Multimédia tartalmak intelligens feldolgozása - Házi feladat dokumentáció

Távközlési és Médiainformatikai Tanszék

Készítette:	**Benda Krisztián**
Neptun-kód:	**J1CEI3**
Ágazat:	**Adat- és Médiainformatika**
E-mail cím:	**krisztianbenda@gmail.com**
Konzulens(ek):	**Dr. Szűcs Gábor**
E-mail címe(ik):	**szucs@tmit.bme.hu**

Téma címe: Videókon arcok érzelmi elemzése

Feladat

A 2021-as évben a szorgalmi házi feladat **videókon arcok érzelmi elemzése** témakörhez kapcsolódik. Ehhez az osztályozási feladathoz rendelkezésre áll egy tanulóállomány (felcímkézett érzelem információkkal: boldogság, düh, szomorúság, undor, meglepetés és félelem); a félév során el kell végezni a prediktív becslést a tesztállományra és ki kell értékelni az eredményt.

A megoldáshoz bármilyen segédszoftver használható; cél a tesztadatsor célváltozójának minél jobb predikciója (az 1. szakaszban 1 beadási lehetőség van, a 2. szakaszban pedig plusz egy beadási lehetőség) és minél több saját kontribúció elérése. A házi feladat elfogadásához szükséges tennivalók és határidejük:

- Február 28-ig el kell készíteni egy dokumentumot (1 - 3 oldal) a feladat rövid értelmezésével és megoldási ötletekkel, tervvel (szucs@tmit.bme.hu címre).
- Április 20-án: A megoldásnak azt a részét, ami már készen van, be kell mutatni szóban (Teamsen) egy kiselőadás keretében. Ha van már egy kezdetleges, egyszerű megoldással előállított predikció a célváltozóhoz, akkor azt kérem elküldeni e-mail-en április 18-ig (1. szakasz).
- Május 7-ig lehetőség van a tesztadatsor célváltozójának még egy predikciójára (2. szakasz). Ezen kívül május 7-ig kell elkészíteni a megoldást bemutató dokumentumot.
- Május 11-én kiselőadást kell tartani a végső megoldásról, valamint meg kell válaszolni a felmerülő kérdéseket.

2020/2021. 2. félév

Megoldási tervezet

Feladat rövid értelmezése

A feladat során videók tartalmát kell elemezni mesterséges intelligencia algoritmusokkal. Osztályozni kell a videókon látható vizualitást a megjelenő érzelmek tekintetében. A feladat elvégzéséhez rendelkezésre áll egy tanuló és egy teszt videóállomány, melyek az alábbi linken érhetők el:

https://drive.google.com/drive/folders/1kYYWZiPBjaxX99wI75kDbirgS_YFdioY?usp=sharing

A teszt adatállomány 270 videót tartalmaz. A tanuló adathalmazban 6 érzelem van elkülönítve: düh, undor, félelem, boldogság, szomorúság, meglepetés. Nagyjából 70 videóval tudunk számolni érzelmekként külön-külön.

A tanuló adathalmazon mesterséges intelligencia modell felépítése a feladat, amely képes a videókat feldolgozni és hozzájuk érzelmet társítani. A modell használhatóságát a teszt adathalmazra adott predikcióval tudjuk bemutatni. Ezt a predikciót kell beadni, melynek egyszeri javítására is lehetőség van.

A munka követésére két szóbeli és egy írásbeli beszámoló szolgál a félév során.

Megoldási ötletek

A feladat elvégzéséhez valamilyen mesterséges intelligencia alapú megoldásra gondolhatunk ugyanis tanító adatok alapján kell osztályozni ismeretlen videókat. Megoldási ötletként a konvolúciós neurális hálózatokat tudnám mondani, mint felügyelt gépi tanulási technikát, mert a mély neurális hálók már számtalanszor bizonyították képosztályozási rátermettségüket a tématerületen. De, ha a multimédia tartalomból sikeresen előállítottunk számadatokat a megfelelő struktúrában, akkor más predikciós osztályozási algoritmusok is szóba jöhetnek, mint például döntési fákon alapuló modellek.

