrszínt.

Teljes arckifejezéscsere A mimika, az érzelmek manipulálására összpontosít, anélkül, hogy a személy identitását megváltoztatnák.

Identitáscsere Egy személy arcát egy másik személy arcával cserélik le, ami gyakran látható deepfake videókban.

Face Synthesis Teljesen új arcot hoznak létre, ami nem feltétlenül létezik a valóságban.

1.2. Deepfake videók generálásának módszerei

A „deepfake” kifejezést 2017-ben egy Reddit-felhasználó alkotta meg, aki arc-cserélő technológiát használt hírességekről készült felnőtt tartalmak közzétételére.[2] Azóta a technológia fejlődött, és több területen is alkalmazásra került, mint például a szórakoztatóiparban, a politikában és a közösségi médiában.

A Generative Adversarial Network (GAN) egy mesterséges intelligencia modell, amely két versengő hálózathoz áll. Az interneten található deepfake képek és videók többsége GAN-alapú technológiák segítségével készül. [3]Működése a generatív és diszkriminatív hálózatok közötti versengésen alapul: a generatív hálózat célja, hogy minél valósághűbb adatokat generáljon, míg a diszkriminatív hálózat feladata, hogy megkülönböztesse a valódi adatokat a generáltaktól. Ezek a modellek úgy működnek, hogy egy bemene-ti mintagyűjteményt elemezve megtanulják a mintákat létrehozó valószínűségi eloszlást, majd ebből az eloszlásból új példákat generálnak.

Léteznek ezenkívül más deepfake-generálási módszerek is, mint például a diffúziós modellek. A diffúziós modellek működési elve a zaj hozzáadásának és eltávolításának folyamatán alapul.

1.3. Megtörtént esetek

A deepfake technológia egyre nagyobb fenyegetést jelent a közvélemény tájékoztatására, hírességek magánéletére és a pénzügyi biztonságra. 2022-ben például egy Volodimir Ze-

lenszkij ukrán elnökről készült manipulált videó zavarta meg az embereket, mivel azt a látszatot keltette, hogy az elnök megadja magát az orosz csapatoknak.

Továbbá 2024-ben Taylor Swift-ről hamisított szexuális tartalmak jelentek meg, amelyek széles körű felháborodást váltottak ki. Ugyanebben az évben egy deepfake támadással 2 millió dollárt loptak el egy OKX kriptotőzsde felhasználójától.

1.4. Célkitűzések és módszerek

A dolgozat célja, hogy a deepfake tartalmak felismerésére egy minél gyorsabb és hatékonyabb modellt találjon, amely különböző felhasználási környezetekben, például mobiltelefonokon is alkalmazható. Kezdetben alacsony paraméterszáma és hatékonysága alapján a ResNet-18-at választottam deepfake detektáló modellként. A fejlesztési folyamat során azonban világossá vált, hogy a ResNet-18 és a MobileNetV3 modellek eltérő erősségeik révén különböző feladatokra alkalmasak.

A kutatás során egy olyan modell azonosítására törekedtem, amely különböző környezetekben, például mobilalkalmazásokban vagy akár böngészőbővítményekben is hatékonyan alkalmazható. A tervezett modell alkalmas legyen a képfeltöltésen alapuló elemzésre, amely egy mélyebb és részletesebb vizsgálatot tenne lehetővé, valamint valós idejű monitoring funkcióval is bővíthetne, amely gyorsasága és alacsony erőforrás-igénye révén élő videók elemzésére alkalmas.

A deepfake technológia terjedése súlyos etikai és biztonsági problémákat vet fel, beleértve a dezinformáció terjesztését, magánéleti visszaéléseket, valamint pénzügyi és politikai manipulációkat. Egy ilyen modell lehetőséget adna a felhasználóknak, hogy gyorsan és hatékonyan ellenőrizzék a digitális tartalmak hitelességét. Emellett hozzájárulna a megbízható információáramlás fenntartásához, valamint a deepfake technológiák elleni küzdelemhez.

2. fejezet

Irodalmi áttekintés

A deepfake technológia elleni küzdelem érdekében számos kutatás született, amelyek különböző módszerekkel és modellekkel igyekeztek hatékony detektáló rendszereket kidolgozni. Az alábbiakban néhány kiemelkedő tanulmányt ismertetek, amelyekben különböző megközelítéseket alkalmaztak a cél érdekében.

Z. Li, X. Jin, Q. Jiang, P. Wang, S.-J. Lee, S. Yao és W. Zhou kutatásukban (Combating deepfakes: a comprehensive multilayer deepfake video detection framework) egy háromrétegű keretrendszert fejlesztettek ki a deepfake videók azonosítására.[4]

Az arcok felismerésére a Dlib könyvtárat alkalmazták. Az első réteg a képkockák RGB jellemzőit vizsgálta, amelynek célja a manipulációk okozta térbeli eltérések azonosítása volt. Ehhez a kutatók az Xception modell előre betanított változatát használták, amely a képkockák arcterületeiből nyert vizuális jellemzőket elemezte. A második réteg a generatív modellek, tehát a GAN-ok által hagyott jellemzők felismerésére koncentrált, magas frekvenciájú információk kinyerésével. A harmadik réteg az arc belső és külső régióinak közötti anomáliákra összpontosított, például a felbontás, megvilágítás vagy zaj szintjén jelentkező eltérések alapján.

A módszer hatékonyságát különböző adatkészleteken értékelték, mint például a FaceForensics++, DFD és Celeb-DF (v2). A kísérletek során a keretrendszer kimagasló eredményeket ért el, 99,90%-os AUC-értéket mutatva a FaceForensics++ adatkészleten és 95,40%-ot a Celeb-DF (v2) adathalmazon.

Az előző tanulmányban alkalmazott háromrétegű megközelítéssel szemben Rahul U., Ragul M, Raja Vignesh K. és Tejeswinee K. kutatásukban, Deepfake Video Forensics Based on Transfer Learning című munkájukban a MobileNet alapú transzfer tanulás

módszerére épülő megközelítést alkalmaztak.[5]

A módszer során a videókat képkockákra bontották, majd az arcfelismerő algoritmus segítségével az arcokat kivágták és a hálózatba táplálták. Az arcok detektálására a Multi-task Cascade Convolutional Neural Network (MTCNN) algoritmust alkalmazták. Ez az algoritmus lehetővé teszi az arcok pontos lokalizálását a videók képkockáin belül, viszont a cikk kitér arra is, hogy az MTCNN több arcot jelölt meg hamisan, mint más módszerek, például a Dlib könyvtár arcfelismerő rendszere.

