



DIPLOMATERVEZÉSI FELADAT

Benda Krisztián
mérnökinformatikus hallgató részére

Feliratok illesztése képekre és videókra mesterséges intelligencia használatával

A multimédia és filmes iparban a videók feliratozása, a feliratok elhelyezésének megfelelő kiválasztása fontos feladat, és már régóta igény mutatkozik ennek az automatizálására. Ha a videó tartalmához szeretnénk igazítani a megjelenítendő szöveget, akkor megfelelő feliratpozíció kiválasztása történhet fontos objektumok, például logók, feliratok/szövegek, emberek, illetve a videó egyéb elemeinek és szereplőinek felismerése, behatárolása, relatív helyzetének elemzése, követése révén. A hallgató feladata, hogy ez utóbbinak az automatizálására kidolgozzon és megvalósítson egy olyan mesterséges intelligencia algoritmusokat használó rendszert, mely elkerüli a buktatókat (pl. a felirat ne takarjon ki logót, más feliratot/szöveget, arcot, vagy fontosabb részt a képen) és lehetővé teszi, hogy a kidolgozott elhelyezési startegiát használva, megfelelő időzítéssel játszunk le feliratozott videókat.

A hallgató feladata terjedjen ki az alábbiakra:

- A multimédia feldolgozás szakirodalmán belül mutassa be a feliratozás fontosságát, az automatizálásra használható módszereket és feladat kihívásait!
- Válasszon ki megfelelő képeket és videókat, melyek alkalmasak a feliratelhelyezés változás megfigyelésére. Szükség esetén címkezz fel őket, hogy mesterséges intelligencia használatával is feldolgozhatók legyenek!
- Azonosítson olyan jellemzőket, amelyeket érdemes figyelembe venni a feliratelhelyezés során! Térképezze fel, hogy kiválasztott jellemzők milyen algoritmusok és eszközök segítségével nyerhetők ki a rendelkezésre álló adatfolyamokból!
- Tervezzen meg egy rendszert, mely automatikusan felirat elhelyezésre képes a videók képeire! Gyűjtse össze ehhez a különböző algoritmusokat, melyek eltérő szempontok szerint helyezik el a feliratot az egyes képeken a képek tartalmának figyelembenél, majd adjon saját, mesterséges intelligencia alapú megoldási módszert a meglevő algoritmusok felhasználásával!
- A feliratok elhelyezését két lépcsőben valósítsa meg! Az első lépcsőben csak különálló képeken oldja meg az illesztést úgy, hogy ehhez implementálja a választott algoritmusokat! Második lépcsőben pedig az egymás utáni képeken (azaz a videón) valósítsa meg az illesztést!
- Tárja fel, hogy minden más szempontok szerint érdemes feliratokat elhelyezni, megbontani (például párbeszéd feliratozása esetén) és adjon megoldási módszert rájuk!
- Definiáljon néhány jósgági tényezőt, amik alapján a megalkotott feliratelhelyező rendszer kiértékelhető és végezze is el az értékelést!
- Foglalja össze a munkáját, adjon a rendszer továbbfejlesztésére javaslatokat!

Tanszéki konzulens: Dr. Szűcs Gábor

Külső konzulens: Rechner István, IBM Watson Media

Budapest, 2019. szeptember 30.

/ Dr. Magyar Gábor /
tanszékvezető



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Távközlési és média-informatikai tanszék

Benda Krisztián

**FELIRATOK ILLESZTÉSE KÉPEKRE ÉS
VIDEÓKRA MESTERSÉGES
INTELLIGENCIA HASZNÁLATÁVAL**

KONZULENS

Dr. Szűcs Gábor

BUDAPEST, 2019

Tartalomjegyzék

Összefoglaló	5
Abstract.....	6
1 Bevezetés	7
1.1 A témakör ismertetése	8
1.1.1 Felirattípusok	9
1.1.2 Feliratozás folyamata	10
1.2 A feladat részletes bemutatása	11
2 Irodalomkutatás.....	13
2.1 Párbeszéd alapú feliratelhelyezés	13
3 Tervezés	18
3.1 Képtulajdonságok	18
3.2 Rendszerarchitektúra	20
4 Megvalósítás	22
4.1 Éldetekció	22
4.2 Objektumdetektálás	23
4.3 Karakterlokalizáció	25
4.4 Logó és márkaeljelzés detekció	27
4.5 Algoritmusok felhasználása	29
5 Lezáráás	32
5.1 Továbbfejlesztési javaslatok	32
5.2 Összefoglalás	32
Irodalomjegyzék.....	33
6 Függelék	35

HALLGATÓI NYILATKOZAT

Alulírott **Benda Krisztián**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettettem, egyértelműen, a forrás megadásával megjelöltetem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2019. 12. 11.

