**Mesterséges intelligencia által generált tartalom detektálása**

**Önálló Labor**

**Nagy László, BC7TB3**

**Használt adatok**

A neurális háló tanítása 5, a kaggle.com-on megtalálható képgyűjteménnyel történt. Ezek egy része eleve szét volt választva tanítási és tesztelési adathalmazra, viszont a legtöbbnél csak az alapján voltak megkülönböztetve a képek, hogy AI generálta-e vagy sem.

Így az első lépés az adatok előkészítése volt: mindegyik adathalmazt egységesen 80%-20% arányban szétbontottam. A képek 80%-a a tanításhoz lett felhasználva, a maradék 20% pedig a teszteléshez.

Ezután ezeket az adathalmazokat külön-külön kipróbálta. Egyszerre egy halmaz tanítási képeit felhasználva tanítottam a hálót, majd az adott halmaz teszt képein megvizsgáltam, hogy mennyire volt hatékony a tanulás. Erre azért volt szükség, mert ha már az adott adathalmaz sem tudott volna értelmezhető eredményt elérni a tanításban, akkor nem valószínű, hogy a későbbiekben hasznos lett volna. Az eredmények:

Cash Bowman – AI Generated Images vs Real Images

<https://www.kaggle.com/datasets/cashbowman/ai-generated-images-vs-real-images>

Eredmény: 60,25%

Epoch szám: 3

Batch méret: 4

Jordan J. Bird – CIFAKE: Real and AI-Generated Synthetic Images

<https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images>

Eredmény: 88,22%

Epoch szám: 1

Batch méret: 64

SuperPotato9 – Ai recognition dataset

<https://www.kaggle.com/datasets/superpotato9/dalle-recognition-dataset>

Eredmény: 84,82%

Epoch szám: 3

Batch méret: 64

Shahzaib Ur Rehman – Detect AI-Generated Faces: High-Quality Dataset

<https://www.kaggle.com/datasets/shahzaibshazoo/detect-ai-generated-faces-high-quality-dataset>

Eredmény: 95,3%

Epoch szám: 1

Batch méret: 32

Sunny Kakar – Shoes Dataset: Real and AI-Generated Images

<https://www.kaggle.com/datasets/sunnykakar/shoes-dataset-real-and-ai-generated-images>

Eredmény: 91,74%

Epoch szám: 2

Batch méret: 32

Ezek alapján az első adathalmaz (AI Generated Images vs Real Images) nem elég jó ahhoz, hogy a továbbiakban használni lehessen.

Újabb probléma, hogy ezekben az adathalmazokban nem mindig ugyan annyi AI által generált kép van, mint rendes kép. Emiatt hogyha összerakjuk őket egy nagy adathalmazzá, akkor több AI által generált kép fog szerepelni a tanítási adathalmazban (és a teszt adathalmazban is), mint rendes kép.

Erre a problémára két megoldást próbáltam ki. Elsőre kitöröltem annyi AI által generált képet, amennyivel több volt belőlük, így az arányok már helyesek voltak. Ezt a módosított adathalmazt felhasználva a modell már értelmezhető eredményeket ért el a tanulás után.

A második megoldás az augmentáció volt az adathalmazon. A rendes képekből 180 fokos elforgatással generáltam „új” képeket. Ezt annyi képpel tettem meg, amennyivel kevesebb volt eredetileg, mint AI generált kép. Így az első módszerhez hasonlóan az arányok megfelelőek voltak, viszont meglévő adatok eldobása helyett újak kerültek be a tanulási és tesztelési adathalmazba, ami jobb megoldásnak bizonyult: míg az első megoldással 80,4%-os teszt eredményt sikerült elérni, a másodikkal 86,3%.

**A képek feldolgozása**

A képek, amikből tanul, amin tesztel és amit később osztályoz a modell, nem azonos méretűek, és nem is feltétlen azonos képarányúak. Emiatt szükség van egy egységes formátum, amire a program átalakít minden képet, mielőtt feldolgozná őket.