A kutatás három különböző méretű adathalmazzal dolgozott a FaceForensics++ adatkészletből, amelyek 500, 2500 és 15 000 képet tartalmaztak. Az alkalmazott módszer a vizuális hibák azonosításával 86,9%-os pontosságot ért el a hamisított videók detektálásában.

Az eddigi megközelítésekkel szemben a YOLOV3-Based Method for Detecting Deepfake Manipulated Facial Images című cikkben a YOLOv3 algoritmust alkalmazták a deepfake arcképek felismerésére.[6] Ehhez egy kevert adathalmazt használtak valós és deepfake manipulált képekkel. Az adatokat 70%-ban tanításhoz, 10%-ban validáláshoz, és 20%-ban teszteléshez osztották fel. Az adathalmaz képeit előfeldolgozták, beleértve az átméretezést 416×416 képpontos méretre, forgatást és fényerő-korrekciót az adatok változatosságának biztosítása érdekében.

Az arcok detektálása része volt a YOLOv3 általános objektumfelismerési folyamatának. A YOLOv3 modell, amely a Darknet-53 architektúrára épül, gyors és hatékony osztályozást tett lehetővé. Az eredmények szerint a modell 95%-os pontosságot, 92%-os recall, és 94%-os F1-mutatót ért el. Ezek az eredmények kiemelkedő detektálási képességet mutattak a valós és manipulált képek közötti különbségek felismerésében. A YOLOv3 gyorsasága és pontossága azt is bizonyította, hogy a technológia alkalmazható valós idejű deepfake felismerésére, amely releváns lehet a különböző biztonsági és hitelesítési rendszerekben.

A MobileNet architektúra fejlesztésével kapcsolatos kutatások jelentős előrelépéseket mutatnak a deepfake detekció terén, különös tekintettel az aktivációs függvények és regularizációs technikák alkalmazására. Egyik kiemelkedő megoldás a ReLU aktivációs függvényt Dropout-al kombináló MobileReLuDr modell, amely kimagasló tréning pontosságot (99,17%) ért el, ugyanakkor a tesztelési pontosság (70,60%) az overfitting miatt csökkent.[7] Ezzel szemben a ReLU-t L2 regularizációval kombináló MobileReLuL2

modell jobban generalizált a tesztelési adatokra, 72,18%-os pontosságot mutatva. Ezek a módszerek kiemelik a MobileNet könnyű és hatékony felépítését, valamint annak képességét, hogy valós idejű deepfake detekcióra alkalmazható legyen. Kutatásomban ezen eredmények inspirációként szolgáltak, különösen a MobileNet architektúrájának erőforráshatékony alkalmazhatóságának és optimalizálásának szempontjából.

2.1. Összegzés és kapcsolódás a saját munkához

Az irodalmi áttekintés során számos deepfake detekciós módszert ismertettem, amelyek különböző technikákra épülnek, például a MobileNet modell transzfer tanuláson alapuló megközelítésére, a YOLOv3 gyors és hatékony objektumdetektáló algoritmusára, vagy a több rétegű keretrendszerekre, amelyek különféle anomáliák és vizuális jellemzők detektálására specializálódtak. Ezek a megoldások a deepfake technológia eltérő aspektusaira koncentrálnak, mint például a texturális eltérések azonosítása vagy a vizuális hibák kiemelése.

Az irodalomkutatás során az arcdetektáló módszerek közül a Dlib arcfelismerő tűnt a legegyszerűbben használható és pontos opciónak, így a megoldásom során én is ezt használtam fel.

A saját munkám a ResNet-18 modell képességeinek vizsgálatára és optimalizálására összpontosít, amely egyszerűbb architektúrája ellenére versenyképes teljesítményt nyújthat a deepfake tartalmak felismerésében. Emellett a MobileNetV3-Large modell lehetőséget nyújt a valós idejű, erőforrás-korlátozott környezetek kezelésére, mint például mobilalkalmazások. Az irodalomban szereplő módszerek közül különösen a MobileNet által inspirált megközelítés befolyásolták a kutatásomat. A megoldás során a választott architektúrák alkalmazhatóságára és gyakorlati szempontjaira helyeztem a hangsúlyt.

A dolgozat célja, egy olyan erőforráshatékony, pontos és akár valós idejű detektálásra képes modell kidolgozása, amely a mindennapi felhasználók számára is könnyen elérhető és használható.

3. fejezet

Módszerek

3.1. Tanítási környezet

A modelleket Python nyelven, a *PyTorch* keretrendszer használatával implementáltam. A tréninget Google Colab környezetben végeztem, amely biztosította az NVIDIA L4 GPU használatát, ami lehetőséget biztosított a gyors és erőforrás-hatékony modellfejlesztéshez és teszteléshez.

3.2. Dataset

A kutatás során három deepfake-adathalmazt használtam fel: a FaceForensics++, a Deepfake Detection Challenge (DFDC) és a Celeb-DF v2.0 adatbázisokat. Ezek az adathalmazok a különböző deepfake-generálási technológiák széles skáláját reprezentálják, lehetővé téve a modellek sokoldalú tesztelését és finomhangolását.

A **FaceForensics++** adatbázis 1000 valós videót tartalmaz, valamint ezen videók többféle deepfake technológiával (Deepfakes, Face2Face, FaceSwap, NeuralTextures) manipulált változatait. Az adatbázis kiemelkedő részletessége és technológiai sokszínűsége ideális alapot nyújtott a különféle arcmanipulációk felismerésére és osztályozására.[8]

A **Celeb-DF v2.0** adatbázis 590 valódi YouTube-videót és ezek 5639 manipulált változatát tartalmazza. Az adatok etnikai, nemi és életkori sokszínűsége lehetővé tette, hogy a modellek az eltérő csoportokra is általánosítsanak, ami a valós alkalmazási környezetekben alapvető fontosságú.

A **Deepfake Detection Challenge** (DFDC) a legnagyobb nyilvánosan elérhető

deepfake-adathalmaz, több mint 100.000 videoklipet tartalmaz, amelyeket fizetett színészek közreműködésével készítettek. Ez az adathalmaz a generált arcok és arccsere alapú videók széles skáláját öleli fel, ami különösen hasznos a mély tanulási modellek skálázhatóságának teszteléséhez.