.....
Benda Krisztián

Összefoglaló

Az egyre növekvő igény az internetes tartalomfogyasztás iránt a videó készítési és feldolgozási folyamatokat is felgyorsítja. A videók mellett egyre nagyobb hangsúly kapnak a feliratok generálása és használata. Manapság már nem csak a süketek és nagyothallók számára készülnek szövegezési megoldások, hanem az átlagember számára is, hogy megkönnyítse a tartalomfogyasztást zajos környezetben.

A feliratok pozíciója számos esetben zavaró helyre esik, amely gátolja az alatta lévő képi információ átadását. Sokszor letakar más a képen megjelenő szöveget, vagy fontos vizuális tartalmat. Dolgozatomban bemutatok egy okos feliratpozicionáló megoldást, melynek használatával elkerülhetjük a zavaró feliratpozíciókat.

Először a téma kört és a videó feliratozási problematikát ismertetem, majd a kapcsolódó tudományos cikkeket elemzem részletesen. A detektálható képtulajdonságok alapján rendszerarchitektúrát tervezek, amely képes elvégezni az okos pozícionálást képeken. Végül az implementációs részletek és eredmény bemutatása után továbbfejlesztési javaslatokat teszek, hogy munkám széleskörben felhasználható legyen.

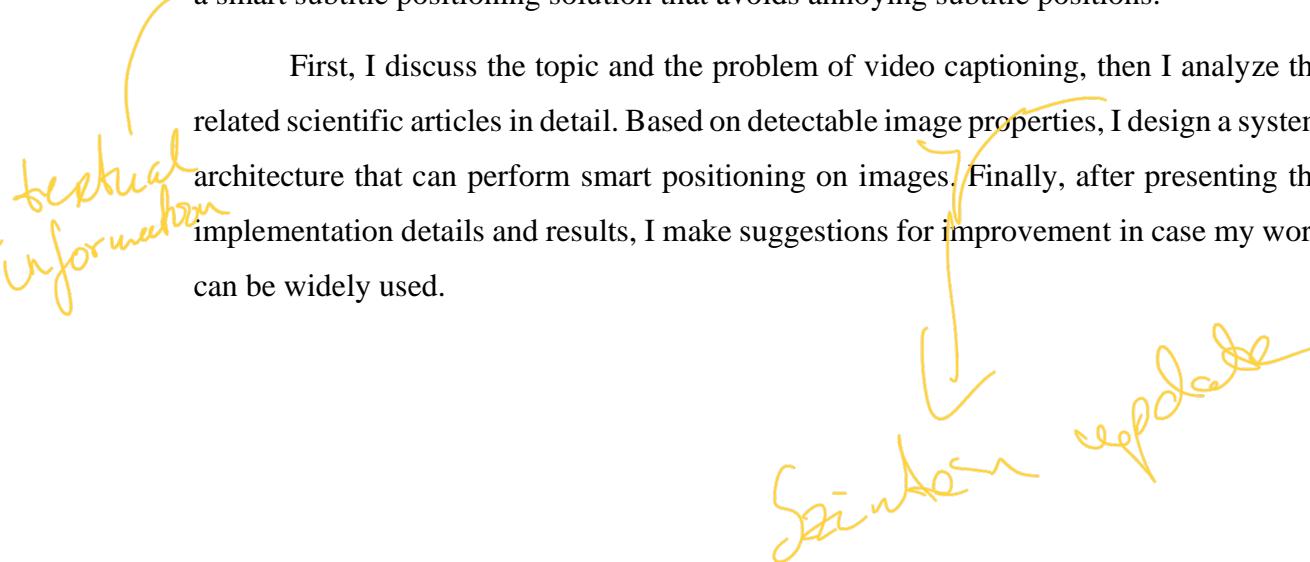
Itt majd be kell rakni a
további haladást

Abstract

The growing need for Internet content consumption also accelerates video production and processing. Besides the increasing number of videos and films, generating and using captions are also got emphasis. Nowadays, not only deaf and hard of hearing people aimed by captioning solutions. The average people are also targeted because facilitating content consumption in a noisy environment and better transcription are needed as well.