Terv

1. Irodalom kutatás a témában, különböző megoldások vizsgálata, kipróbálása
2. Órán tanult módszerek áttanulmányozása
3. Videók letöltése, szoftveres beolvasása, kezelése
4. Videók képekre bontása, felvágása
5. Képekről jellemzők kinyerése, tulajdonságvektorok megállapítása
6. Neurális hálózat és/vagy más tanuló megoldás elkészítése
7. Tanítás, hálózat javítása
8. Predikció elkészítése
9. Az elkészült megoldás javítása, továbbfejlesztése
10. Bemutató elkészítése és előadása

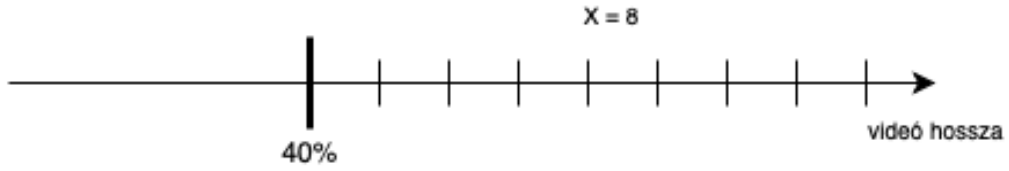
Megvalósítási dokumentáció

A feladat megoldása során kétféle módszert próbáltam ki. Elsőként mimikai jellemzővektorok kinyerésén alapuló predikciót készítettem, majd konvolúciós neurális hálózat alkalmazásával is vizsgáltam a teszt adathalmaz videóit. Mindkét megközelítés alapja egy videó feldolgozó program, ami a bemeneti videóból képeket állít elő a megfelelő attribútumok mentén.

Videók feldarabolása

A videók feldarabolása során arra törekedtem, hogy a tanuló algoritmusok számára minél könnyebben emészthető képek szülessenek. Ezért a képek kinyerése közben az algoritmus megkereste a képkockákon látható arcokat. Szükség esetén újra méretezte őket. Majd egy külön mappában helyezte el szürkeárnyalatos verziójukat. A videó és kép manipulációjához a OpenCV[1] könyvtárat használtam, amely rendelkezik arcdetektáláshoz szükséges függvényekkel is. A program inputja egy képkocka szám, ami meghatározza, hogy egy adott videóból hány frame kerül lementésre. A kivágott részek között egyenlő idő telik el, amely mértéke dinamikusan videonkként külön-külön kiszámításra kerül.

Vizsgálva a tanuló és tesztadathalmazt jól látható, hogy a videók elején az érzelmet kifejező ember még csak rákészül a mondandójára, az arcokon nem tükröződik a felismerendő érzelem. Ezért úgy alkottam meg a feldaraboló algoritmust, hogy a videó tartalom első 40%-át ne vegye figyelembe az időközök megállapításánál és a képek készítésénél. Tehát a mintavételezés működési mechanizmusa az alábbi ábrának felel meg (1. ábra):



1. ábra: videó mintavételezése

Látható, hogy az algoritmus a videó első 40%-ából nem mintavételez semmilyen képet. Majd a maradék 60%-ot 9 egyenlő részre osztja és 8 képkocka kerül lementésre (arckeresés és átalakítás után).

Ezután a két megoldás kettévált kódszervezési és működési szempontból is.