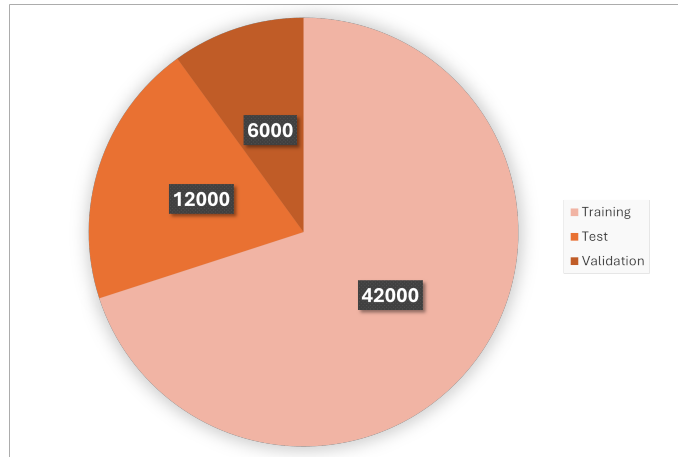
Az elemzés során mindegyik adatbázisból 100 valós és 100 manipulált videót választottam ki, hogy kiegyensúlyozott és reprezentatív mintát biztosítsak a modellek trainingjéhez és validációjához. Ez a mintaválasztás lehetővé tette, hogy az adathalmazok eltérő tulajdonságait integráljam a kutatásba, és biztosítsam a modellek robusztus teljesítményét különböző típusú deepfake tartalmak felismerésében.

3.3. Előfeldolgozás

A projektem során videókat használtam, amelyeket képkockákra bontottam, hogy elegendő mennyiségű képet biztosítsak a feldolgozáshoz. Ez a megközelítés lehetővé tette, hogy a videók különböző „frame”-jein megjelenő arcokat különböző szögekből, megvilágításokkal és arckifejezésekkel rögzítsem. A minél változatosabb képek elérése érdekében egyenletes mintavételt alkalmaztam a videók feldolgozásánál.

Az eljárás bemeneti paraméterként fogadta a videók elérési útvonalát, valamint egy beállítható paraméterrel határoztam meg, hogy hány képkockát szeretnék kinyerni. Ezt az értéket 100-ra állítottam, ami biztosította a teljes videó tartalmának reprezentatív lefedését. Ez egy változatos képi adatbázist biztosított számomra, amely olyan helyzeteket ábrázol, mint például egy utcán való séta, ahol a kamera szöge, a megvilágítás és az árnyékok folyamatosan változnak.

A képkockákra bontás után az adatokat három különálló részre osztottam, amely a 3.1 ábrán látható. Az adathalmaz 70%-a a train mappába került, amelyet a modellek tanítására használtam. A test mappa a képek 20%-át tartalmazta, amelyeken az algoritmus teljesítményét értékeltem, míg a fennmaradó 10% a validation mappába került, amelyet a modellek finomhangolására alkalmaztam. Ez az arányos szétosztás biztosította, hogy a tanulási és értékelési folyamat során az adathalmaz kiegyensúlyozott legyen.



3.1. ábra. A felhasznált képek aránya
Forrás: saját készítés

3.4. Adataugmentáció

Miután az arcokat azonosítottam és kivágtam a képekről, a meglévő adatok módosított változataival bővítettem az adathalmazt az adataugmentáció módszerének segítségével. Az adataugmentáció célja, hogy növelje az adatkészlet változatosságát mesterségesen létrehozott példákkal, ami javítja a modell pontosságát és csökkenti az overfitting kockázatát. A PyTorch könyvtár torchvision.transforms segítségével végeztem el ezeket a műveleteket.

Az általam alkalmazott adataugmentációs technikák a következők voltak:

Transzformáció	Leírás
Resize((224, 224))	Átméretezi a képet 224x224 pixel méretűre
RandomHorizontalFlip()	Vízszintesen tükröz
RandomRotation(15)	Forgatja a képet -15 és +15 fok között
ColorJitter (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1)	Módosítja a kép fényerejét, kontrasztját, telítettségét és árnyalatát

3.1. táblázat. Képfeldolgozási transzformációk és azok leírásai.

3.5. Early Stopping

A modell túltanulásának elkerülése érdekében bevezettem a "korai megállítást" (early stopping) technikáját. Ez nem csak a túltanulást akadályozza meg, hanem csökkenti az tanulási folyamat idő- és erőforrásigényét is azáltal, hogy automatikusan leállítja a tanulást,

ha a modell teljesítménye már nem javul.

A validációs veszteséget minden epoch végén kiszámolja a modellem, és azt az eltérést méri, hogy mennyire pontos a validációs adatokra vonatkozóan. Minél kisebb a validációs veszteség, annál jobban teljesít a modell a validációs adatokon. Ha a validációs veszteség csökken, az azt jelzi, hogy a modell egyre jobban tanul, amennyiben viszont nőni kezd, az arra utalhat, hogy a modell már nem az általános mintákat tanulja meg, hanem az training adathalmaz specifikus részleteit.

3.6. Arcdetektálás

A dolgozatomban arcokra specializált modellt hoztam létre, amelyhez elengedhetetlen volt egy megbízható arcdetektáló eljárás használata. Ehhez a Dlib könyvtárat választottam, amely a Histogram of Oriented Gradients (HOG) módszert használja az arcok detektálására.

A HOG alapötlete, hogy a képet kisebb térbeli egységekre, úgynevezett cellákra bontja. Az egyes cellákban számított gradiens irány- és nagysághisztogramokat egyesítve állítja elő az adott kép, jelen esetben az arc egyedi jellemzővektorát.[9] A módszer célja, hogy az arc karakterisztikus elemeit - például a szemeket, az orrot és az arcéleket - ezekkel a vektorokkal hatékonyan jellemezze. A jellemzők kiszámításakor a kép apró cellákra oszlik, ahol a gradiens irányának és nagyságának hisztogramja kerül meghatározásra. Ezen hisztogramok kombinálásával jön létre a teljes jellemzővektor, amely a detektálás alapját képezi [10].



3.2. ábra. Az objektumdetektálási folyamat vázlata.