In many cases, the position of the subtitles and captions are in a confusing area, which prevents the transmission of the underlying visual information. Many times, it obscures other text or important visual content in the image. In my thesis work, I present a smart subtitle positioning solution that avoids annoying subtitle positions.

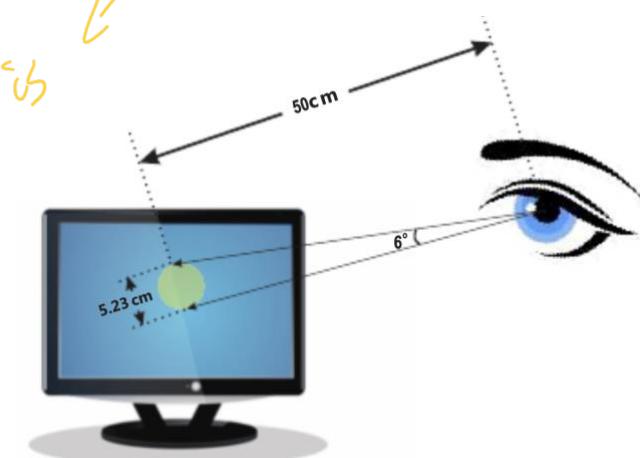
First, I discuss the topic and the problem of video captioning, then I analyze the related scientific articles in detail. Based on detectable image properties, I design a system architecture that can perform smart positioning on images. Finally, after presenting the implementation details and results, I make suggestions for improvement in case my work can be widely used.



1 Bevezetés

Napjainkban a multimédia tartalmak egyre szélesedő körben használatosak. Az emberek már nem csak televízió, számítógép előtt néznek filmeket, videókat, hanem sokszor tömegközlekedve kisebb-nagyobb képernyőkön is fogyasztják a vizuális tartalmat. Ahogy nő az igény a videók iránt, úgy kap egyre nagyobb szerepet a videóhoz tartozó felirat is. Feliratok videóra helyezése különböző célokat szolgálhat. Idegen nyelvű videóknál, hallássérülteknel, audió lejátszására nem alkalmas, zajos környezetben (például tömegközlekedés közben, utcai kivetítők esetén) tagadhatatlan a felirat szükségszerűsége.

Legtöbb esetben a videóhoz tartozó felirat egy fix pozícióba van égetve a képen. Általában ez a videó alján, középre igazítva jelenik meg, amely sokszor optimális megoldás lehet, de sok „felesleges” szemmozgásra kényszeríti a nézőt. Az emberi szem fókuszpontja, amely az olvasást teszi lehetővé, nagyon keskeny. Egy fókuszpont használatával egy-egy szót tudunk csak értelmezni (ez természetesen emberenként eltérhet). Fél méterről 5,23 cm átmérőjű körnek felel meg (1.1 ábra). Ezért a néző mozgalmas, sok párbeszédet tartalmazó részeknél gyakorlatilag nem is tudja szemmel követni a cselekményt, ha a feliratot követi tekintetével, ami rontja a tartalom elvezethetőségét és érthetőségét. Továbbá a gyakori fókuszváltás a felirat és a vizuális cselekmény között, a szem megerőltetésével jár együtt.



1.1 ábra: Az emberi szem korlátott látótávolsága. Fél méterről 5,23 cm átmérőjű kör [1]

A felesleges szemmozgásokon kívül más aspektusban is fontos a felirat pozíciója. Vannak olyan tartalmak, például híradó, sportközvetítések mikor a beszélő nem is

szerepel a képen, csak a lényegi tartalom látszik, de számos más információ van közölve a képernyő szélén. Erre mutat példát az alábbi ábra (1.2. ábra). Jól látható, hogy az autóversenyzés közben fontos adatok jelennek meg a képernyőn az egyes versenyzőkről és az autók aktuális sorrendjéről. A háttérben a kommentátor folyamatos plusz információval gazdagítja a versenyt ezért, ha bekapcsoljuk a feliratot a videóhoz, gyakorlatilag elveszítjük a képernyőn megjelenő információk egy jelentős részét. Egyértelműen látszik, hogy a kép felső része alkalmasabb lenne a felirat elhelyezésére, mert ott csak az elmosódott környezet látszódik.