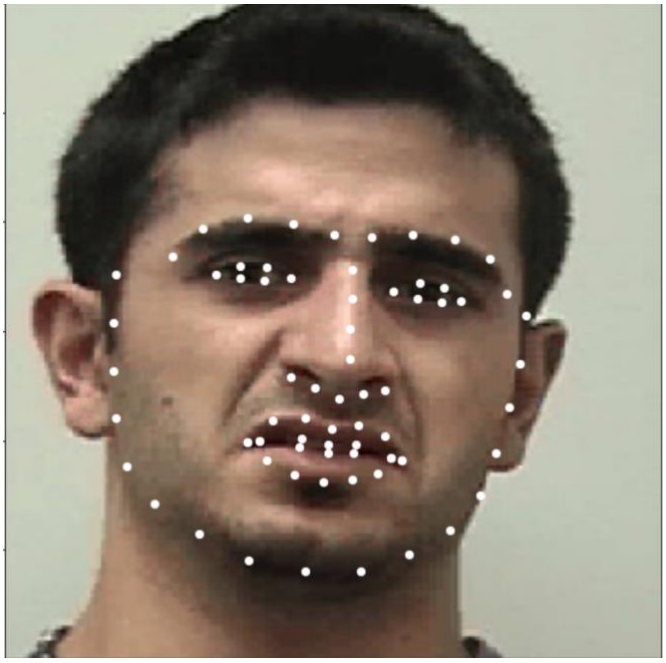
Mimikai jellemzők kinyerése

Az első megközelítéssel az volt a célom, hogy a videókon megjelenő arcváltozásokat, mimikát számszerűsíteni majd predikálni tudjam. A mimikai leírók alapján egy felügyelt tanuló algoritmussal modellt illesszek az adathalmazra és új videók esetén érzelmet tudjak társítani az arcváltozásokhoz. Ehhez nem elég csupán az arcok befoglaló négyzeteit megkeresni. Az arcjellemző pontok lokalizációjára is és változásuk nyomon követésére is szükség van.

A megoldására egy python algoritmust írtam, amely Pandas DataFrame-be gyűjti az arcjellemző koordináták változását. Az arcjellemzőket a *face_recognition*[2] nevű könyvtárral kerestem meg 300x300-as színes arcképeken. A *face_recognition* algoritmusa az alábbi ábrán látható 72 pontot tudja detektálni (2. ábra). Tehát ha például a videóból 12 képkockát vágunk ki, melyeken 72 pontot detektálunk külön-külön és vesszük az x és y

[1] https://pypi.org/project/opencv-python/
[2] https://pypi.org/project/face-recognition/

koordinátánkénti változásukat (11*72*2=) 1580 jellemzőt kapunk egy videóhoz. Ez a tanuló adathalmaz méretéhez képest (~480 videó) is hatalmas jellemzőtérnek mondható. Ebből jól látszik a mimikai jellemzővektoros megközelítés legnagyobb kihívása. A hatalmas jellemzőtér problematikát a kivágott képek számának minimalizálásával és a detektált pontok szűrésével szerettem volna csökkenteni.



2. ábra detektálható arcpontok

Felügyelt tanulás a mimikai jellemzőkön

Miután elkészült a strukturált adathalmaz, melynek sorai egy-egy videónak felelnek meg, oszlopai pedig egy arcjellemző koordináta változásának, következhet az adatelőkészítés. Előfordultak olyan képkockák a videókban, amikor a színészi játék megmozdulásainak köszönhetően nem lehetett arcot detektálni. Ilyenkor a jellemzőváltozásokat nullának tekintettem és a kapcsolódó mező értékét is erre állítottam. Majd az adatokat átskáláztam a 0-1 tartományba.

A tanítási fázis alatt több algoritmust is kipróbáltam: RandomForest, Gradient Bossting, XGBoost. Egy elkülönített validációs adathalmazon (a tanító adatok 20%-án) vizsgáltam az eredményességüket.

Legjobb mért megoldással a Gradient Boosting technika szolgált. A videókból 12 képkockát vágtam ki és mind a 72 arcpont mozgását figyelembe vettem, ugyanis így teljesített legjobban a tanuló algoritmus, amellyel a validációs halmazon 55,3%-ot értem el. A beadott eredményem a tesztadathalmazon 38,5% lett.

Konvolúciós neurális hálózat használata

A másik megoldási ötlet egy CNN használata volt szürkeárnyalatos arcképeken. Itt fontos megjegyezni, hogy a konvolúciós hálózatok általában egyszerűsített, kis méretre vágott képekkel dolgoznak, ezért a feldaraboló algoritmust úgy futtattam, hogy 48x48-as méretű, szürkeárnyalatos képeket mentsen ki a videókból. Videónként 32-t. Ilyen kisméretű képeken már egy arcdetektor nem tudna jó eredménnyel dolgozni, de a neurális hálózat bemeneteként ez a méret kezelhetőnek bizonyult.