Forrás: Saját készítés [9] alapján

A Dlib HOG-alapú arcdetektáló modulja ezen jellemzővektorok alapján, egy lineáris SVM segítségével azonosítja az arcokat. A Dlib könyvtár az alábbi előre betanított funkciót biztosítja az arcdetektálásra:

```
dlib.get_frontal_face_detector()
```

A módszer képes az arcok különböző irányainak felismerésére, beleértve a frontális arcokat, a jobbra és balra fordított arcokat, valamint a balra és jobbra döntött frontális

arcokat.

3.7. Modellek ismertetése

3.7.1. A modellek architektúrája

Resnet-18

A ResNet-18 egy reziduális neurális hálózat, amely 18 rétegből áll. A reziduális blokkok kulcselemei az úgynevezett skip connection-ök, amelyek lehetővé teszik, hogy az információ bizonyos rétegeken „átugorva” közvetlenül a következő blokkhoz jusson. Ez a megoldás kiküszöböli a mély hálózatok gyakori problémáját, a gradiens eltűnését, és javítja a modell tanulhatóságát.

Deepfake detektálás esetén a ResNet-18 hatékony, mivel képes az emberi szem számára alig észrevehető vizuális eltérések, például textúrában, világításban vagy a zajban megjelenő hibák azonosítására.

Az architektúra négy fő szintre tagolódik, amelyek mindegyike két residual blokkot tartalmaz. A konvolúciós rétegek szűrőméretei sorrendben 64, 128, 256 és 512. A hálózat végén található egy globális átlagpooling réteg, amely az utolsó konvolúciós réteg által generált 512-dimenziós jellemzővektort tömöríti.[11]

A ResNet-18 architektúra implementálásához testreszabtam a ResNet-18 hálózatot, hogy egy bináris osztályozót kapjak. A torchvision könyvtár által biztosított ImageNet adathalmazon előre betanított ResNet-18 modellt használtam alapként. Az eredeti modell végén található teljesen összekapcsolt (fully connected) réteget módosítottam, amely eredetileg 1000 osztályra ad előrejelzést. Ezt a réteget lecseréltem egy egyedi, szekvenciális rétegre, amely először egy lineáris réteggel 256 neuronosra csökkenti a bemeneti jellemzőket, majd egy ReLU aktivációs függvény következik.

A túltanulás elkerülése érdekében egy 48%-os dropout réteget is beillesztettem, végül pedig egyetlen neuront tartalmazó lineáris réteggel zártam a hálózatot, amely a bináris osztályozáshoz szükséges kimenetet adja.

MobileNetV3 - Large

A MobileNet család konvolúciós neurális hálózatok gyűjteménye, amelyeket gyorsaságra és pontosságra optimalizáltak. A hálózatot elsősorban könnyű és gyors működésre tervezték. Alkalmas képosztályozásra, objektumfelismerésre és szemantikai szegmentációra alacsony számítási kapacitású eszközökön, mint például mobiltelefonokon.[13]

A dolgozatomban a MobileNetV3-Large modellt bináris osztályozási feladatra szabtam át. A fő módosítás az osztályozó rétegeknél történt, ahol az eredeti, 1000 osztályú kimenetet biztosító rétegeket egy egyszerűsített, bináris osztályozásra optimalizált struktúrára cseréltem.

Az új osztályozó réteg első lépésként a MobileNetV3 által generált 1280 dimenziós jellemzőket egy lineáris réteggel 256 dimenzióra csökkenti, majd egy ReLU aktivációs függvény alkalmazásával biztosítja a modell nemlinearitását. A túltanulás megelőzése érdekében egy dropout réteg is beépítésre került, amely az edzés során véletlenszerűen nullázza a neuronok egy részét. Végül az utolsó lineáris réteg egyetlen logits értéket generál, amely lehetővé teszi a bináris osztályozást, megkülönböztetve a deepfake és valós tartalmakat.

Az architektúra hatékonyságát olyan modern megoldások biztosítják, mint a depth-wise separable convolution és a hard-swish aktivációs függvény, amelyek jelentősen csökkentik a számítási költségeket, miközben megőrzik a magas pontosságot. [12] Az így testreszabott MobileNetV3-Large egy gyors és energiahatékony modellként szolgál, leginkább alkalmassá téve a mobil eszközökön történő mélyhamisítvány-elemzésre. Emellett biztosít gyors feldolgozási sebességet és a megbízható detektálási teljesítményt.

3.7.2. Veszteségfüggvény: BCEWithLogitsLoss

A bináris osztályozási feladatok esetében a veszteségfüggvény kiválasztása alapvetően meghatározza a modell tanulási hatékonyságát és stabilitását. A BCEWithLogitsLoss veszteségfüggvényt használtam, amely a bináris keresztentropia veszteséget és a szigmoid aktivációs függvényt egyetlen lépésben egyesíti.[14] Ez a kombináció alkalmas bináris osztályozásra, mivel az aktiválatlan logitokat közvetlenül kezelve elkerülhetők a szigmoid függvénynél gyakran előforduló numerikus instabilitási problémák.

A BCEWithLogitsLoss nemcsak numerikus stabilitást biztosít, hanem gyorsabb

számítást is lehetővé tesz, mivel az aktiváció és a veszteségszámítás integráltan történik.

3.7.3. Optimalizációs algoritmus: AdamW

Az optimalizáció során az AdamW algoritmust alkalmaztam, amely az Adam optimalizátor továbbfejlesztett változata. Az AdamW különlegessége, hogy a súlycsökkentést (weight decay) elkülöníti az adaptív súlyfrissítésektől, így hatékonyabb és stabilabb szabályosítást tesz lehetővé. A hagyományos Adam algoritmusnál a súlycsökkentés és a tanulási ráta kombinációja néha túl nagy súlyokat eredményezhet, amit az AdamW kiküszöböl. [15]

3.7.4. Konvolúciós mechanizmusainak elemzése

A ResNet-18 és a MobileNetV3-Large modellek eltéréseket mutatnak a konvolúciós mechanizmusok és számítási hatékonyság szempontjából. A ResNet-18 hagyományos konvolúciós műveletekre épül, míg a MobileNetV3-Large a Depthwise Separable Convolution technológiát alkalmazza a számítási igény csökkentése érdekében. Az alábbiakban a két architektúra számítási költségeit és hatékonyságát hasonlítom össze egy 224×224 -es bemeneti képméret esetén.