1.2. ábra: Autóversenyzés a kommentátor szövegének feliratával

Már csak a fentebb említett két probléma kapcsán is érdemes elgondolkodnunk a felirat pozíciójának optimalizálásán, dinamikus igazításán. Mivel figyelmemet felkelte az ismertetett problematika, ezért diplomamunkámmal mélyebben is elkezdtem foglalkozni a feladattal és megoldási lehetőségeivel.

1.1 A téma kör ismertetése

A videó és film feliratozás problémakör nagyon sokrétű, melyet több részfeladatra tudunk bontani. A legfontosabb lényegi elem egyértelműen a *felirat szövegi tartalma*. Ezután jön a *szövegezés pontos időzítése* és igazítása a hangsávhoz, majd *pozicionálás* a videón megfelelő vizuális megjelenéssel. Ezek közül diplomamunkám a megfelelő pozicionálással foglalkozik széleskörűbben, de különböző részletességgel a másik két

feladatkörre is kitérek. Viszont a részfeladatok előtt érdemes megismerni a különböző felirattípusokat.

1.1.1 Felirattípusok

A magyar nyelv kevésbé tesz különbséget az angol *caption* és *subtitle* szavak jelentése között, hisz mindenkor a felirat szót használjuk, holott a két szó jelentése eltérő idegen nyelven [2]. A *caption* szó általában süket és nagyothallók számára készített, a videó nyelvével azonos feliratot takar. A *subtitle* pedig a feliratba ágyazott fordítást teszi lehetővé. Tehát a *subtitle* elsősorban nem hallássérülteknek készül, ezért sokszor nem találhatók meg benne hangutánzó szavak, vagy nem emberi hangok feltüntetései. A *caption* pedig a videó eredeti nyelvén érhető el és hangleíró szavakat és kifejezéseket minőségtől függően tartalmazza. Mivel én elsősorban a felirat elhelyezésével foglalkozom, ezért az én szempontomból neutrális, hogy *caption*-nel, vagy *subtitle*-lel kell dolgoznom, ezért a két kifejezést dolgozatomban én is szinonimaként használom (akkárcsak a magyar nyelv).

Az angol szakirodalom megkülönböztet két másik felirat típust is. Az *open* és a *closed captioning*-et. *Open captioning (OC)* esetén a felirat kikapcsolására nincs lehetőség, az a videóba égetve jelenik meg a kép részeként. A *closed captioning (CC)* pedig opcionális és a néző által aktivált feliratot takar [2] [3]. Mind a két típusnak vannak előnyei és hátrányai, melyek munkám szempontjából is igen fontosak, ezért ezeket egy táblázatban foglaltam össze (1.1. táblázat).

	Open Captioning	Closed Captioning
<i>Használat, aktiválás</i>	A hozzá nem értő felhasználónak nem kell külön bekapsolnia, de kikapsolni sem lehet.	Ki-be kapcsolható, de minimális hozzáértés szükséges hozzá, mely televízióként és lejátszónként változhat.
<i>Időzítés</i>	Mindig együtt mozog a hanggal, ezért nem fordul elő aszinkronitás.	Gyakran elcsúszik a videó hangjától, de a sebesség lejátszás közben is állítható.
<i>Pozíció</i>	Készítésnél a videón belül bárhol mozgatható, ezért	Alaphelyzetben (közép alul) zavaró lehet. Pozíciója csak minimálisan testreszabható,

	beállítható úgy, hogy ne takarjon ki lényeges tartalmat.	melyet a médialejátszónak kell támogatnia.
Tárolás	Nem kell külön tárolni, mozgatni.	Külön tárolás és fájlformátum szükséges.
Cserélhetőség	Nem cserélhető más nyelvre, de mellette alkalmazható close caption. Viszont ez nagyban rontja a felhasználói élményt.	Könnyen leváltható egy másik nyelvű verzióra.
Fájlformátum, dekódolás	Nem szükséges külön dekóder hozzá <i>Pont Yell a mondat nagyre</i>	Különböző fájlformátumok: <i>.srt, .scc, .ttml, .dxfp, .vtt, .cap, .ass,</i> melyekhez különböző dekóder szükséges és a lejátszók különbözőket támogatnak.