Első lépésként a bemeneti képet egy 240x240 pixeles képpé alakítja úgy, hogy a közepéből vág ki egy ekkora képet.

Ezután a képpontok RGB értékeit 0-255 helyett 0-1 tartományba skálázza.

Végül az RGB értékeket 0-1 tartomány helyett -1 és 1 közötti értékekre skálázza. Ez a neurális háló tanításában előnyösebb, mint a 0-1 tartomány.

**A modell**

Az előkészített képeket egy konvolúciós neurális háló dolgozza fel.

Az első konvolúciós réteg bemeneti csatornáinak száma 4, mivel az RGB miatt 3, plusz 1 a Fourier-spektrum csatornával. Ez a réteg 32 darab 3x3-as szűrőt alkalmaz.

Ezután egy méretcsökkentő réteg jön, amely minden 2x2-es régióból csak a legnagyobb értéket tartja meg, így a kép szélességét és magasságát is a felére csökkenti.

A második konvolúciós réteg bemeneti csatornáinak száma 32, szűrőinek száma pedig 64.

Az első és a második konvolúciós réteg után is alkalmazzuk a méretcsökkentő réteget, így az eredetileg 240x240-es kép 60x60-ra csökken. Mindkét konvolúciós réteg után, de méretcsökkentés előtt az értékek egy ReLu rétegen is átmennek.

Ezután a 2 dimenziós kép pixeleit 1 dimenzióssá tesszük (flatten), ami így már alkalmas egy neurális háló bemeneteként viselkedni. A háló 2 rétegből áll: az első bemenetének mérete 230 400 (64 x 60 x 60, ahol 60 x 60 a kép mérete, 64 pedig a csatornák száma), kimenetének mérete 128. A második réteg bemeneti mérete az első kimenetének mérete, azaz 128, kimenetének mérete pedig 2, hiszen a modell 2 osztályba fogja sorolni a képet (valós vagy AI generált). A két réteg között egy ReLu réteg is elhelyezkedik.

A modell felépítése így:

* konvolúciós réteg 1
* ReLu
* méret felezés (pool)
* konvolúciós réteg 2
* ReLu
* méret felezés (pool)
* laposítás (flatten)
* neurális háló 1. rétege
* ReLu
* neurális háló 2. rétege

Az első konvolúciós réteg előtt történik a Fourier-spektrum számítása. Ez az RGB 3 értéke mellé érkezik 4. értéknek minden pixelhez. Ehhez a Pytorch beépített függvényeit használtam.

A Fourier-spektrum használatának értelme az, hogy olyan plusz információt kapjon a neurális háló, amit az eredeti RGB értékek önmagukban nem tudnának megadni. A Fourier-spektrum azt mutatja meg, hogy milyen típusú frekvenciakomponensek (ismétlődő mintázatok, textúrák) vannak jelen egy képen. A lassan változó minták frekvenciája alacsony, míg a gyorsan változóké magasabb. Egy valódi fénykép inkább lassabb változásokat tartalmaz, míg egy AI generált gyakran magas frekvenciájú, furcsa, esetleg ismétlődő textúrákból áll, amit ezzel az értékkel jól lehet jellemezni.

Először minden pixelre csatornánként kiszámítja a 2D diszkrét Fourier-transzformációt. Ebben lehetnek komplex számok is. Ezen értékek segítségével meg lehet mondani, hogy a kép adott része milyen frekvenciájú mintázatokat tartalmaz.

Mivel jelen esetben még vannak komplex számok a spektrum értékekben, ezeket valós számokká alakítja. Ezzel megmarad a frekvencia értékben tárolt jelentés, tehát továbbra is használható az eredeti ötletre a spektrum, viszont valós számként már a neurális háló is fogja tudni használni.

Azonban itt még 3 spektrum érték van pixelenként, amit átlagolunk, így már csak 1 marad. Ezeket normalizáljuk, hogy az érték 0 és 1 közé essen. Végül pedig a megmaradt 1 darab spektrum értéket pixelenként a meglévő RGB értékek mellé tesszük.