ResNet-18 számítási igénye

A ResNet-18-ban a hagyományos konvolúciók számítási igénye nagyobb a csatornák közötti kapcsolatok és a teljes térbeli pozíciók feldolgozása miatt. Az én modellem esetében, egy középső rétegnél, ahol:

- A bemeneti csatornák száma: $M = 64$,
- A kimeneti csatornák száma: $N = 128$,
- A térbeli méret: $D_f = 224/4 = 56$ ¹
- A szűrő mérete: $D_p = 3$,

a számítási igény a következőképpen alakul:

$$C_{\text{ResNet-18}} = D_p \cdot D_p \cdot M \cdot N \cdot D_f \cdot D_f$$

¹Az első konvolúciós réteg (7×7 -es szűrő, $\text{stride} = 2$) az eredeti 224×224 -es képméretet 112×112 -re csökkenti. Ezt követően egy 3×3 -as max pooling réteg ($\text{stride} = 2$) tovább csökkenti a méretet 56×56 -ra.

$$C_{\text{ResNet-18}} = 3 \cdot 3 \cdot 64 \cdot 128 \cdot 56 \cdot 56$$

$$C_{\text{ResNet-18}} = 231211008 \text{ művelet}$$

MobileNetV3-Large Inverted Residual Block számítási igénye

Az Inverted Residual Block a *Depthwise Separable Convolution* alkalmazásával két lépésben dolgozza fel az adatokat: a depthwise convolution során csatornánként, a pointwise convolution során pedig csatornák között végez műveleteket. Az adott blokk paraméterei:

- A bemeneti csatornák száma: $M = 16$,
- A kimeneti csatornák száma: $N = 16$,
- A térbeli méret: $D_f = 112$,
- A szűrő mérete: $D_k = 3$.

Depthwise convolution: A *depthwise convolution* minden csatornát önállóan dolgoz fel, ezért a számítási igény a következőképpen írható fel:

$$C_{\text{depthwise}} = D_k \cdot D_k \cdot M \cdot D_f \cdot D_f$$

$$C_{\text{depthwise}} = 3 \cdot 3 \cdot 16 \cdot 112 \cdot 112$$

$$C_{\text{depthwise}} = 1806336 \text{ művelet}$$

Pointwise convolution: Az 1×1 -es szűrő a csatornák közötti kapcsolatok feldolgozására szolgál. A MobileNetv3 architektúrája alapján, itt a kimeneti csatornák száma 24-re módosul. A számítási igény a következőképpen írható fel:

$$C_{\text{pointwise}} = M \cdot N \cdot D_f \cdot D_f$$

$$C_{\text{pointwise}} = 16 \cdot 24 \cdot 112 \cdot 112$$

$$C_{\text{pointwise}} = 4816896 \text{ művelet}$$

Teljes számítási igény: A teljes számítási igény a két rész összege:

$$C_{\text{Inverted Residual}} = C_{\text{depthwise}} + C_{\text{pointwise}}$$

$$C_{\text{Inverted Residual}} = 1806336 + 4816896$$

$$C_{\text{Inverted Residual}} = 6623232 \text{ művelet.}$$

Összehasonlítás

A számítási igény aránya a két modell között:

$$\text{Arány} = \frac{C_{\text{MobileNetV3}}}{C_{\text{ResNet-18}}}$$

$$\text{Arány} = \frac{6623232}{231211008} \approx 0.02864583$$

Ez azt jelenti, hogy a MobileNetV3-Large számítási igénye a ResNet-18 modell-hez képest körülbelül 2,86%-ot igényel, miközben hasonló teljesítményt biztosít. Fontos hangsúlyozni azonban, hogy ez a számítás kizárólag a konvolúciós mechanizmusok különbségeire kívánt rávilágítani, és nem elegendő egyetlen réteg alapján messzemenő következtetéseket levonni a modellek teljesítményéről és alkalmazhatóságáról.

A Spatial Sharing of GPU for Autotuning DNN Models című kutatás azt vizsgálta, hogy a GPU erőforrások eltérő kiosztása hogyan hat a ResNet-18 és MobileNetV3 modellek futtatási idejére.[16] Az eredmények alapján, amelyek a 3.3 ábrán is láthatóak, a MobileNetV3 teljesítménye jobb, ha 25%-os GPU-erőforrással hangolták, és 25%-os GPU-kapacitáson futtatták, ezzel szemben a ResNet-18 teljesítménye jelentősen visszaesett alacsony GPU-kapacitáson, különösen akkor, ha az optimalizálás során magasabb GPU-erőforrást használtak.

Ez megmutatja, hogy a MobileNetV3 jobban alkalmazkodik a kevesebb erőforráshoz, míg a ResNet-18 inkább nagyobb GPU-teljesítmény mellett működik optimálisan. Az eredmények rávilágítanak arra, hogy a modellválasztást jelentősen befolyásolja az alkalmazási környezet és az elérhető erőforrások. A ResNet-18 nagy pontosságú és érzékeny feladatokhoz ideális, míg a MobileNetV3 kisebb számítási igénye miatt valós idejű, erőforrás-korlátozott környezetekben, például mobil- és beágyazott rendszerekben hatékony.

Inference%	Tuning% (ResNet-18)						Inference%	Tuning% (Mobilenet)					
	10	25	50	75	100	Untuned		10	25	50	75	100	Untuned
10	3.13	3.35	4.03	4.32	4.38	13.18	10	1.70	1.94	2.20	2.34	2.59	2.45
25	2.06	1.83	2.03	2.10	2.18	5.54	25	1.24	0.97	1.06	1.13	1.22	1.32
50	1.75	1.41	1.34	1.43	1.42	3.06	50	1.13	0.77	0.67	0.71	0.73	0.79
75	1.71	1.27	1.21	1.17	1.22	2.24	75	1.13	0.73	0.64	0.62	0.62	0.75
100	1.64	1.22	1.17	1.11	1.10	1.87	100	1.13	0.72	0.60	0.59	0.57	0.75

3.3. ábra. A modellek inferencia-időit (ms) mutatja különböző GPU-hangolási és inferencia-kapacitás százalékos értékein.

Forrás: [16]

Következtetés

Az összehasonlítás alapján a ResNet-18 és a MobileNetV3-Large modellek számítási igényei jelentősen különböznek. A ResNet-18 összesen 3.63 GFLOPs-ot használ, ami a mélyebb rétegek és a hagyományos konvolúciós műveletek számítási terhét tükrözi. Ezzel szemben a MobileNetV3-Large számítási igénye mindössze 0.43 GFLOPs, ami a hatékony depthwise separable convolution és invertált reziduális blokkok alkalmazásának köszönhető. Ez azt jelenti, hogy a MobileNetV3 számítási igénye a ResNet-18-nak mindössze 11,8%-a, jelentősen csökkentve az energiafogyasztást és növelve a sebességet.