1.1. táblázat: Open és Closed Captioning összehasonlítása

Ahogy látható (nevével ellentmondásos módon) a *Closed Captioning* egy sokkal kötetlenebb feliratozási módot tesz lehetővé, ahol a szöveg szabadon módosítható, ki-be kapcsolható. Az én szempontomból viszont egy nagyon jelentős hátrány, hogy a legnépszerűbb felirat fájlformátumok nem támogatják a feliratszegmens pozícionálását. Egy feliratszegmensnek azt a felirat szövegrészét nevezzük, amely egy megadott intervallumon belül egyszerre jelenik meg a videón. Ez lehet egy mondat, vagy rövidebb kérdés-válasz, de akár egy hosszabb félmondat is.

A médialejátszókban általában csak egy konstans pozíciót lehet megadni a felirat helyének, mely nem variálható. Tehát ha én feliratszegmens specifikusan szeretném változtatni a pozíciót, akkor vagy egy nagyon egyedi fájlformátumot kell alkalmaznom (*.ass*), vagy *Open Captioning* feliratozási technikát használnom. Mivel az *.ass* formátumnak nagyon kicsi a támogatottsága, ezért az *Open Captioning* megoldást részesítettem előnyben.

1.1.2 Feliratozás folyamata

A *felirat szövegi tartalma* különböző forrásból származhat. Napjainkban is gyakori, hogy a feliratozást dedikált ember végzi egy-egy élő videó közvetítése közben,

vagy egy önkéntes jelölt lefordít filmeket feliratozott formában. Szerencsére már léteznek hatékony megoldások az automatikus generálására is. Ilyenkor gyakorlatilag egy *speech-to-text* (élő beszéd szöveggé konvertálása) problematikáról van szó, melyeket mesterséges intelligencia algoritmusokkal már egész jól meg tudunk oldani. Rendelkezik ilyen szolgáltatással a Google [4], az Amazon [5], az IBM Watson [6], de ha szeretnénk találhatunk nyílt forráskodú lehetőségeket is [7].

A *felirat időzítés* problematikára legkézenfekvőbb megoldás lenne az *Open Captioning* használata, de ez jellegéből adódóan sok kompromisszumot rejt magában. *Closed Captioning* esetén a feliratfájlformátumok, a médialejátszók sokfélesége, illetve a különböző videó kiterjesztések miatt a vetítés közbeni szöveg teljesen aszinkron módon tud mozogni a hanggal. Egyes lejátszók (pl.: VLC [8]) képesek a lejátszás közben is időben tolni a feliratozást, amely nagy könnyebbéget jelent, de sokszor nem oldja meg teljesen a problémát. Találhatunk olyan módszereket is, amelyek a mesterséges intelligencia előnyeit használják ki és egy adathalmaz alapján hálózatot tanítanak fel a problematikára [9].

A harmadik megoldandó feladatkör pedig a *pozícionálás*, amit fő csapásiránynak jelöltetem ki diplomamunkám során, ahogy már korábban is írtam. Legelterjedtebb megoldásként a videó alján, középen szokták megjeleníteni a feliratot, de ez sokszor fontos információveszteséggel jár. Ahogy az 1.2. ábra is látható. Megoldásként egy olyan algoritmust dolgozok ki, amely képes képeket és videókat elemezni tartalmuk alapján és a feliratot olyan helyre pozícionálja, ahol semmilyen lényeges információt, logót, szöveget, objektumot nem takar ki, amennyiben ez lehetséges.

1.2 A feladat részletes bemutatása

Munkámmal szeretnék egy olyan megoldást adni feliratok elhelyezésére képeken és videókon, mely az eddigiekől eltérő módon, automatikusan elemezi a vizuális tartalmat emberi interakció nélkül és több szempont alapján megvizsgálva pozícionálja a szöveget. Jelen fejezetben a diplomatervezési feladatomat taglalom részletesebben és ismertetem végrehajtásának lépéseit.