Elsőként a már ismert VGG hálózattal próbálkoztam[3], de ez túl robusztusnak bizonyult

[3]https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-

a problémára. Nagyon lassan (több óra alatt) is csak nagyon kis accuracy-t (0.21) tudott elérni a tanító adathalmazon. Ezért lényeges redukálás mellett döntöttem. Hosszabb optimalizálás és számítás után az alábbi rétegek összeállításával dolgoztam tovább:

```
    model = Sequential()
    model.add(Conv2D(input_shape=(48,48,1), filters=32, kernel_size=(3,3), padding="valid",
activation="relu"))
    model.add(Conv2D(filters=32, kernel_size=3, padding="valid", activation="relu"))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2,2)))
    model.add(Dropout(0.3))

    model.add(Conv2D(filters=64, kernel_size=3, padding="same", activation="relu"))
    model.add(Conv2D(filters=64, kernel_size=3, padding="same", activation="relu"))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2,2)))
    model.add(Dropout(0.3))

    model.add(Conv2D(filters=128, kernel_size=3, padding="same", activation="relu"))
    model.add(Conv2D(filters=128, kernel_size=3, padding="same", activation="relu"))
    model.add(BatchNormalization())
    model.add(MaxPool2D(pool_size=(2,2)))
    model.add(Dropout(0.3))

    model.add(Flatten())
    model.add(Dense(512, activation="relu"))
    model.add(Dropout(0.3))

    model.add(Dense(256, activation="relu"))
    model.add(Dropout(0.3))

    model.add(Dense(6, activation="softmax"))
```

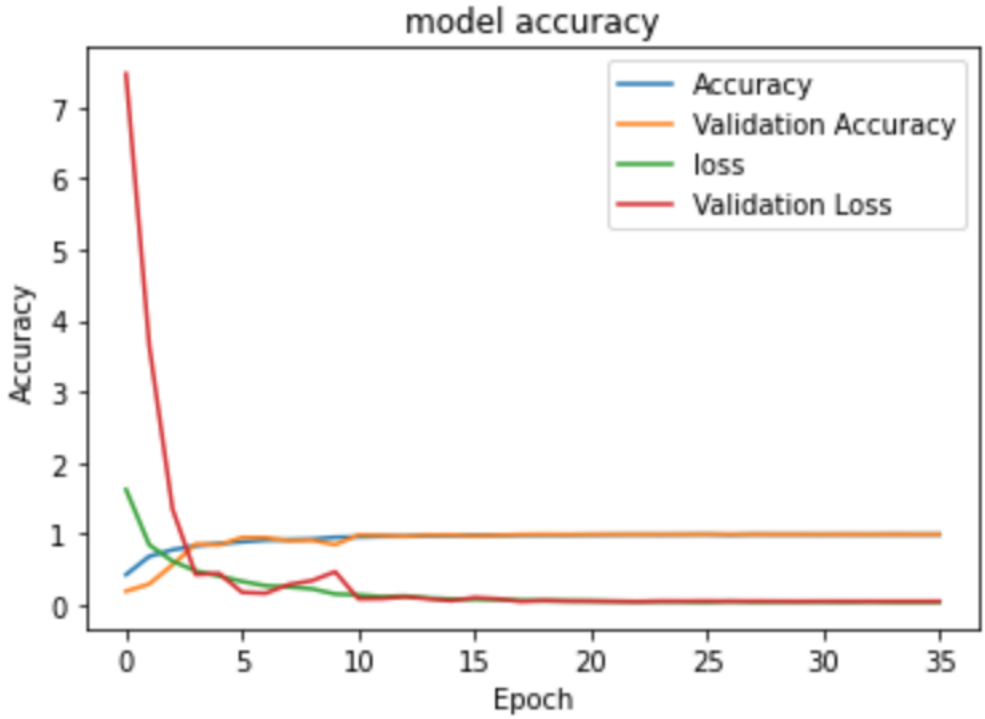
A VGG hálózat alapján érdemesnek tűnt két konvolúciós réteg használta egymás után. Az órán tanult eljárásokat is ki akartam próbálni (Dropout, Batch Normalization), melyeket Keras-ban plusz rétegek hozzáadásával lehet a modellbe építeni. A hálózat végére 3 fully-connected réteget helyeztem. A hálózat elején valid párnázást használtam, hogy a képek szélén található információ is számításba kerüljön. Későbbi rétegeknél igyekeztem csökkenteni a paraméterteret. A hálózat így végül közel 2,06 millió paraméterrel rendelkezett (a VGG 50 milliójához képest).