A ResNet-18 magas számítási igénye miatt előnyös lehet olyan környezetekben, ahol nagy számítási kapacitás áll rendelkezésre, például szervereken vagy dedikált GPU-kkal rendelkező gépeken, mivel pontosabb lehet komplex mintázatok felismerésében. A MobileNetV3-Large ezzel szemben kiválóan alkalmazható alacsony erőforrás-igényű környezetekben, például mobil eszközökön vagy beágyazott rendszerekben, ahol a valós idejű feldolgozás és az energiahatékonyság kulcsfontosságú.

Összességében a ResNet-18 és a MobileNetV3-Large eltérő célokat szolgál. A ResNet-18 nagyobb pontosságot nyújthat nagyobb számítási kapacitású rendszereken, míg a MobileNetV3-Large hatékonyságával ideális választás valós idejű alkalmazásokhoz, különösen olyan esetekben, ahol a gyors feldolgozás és alacsony számítási terhelés prioritást élvez.

4. fejezet

Eredmények

4.1. Módszertan és metrikák

A kísérleti eredmények értékelése során több teljesítménymutatót használtam a modellek összehasonlítására. Ezek közé tartozik a ROC-AUC, a precízió, a recall, az F1-mutató, valamint az inferencia idő. Az alábbiakban részletesen bemutatjuk az eredményeket és ezek értelmezését.

Precízió

A precízió a helyesen azonosított deepfake képek aránya az összes deepfake-ként jelölt képekhez képest.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.1)$$

Recall

A recall a helyesen azonosított deepfake képek aránya az összes tényleges deepfake képekhez képest:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.2)$$

ROC Görbe és AUC (Area Under Curve)

A ROC – *Receiver Operating Characteristic* görbe a modell teljesítményét a valós pozitív(TPR) és hamis negatív(FPR) kimenetek viszonyát ábrázolja. [17].

Az *Area Under Curve* (AUC) a ROC görbe alatti területet jelenti, amely a modell

teljesítményének átfogó mérőszáma. A jól teljesítő modellek esetében az AUC közel van az 1-hez, míg a véletlenszerű osztályozás AUC értéke 0,5 körül mozog.

F-score

Az F-score egy olyan mérőszám, amely a bináris osztályozási pontosságot mutatja, és a precízió és recall harmonikus átlaga.[2] Az F1-score, az F-score egyik speciális esete, ahol a precízió és recall egyenlő súlyt kap:

$$F1 = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (4.3)$$

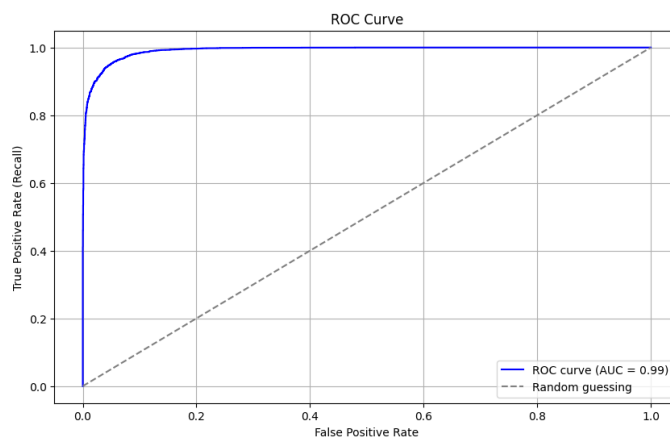
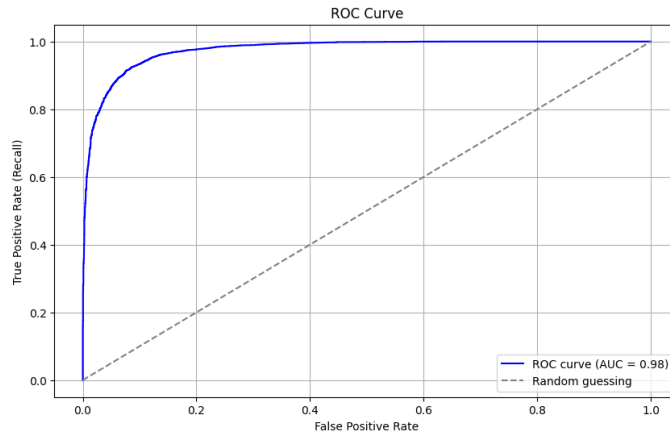
4.2. Eredmények

Az összehasonlított ResNet-18 és MobileNetV3 modellek teljesítményének részletes eredményeit a 4.1 táblázatban található metrikák mutatják be.

Metrika	Resnet-18	MobileNetV3	Jobban teljesített
ROC-AUC	0.9916	0.9829	Resnet-18
Precízió (Precision)	0.8915	0.9490	MobileNetV3
Visszahívás (Recall)	0.9927	0.921	Resnet-18
F1-mutató	0.9394	0.9348	Resnet-18

4.1. táblázat. A két modell teljesítményének összehasonlítása.

Az eredmények alapján a ResNet-18 modell AUC értéke magasabb (0.9916), mint a MobileNetV3 modellé (0.9829). Ez azt mutatja, hogy a ResNet-18 valamivel pontosabban képes azonosítani a deepfake tartalmakat, és jobban különbözteti meg az osztályokat. Bár a különbség nem jelentős, a ResNet-18 teljesítménye közelebb áll az optimális 1-es értékhez, ami azt jelzi, hogy a modell erősebb detekciós képességekkel rendelkezik, leginkább olyan esetekben, amikor a finom részletek felismerése kritikus. Ezen eredmények a 4.1 ábrán láthatóak.



4.1. ábra. MobileNetV3 és ResNet-18 ROC görbék összehasonlítása.

4.3. Eredmények összehasonlítása

Az eredmények alapján a ResNet-18 és a MobileNetV3 modellek különböző metrikák szerint teljesítettek jobban, ami rámutat az erősségeik és gyengeségeik közötti eltérésekre.