A témakör ismertetésénél (1.1 fejezet) már bemutattam a feliratozás fontosságát, körülményeit és a különböző lehetőségeket. A továbbiakban először az irodalomkutatási rész olvasható, melyet a feliratpozícionálás témakörében végeztem. Megkerestem a releváns cikkeket, kutatásokat és elemzem őket a diplomamunkámra vonatkozóan. Itt

kitérek a létező automatizálási módszerekre és optimalizálásra szoruló részletekre. Az ismertetett források tartalmaznak olyan pozícionáló eljárásokat, amelyek a párbeszédek könnyebb értelmezése céljából születtek meg és az aktuális beszélőhöz mérten helyezik el a feliratot.

Ahhoz, hogy egy dinamikusan változó videón feliratokat tudunk pozícionálni, a vetített képeket kell megvizsgálni. Ezért a munkám első szakaszában csak a lejátszás közben megjelenő képekkel foglalkozom, majd később a második szakaszban térek át videókra. Ennek következtében jelen dolgozat elsősorban képeken történő feliratelhelyezést mutatja be, de sok részlet videós környezetben is megállja a helyét (pl.: forráselemzés, elméleti háttér).

A munka elvégzéséhez megfelelő vizuális tartalmat keresek, melyeknél egyértelműen megfigyelhető a felirat zavaró elhelyezése (egy ilyen már a 1.2. ábra is látható). A beszámolómban külön-külön, a releváns részknél több ilyen példa is megtalálható, de a külön fejezetet, hosszabb bemutatást nem látom indokoltak.

A problémakört részletesebben megismerve elmondható, hogy tanító adathalmaz készítése képek felcímkézésével és hálózat építése kifejezetten pozícionálási célokra optimalizáltan nem lenne célszerű. A feliratok legjobb pozíciójának helyzete szubjektív és nagyban függ a vizuális tartalomtól, ezért nagyon sok manuális munkára lenne szükség és az eredményben ez nem tudna kellőképpen tükröződni, hisz mesterséges intelligencia alapokon más megközelítéssel precízebb és determinisztikusabb megoldást lehet készíteni. Ennek az elkészítését és háttérmunkálatait szeretném bemutatni diplomamunkámban,

Az irodalomkutatás után olyan jellemzőket, képtulajdonságokat azonosítok melyeket érdemes figyelembe venni felirat pozícionálásnál. A tulajdonságok automatikus felderítéséhez olyan algoritmusokat párosítok melyekkel könnyen kinyerhetők ezek az információk a képekből.

Egy összetett rendszer keretében hasznosítom az olvasottakat, ahol elsőkörben képekhez, majd később videóhoz tudunk automatikus feliratot illeszteni, pozícionálni. A megoldáshoz mesterséges intelligencia alapú algoritmusokat használok, amelyek lehetővé teszik a modern képelemzést, illetve egyéb képfeldolgozási eljárásokat is figyelembe veszek. Először a rendszerterv olvasható, majd később a megvalósítás. Legvégül összefoglalom munkámat és továbbfejlesztési javaslatokat teszek.

2 Irodalomkutatás

Videó feliratozás igen nagy téma körnek tekinthető a már ismertetett részfeladatok alapján (1.1.2 fejezet). A három feladat közül a Speech-To-Text problematika örvend a legnagyobb népszerűségnek, melyre már nagyon sok jól használható megoldás létezik. A feliratok automatikus pozicionálása azonban bőven nem rendelkezik ekkora szakirodalommal. Lokalizációra legelterjedtebb megoldás a különböző szoftverek által biztosított manuális beállításra/igazításra alkalmas környezet [10].