A rendelkezésre álló adathalmazt többféleképpen is szét lehet bontani tanuló és validációs halmazra. A „hagyományosabb” megoldásnak tekinthető esetben a *képeket* osztjuk szét két adathalmazra tetszőleges arányoknak megfelelően. Ebben az esetben, nagy eséllyel az összes videóból kerül egy-egy kép a nagyobbnak tekinthető tanuló adathalmazba. Így a hálózat tanulása során valamennyire arc sajátosságokat is párosítani fog az érzelmekhez, ami esetünkben nem túl szerencsés, hiszen a kifejezett érzelem nem függ a színész személyétől.

A másik módszer, mikor a *videókat* osztjuk szét a két adathalmaz között. Így lesz olyan videó (és hozzátartozó képhalmaz), amit a tanítás során nem lát a hálózat, csak visszaméréskor szembesül vele. Én mind a két megoldást kipróbáltam majd végül a videó alapú szétbontással dolgoztam tovább.

A tanítás során alkalmaztam Early Stoppingot és Reduce Learning Rate on Plateau technikákat is, hogy a túltanulást elkerüljem és a paraméteroptimalizálást, globális minimum keresését elősegítsem. A tanítás során a modell mérőszámai az alábbiak szerint alakultak (3. ábra):

a833c686ae6c



3. ábra mérőszámok alakulása a tanítás során

A tanító halmazt 20-80%-ban osztottam tovább validációs és tanító adathalmazra. Ezen halmazok mérőszámai láthatóak a külön egyeneseken. A modell nagyon gyorsan minimalizálni tudta a hibát a validációs és tanuló adathalmazon is egyaránt. A 20-ik epoch környékén eléri a 98,9%-os accuracy-t mindkét halmazon majd az early stopping állítja meg a tanulást 35-ik epoch után. Ezek alapján valószínűsíthető, hogy kisebb hálózat is elegendő lenne a probléma kezelésére, amely kevesebb paraméterrel rendelkezik, de később elér hasonló eredményeket.

A tanulás után végül elkészítettem a tesztadathalmaz predikcióit is. Minden videóhoz rendelkezésre állt 32 arckép a videódaraboló algoritmus segítségével, amelyekhez a feltanított hálózattal könnyen érzelmet tudtam társítani. A 32 predikált eredmény közül kiválasztottam a legtöbbször előfordulót. Kérdést azon esetek jelentettek, mikor a 32 eredmény közül nem volt egyértelműen kimagasló előfordulás. Például olyan videó, ahol a becslések felében félelmet társított az algoritmus egy képhez, a másik felében meg dühöt. Ezeket az eseteket közelebbről is megnéztem. 13 videó predikciója volt ilyen szempontból kérdéses.

Megoldást a problémára az jelentette, hogy egy újabb modellt építettem hasonló paraméterekkel, de a véletlenszerű választásnak köszönhetően, (részben) más adatokon tanulva. Így már két modell sorolta be a teszt videókat valamilyen érzelmi osztályba. A 64 predikciót összesítve egy fokkal könnyebb volt eldönteni, hogy milyen érzelem szerepelhet adott videón. A bizonytalannak mondható videók száma 12-re csökkent.

A maradék 12 videót szemügyre vettem. Bizonyos esetekben szemmel is nehezen volt megállapítható az érzelem és voltak, amiknél passzoltak a számított érzelmek. Végül az alábbi módosításokat végeztem el (összesen 8 videó predikciója módosult, 4 ábra):

Teszt videó	Predikált érzelem	Javított érzelem
test018	Surprise	Happiness
test033	Sadness	Surprise
test101	Sadness	Disgust

test142	Happiness	Disgust
test143	Happiness	Disgust
test144	Happiness	Surprise
test151	Surprise	Fear
test225	Anger	Fear

4 ábra: érzelmek javítása videókon

A fentiek elkészítésével adtam be a predikciómat, amely a teszt adathalmazon 0,644%-os accuracy-t ért el.