A ROC-AUC és a recall tekintetében a ResNet-18 bizonyult jobbnak, ami azt jelzi, hogy hatékonyabban különbözteti meg a deepfake és valós tartalmakat, és érzékenyebben reagál a valódi deepfake-ek azonosítására. Ugyanakkor a MobileNetV3 a precízió szempontjából jobban teljesített, ami kevesebb hamis pozitív azonosítást eredményez.

Az F1-mutató szerint a ResNet-18 valamivel jobb, mivel kiegyensúlyozott teljesítményt nyújtott a precízió és a visszahívás között, de a MobileNetV3 modell is csak pár ezreddel marad el.

4.4. Valós idejű detekció tesztelése

A valós idejű detekció kritikus szerepet játszik a deepfake tartalmak elleni küzdelemben, például a böngészőbővítmények esetében, ahol azonnali visszajelzés szükséges a felhasználók számára. A MobileNetV3 modell frame skip=5 beállítással képes volt hatékonyan elemezni egy 34 másodperces, 25 FPS sebességű videót, amelyet 33,13 másodperc alatt dolgozott fel CPU-n. Ez a beállítás lehetővé tette, hogy a modell minden ötödik képkockát dolgozzon fel, ezzel jelentősen csökkentve a számítási terhelést, miközben elegendő adatot biztosított a hitelesség ellenőrzéséhez.

A frame skip mechanizmus alkalmas híroldalak és közösségi média tartalmainak elemzésére, ahol a dinamikus változások jellemzően lassabbak. Ez az eljárás optimalizálja a feldolgozást, miközben fenntartja a magas pontosságot, és minimalizálja a számítási erőforrások igénybevételét. Az ilyen rendszer hatékonyan működhet böngészőbővítményként, amely valós időben képes az élő közvetítések vagy videók feldolgozására, figyelmeztetve a felhasználókat a manipulált tartalmakra. A MobileNetV3 architektúrájának gyorsasága és alacsony számítási igénye miatt ez a modell ideális választás lehet CPU-alapú rendszerekben is, ahol a feldolgozási sebesség kulcsfontosságú.

A rendszer továbbfejlesztésére számos lehetőség kínálkozik. A frame skip értékének dinamikus szabályozása révén a modell alkalmazkodhatna a különböző jelenetek komplexitásához. Az eredmények azt mutatják, hogy a MobileNetV3 modell nemcsak valós idejű detekcióra alkalmas, hanem egyben megbízható eszközt biztosíthat a felhasználók számára a deepfake tartalmak hitelességének gyors és hatékony ellenőrzésére, miközben a böngészők erőforrás-korlátozott környezetében is megfelelő teljesítményt nyújt.

5. fejezet

Értékelés és következtetések

Az elvégzett kísérletek alapján világosan látszik, hogy a ResNet-18 és MobileNetV3 modellek eltérő előnyeiket tekintve különböző felhasználási területeken lehetnek ideálisak. A ResNet-18 modell kimagasló pontosságot és érzékenységet mutatott a deepfake tartalmak felismerésében, különösen olyan helyzetekben, ahol a maximális recall és a részletek alapos elemzése a cél, megfelelő számítási erőforrások rendelkezésre állása mellett. Ezzel szemben a MobileNetV3 gyorsasága és alacsony számítási igénye miatt erőforráskorlátozott környezetekben, például mobiltelefonokon vagy beágyazott rendszerekben, nyújt kiemelkedő teljesítményt.

Ezekre az eredményekre építve egy képfeltöltésen alapuló elemző rendszer fejlesztése releváns lehet a deepfake tartalmak detektálására. Ez a megoldás akár mobilalkalmazásban, akár desktop alkalmazásban is hatékonyan alkalmazható lenne. A rendszer egyik kulcsfunkciója egy fájlfeltöltő modul, amely lehetővé teszi a felhasználók számára, hogy feltöltsék az elemzendő képeket vagy videókat. Ehhez a ResNet-18 modell lenne alkalmas, amely a statikus tartalmak elemzésében nagy pontosságot nyújt, és megbízható eredményekkel segíti a deepfake tartalmak felismerését.

A valós idejű monitoring funkciót egy böngészőbővítmény formájában lehetne megvalósítani, amely híroldalakon és közösségi média platformokon megjelenő vizuális tartalmak valós idejű elemzésére alkalmas.

A MobileNetV3 modell alacsony számítási igénye és gyors feldolgozási sebessége lehetővé teszi a valós idejű feldolgozást a böngészőben, ezzel segítve a felhasználókat az online tartalmak hitelességének ellenőrzésében. Bár a böngészőbővítmények használata mobiltelefonokon jelenleg korlátozott, desktop környezetben hatékonyan működhet,

és megfelelő eszközt nyújtana a híroldalakon és közösségi média felületeken terjedő manipulált tartalmak azonosítására.

Ez a megközelítés különböző platformokon biztosítana eszközöket a deepfake tartalmak detektálásához: a képfeltöltés alapú elemzés mobil- és asztali alkalmazásokon keresztül érhető el, míg a valós idejű monitoring funkció böngészőbővítményként nyújt segítséget a híroldalakon és közösségi médiában megjelenő vizuális tartalmak gyors és hatékony elemzéséhez.

Ez az applikációs javaslat nemcsak a deepfake detektálás gyakorlati megvalósítását demonstrálná, hanem felhasználóbarát módon tenné elérhetővé a technológiát, elősegítve a hamisított tartalmak elleni küzdelmet.

Összefoglalás

A dolgozatban bemutatott kutatásom a deepfake detekció területén olyan megközelítést kínál, amely az irodalmi módszerek legjobb elemeit integrálja.

Az irodalomban tárgyalt módszerek, mint például az Xception modell és a YOLOv3, kiváló teljesítményt mutatnak a nagyobb dataseteken vagy valós idejű alkalmazásokban. Azonban ezek gyakran magas számítási igényt támasztanak, és nem lehet használni erőforrás-korlátozott környezetekben. Ezzel szemben az általam alkalmazott ResNet-18 és MobileNetV3 modellek egyszerűbb architektúrája révén jelentős előnyt biztosítanak.

Pontosság téren leginkább a ResNet-18 modell volt kiemelkedő a deepfake tartalmak azonosításában, azonban a MobileNet se maradt le szignifikáns különbséggel. Emellett alacsonyabb számítási igényvel rendelkeznek, mint a többretegű modell.

A tesztek alapján mind a ResNet-18, mind a MobileNetV3-Large modellek alkalmasak valós idejű deepfake detekcióra, a ResNet-18 nagyobb pontosságot és gyorsabb inferencia-időt nyújt GPU-val, míg a MobileNetV3-Large alacsonyabb számítási igényével hatékonyan működik CPU és erőforrás-korlátozott környezetekben is.