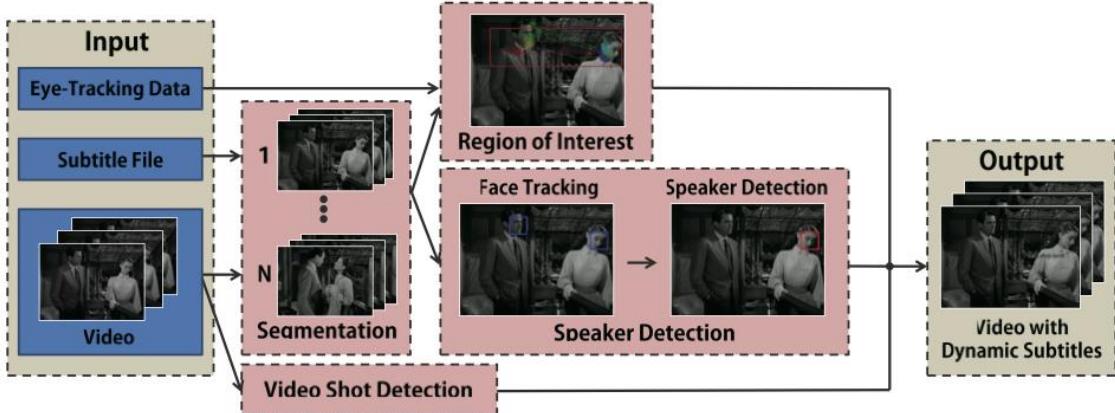
Az elérhető források a feliratpozicionálást elsősorban párbeszédközpontúan közelítik meg. Olyan megoldásokat kínálnak, amely a párbeszéd könnyebb követését tesszik lehetővé. Intelligens pozicionálásra, amely kifejezetten a felirat által kitakarandó területet analizálja és az optimumot keresi, nem találtam példát.

2.1 Párbeszéd alapú feliratelhelyezés

Japán egyetemek tudósai a videó feliratelhelyezés problematikáját különleges megközelítésben vizsgálták. A felirat helyzetének meghatározásához, nem csak a videón megjelenő vizuális elemeket használták, hanem a néző fókuszpontjait is figyelembe vették. A kutatás célja a felesleges szemmozgások minimalizálása volt, melynek eredményét és folyamatát egy tudományos cikkben publikálták [11].

A videón megjelenő képeken úgynevezett region of interest (ROI) halmazokat definiáltak, melyek a kép érdekes, fontosabb információit tartalmazzák egy téglalap alakú területbe zárva. A kutatásban kidolgozott eljárás a ROI terület alatt helyezi el közvetlenül a feliratot. Így a néző számára az érdekesebb részek nem kerülnek kitakarásra. Továbbá az algoritmus hang és vizuális elemek alapján az aktuális beszélő személyt is behatárolja a képen és a feliratot a beszélőhöz közelebb helyezi el.

Tehát a megvalósított eljárás inputként egy videót, egy felirat fájlt (.srt), és egy, a rögzített szemmozgásokat leíró fájlt kap. A kimenet pedig egy videó, amin megtalálható a dinamikusan változó felirat. Az eljárás sematikus felépítése látható a 2.1. ábra. A videó szegmensekre bontása mellett, a jelenetek detektálása is megtörténik Apostolidis és Mezaris technikájával [12]. Ezek után a figyelmi területek (ROI) kiszámításra kerülnek és az aktív beszélő azonosítása is lezajlik. Végül a ROI és a beszélő helyzetének felhasználásával a felirat beégetésre kerül a videóba.



2.1. ábra: Sematikus ábra a szemmozgás alapján történő feliratpozícionálásnak [11]

A ROI kiszámítása az én munkám szempontjából is fontos kérdés. Emberi szemmozgás felvétele túlságosan hosszú és költséges feladat, illetve fenntarthatósági szempontból sem előnyös megközelítés. A tanulmányban ismertetett módszer előnye a pontos statisztikai adat, amely ténylegesen a néző szempontjából fontosnak tartott figyelmi területeket írja le az egyes képkockákon (ROI). A gépi algoritmusok ennél sokkal irányítottabban, például objektumdetektálással tudnak csak dolgozni.

6 angol nyelvű 2 perces videót választottak ki tesztanyagként, melyek 2 film (“*Roman Holiday*” és “*Charade*”) néhány jelenetét tartalmazták. A tesztalanyok japán emberek voltak alapvető angoltudással. A vizsgálat alatt a tesztalanyok randomizált sorrendben nézték végig a videókat, miközben szemmozgásuk folyamatos megfigyelés alatt volt.