A kutatás célja nemcsak a pontosság maximalizálása volt, hanem annak biztosítása is, hogy a modellek mezei felhasználó környezetben alkalmazhatóak legyenek, ezt a MobileNetV3 és a ResNet modell az architektúrája révén teljesíteni tudja más-más célokra.

Dolgozatomban sikerült a MobileNetV3 architektúrát optimalizálva jobb eredményeket elérni, mint az irodalomban közölt hasonló megoldások. Az irodalomban megjelenő MobileNet modell a tesztelési pontosság alapján 72,5%-os eredményt ért el, amit az előre betanított MobileNet architektúra és a transfer learning módszerének alkalmazásával értek el. Ezzel szemben az általam optimalizált MobileNetV3 modell lényegesen jobb teljesítményt nyújtott. A MobileNetV3 modell a precizitás terén 0,9490, míg a recall

esetében 0,921 értéket ért el, amelyek lényegesen magasabbak, mint a korábbi MobileNet verziók eredményei. Az általam elért ROC-AUC (0,9829) és F1-mutató (0,9348) szintén azt mutatják, hogy a modell képes pontosan és megbízhatóan osztályozni a deepfake videókat, még akkor is, ha az erőforrások korlátozottak.

Az eredmények rávilágítanak arra, hogy megfelelő optimalizálással a MobileNetV3 gyorsasága és alacsony számítási igénye mellett is képes magas pontosságot biztosítani, megbízható megoldást kínálva a deepfake detekció területén, ezért kiváló lehet valós időben történő deepfake detektációra is egy böngészőben bővítményként.

Összességében a kutatásomban bemutatott módszerek egyensúlyt teremtenek a pontosság és hatékonyság között. Az eredmények rávilágítanak arra, hogy egyszerűbb modellekkel is elérhető a korszerű megoldásokhoz mérhető teljesítmény, amely lehetővé teszi a technológia szélesebb körű, gyakorlati alkalmazását.

Köszönetnyilvánítás

Köszönöm Dr. Storcz Tamás témavezetőmnek a dolgozat során adott hasznos ötleteit és jó tanácsait.

Külön köszönettel tartozom családomnak és barátaimnak, akik türelmükkel, biztatásukkal és támogatásukkal végig mellettem álltak a dolgozat elkészítése során.

Irodalomjegyzék

- [1] R. Budhiraja, M. Kumar, M. K. Das, A. S. Bafila¹, A. Pundir, S. Singh¹ (2024): *MaD-CoRN: an efficient and light weight deepfake detection approach using convolutional reservoir network* Springer Nature
- [2] V. K. Sharma R. Garg Q. Caudron (2024): *Asystematic literature review on deepfake detection techniques* Springer Nature
- [3] Z. Li, X. Jin, Q. Jiang, P. Wang, S.J. Lee, S Yao, W. Zhou (2024): *Crafting imperceptible and transferable adversarial examples: leveraging conditional residual generator and wavelet transforms to deceive deepfake detection* Springer Nature
- [4] N. Rathoure (2024): *Combating deepfakes: a comprehensive multilayer deepfake video detection framework* Springer Nature
- [5] Rahul U, Ragul M, Raja Vignesh K, Tejeswinee K (2020): *Deepfake Video Forensics based on Transfer Learning* pp. 5069-5073 International Journal of Recent Technology and Engineering (IJRTE)
- [6] M. KARAKÖSE, H. YETİŞ, M. ÇEÇEN (2024): *A YOLOV3-Based Method for Detecting Deepfake Manipulated Facial Images* Turkish Journal of Science Technology
- [7] H. Noprisson, V. Ayumi, M. Purba, N. Ani (2024): *Mobilenet Performance Improvements For Deepfake Image Identification using Activation Function and Regularization* LPPM Universitas Nusa Mandiri
- [8] A. Rossler, D. Cozzolino, L Verdoliva, C. Riess, J. Thies, M. Nießner (2019): *FaceForensics++: Learning to Detect Manipulated Facial Images*

-
- [9] N. Dalal and B. Triggs (2005): *Histograms of Oriented Gradients for Human Detection* 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
- [10] M. Preethi, C. Sunitha, M. Parameshvar, B. Dharshini, S Gokul (2023): *Emotion Detection using HOG for Crime Detection*. pp. 3617–3626. Indian Journal of Science and Technology.
- [11] P. Mhasakar, P. Trivedi, S. Mandal, S. K. Mitra (2021): *Handwritten Digit Recognition Using Bayesian ResNet* Springer Nature Computer Science
- [12] A. Howard, M. Sandler, G. Chu, L-C Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R Pang, V. Vasudevan, Q. V. Le, Hartwig A. (2019): *Searching for MobileNetV3* 2019 IEEE/CVF International Conference on Computer Vision (ICCV)
- [13] R. Singh (2024): *Understanding and Implementing MobileNetV3* Medium <https://medium.com/@RobuRishabh/understanding-and-implementing-mobilenetv3-422bd0bdfb5a>
- [14] R Li, S. Halder, D. Nagaraj (2020): *Reconstructing temporally high resolution pediatric MRI scans from low resolution images* Stanford University
- [15] X. Wang (2024): *How to set AdamW's weight decay as you scale model and dataset size* College of Information and Computer Science University of Massachusetts Amherst
- [16] A. Dhakal, J. Cho, S. G. Kulkarni, K. K. Ramakrishnan, P. Sharma (2020): *Spatial Sharing of GPU for Autotuning DNN models* arXiv. <https://doi.org/10.48550/arXiv.2008.03602>
- [17] Stroc T. (2020): *Intelligens Rendszerek* Pollack Press, Pécs

Ábrák jegyzéke

1.1. Az arcmanipuláció típusai	4
3.1. A felhasznált képek aránya	12
3.2. Az objektumdetektálási folyamat vázlata.	13
3.3. A modellek inferencia-időit (ms) mutatja különböző GPU-hangolási és inferencia-kapacitás százalékos értékein.	19
4.1. MobileNetV3 és ResNet-18 ROC görbék összehasonlítása.	22

Táblázatok jegyzéke

3.1. Képfeldolgozási transzformációk és azok leírásai.	12
4.1. A két modell teljesítményének összehasonlítása.	21