A tanulmányban feliratszegmensekként történt meg a ROI kiszámítása, így a szöveg szegmensek ugyanabban a pozícióban maradtak megjelenésük alatt, csak újabb feliratsor esetén történt pozíció változás. Más tanulmányokban volt arra is példa, hogy a mindenkor aktív beszélőhöz igazították a feliratot, ami azt eredményezte, hogy a feliratszegmensek együtt mozogtak a videón megjelenített történésekkel [13]. Ez a módszer kevésbé bizonyult effektívnak a túl sok szövegmozgásnak köszönhetően, ezért maradtak a feliratszegmens alapú ROI számításnál, de a későbbi kiértékelésnél ezzel a technikával is összehasonlították a megalkotott eljárást.

Az összegyűjtött szemmozgásokat aggregálták és egy téglalap alapú területtel kerítették körbe. Ezt tekintették a figyelmi területnek (ROI). A fókuszpontok 95%-a esett a területbe, így egy szűkebb intervallumot tudtak meghatározni, az átlagtól távolabb eső fókuszpontok kizárással.

A pontosság növelése érdekében a tanulmányban felhasználtak Speaker Detection (aktív beszélő meghatározás) és Face Detection (arc detektálás) algoritmusokat is, melyeket később figyelembe vették a felirat pozícionálásánál.

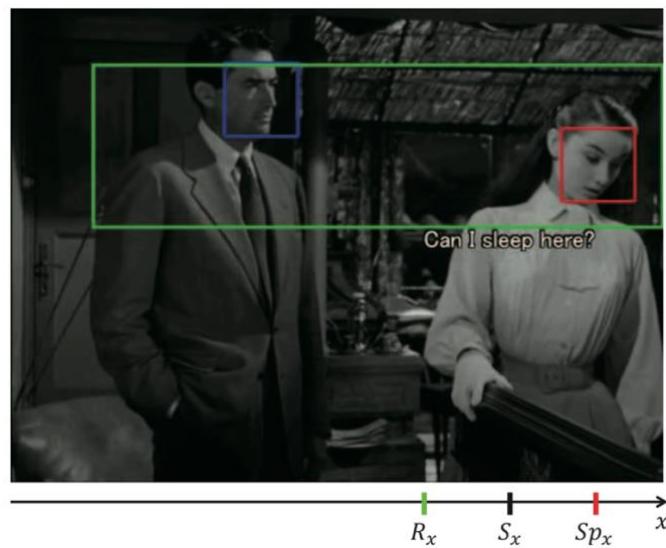
A szöveg elhelyezésnél az alábbi három szabály alapján dolgoztak fontossági sorrendben.

1. A felirat pozíciója alapján a beszélő könnyebben behatárolható legyen, azaz minél közelebb essen a hangforrásához.
2. A felirat ne takarja ki a legrégebbi érdekesebb részeket, ROI területeket.
3. Az egymásutáni feliratszegmensek pozíciója minimálisan változzon.

A feliratot minden esetben a ROI terület alá helyezték, ezzel elkerülve a szöveg jobb, illetve bal szélre (ki)szorulását. Mivel rendelkezésre álltak a ROI és a Face Detection-ből származó adatok, ezért a felirat közepének x-koordinátáját (S_x) az alábbi képlet alapján határozták meg.:

$$S_x = \begin{cases} \frac{(Sp_x + R_x)}{2}, & \text{ha az aktív beszélő meghatározható} \\ R_x, & \text{ha az aktív beszélő nem meghatározható} \end{cases}$$

Tehát amennyiben a beszélő arc helyzete is ismert, akkor a felirat az arc (Sp_x) és a ROI (R_x) középpontja között helyezkedik el félúton. Ellenkező esetben a ROI közepénél. Erre mutat példát az alábbi ábra (2.2. ábra):



2.2. ábra: Felirat a ROI középpont (zöld téglalap középpontja) és a beszélő arca (piros) között [11]

Amennyiben rövid idő (0.3 másodperc) telt el két felirat szegmens változása között, a felirat y koordinátája változatlan maradt, hogy a nézőt ne zavarja meg a túl sok pozíció változás. Ha jelenetváltás is tartozott egy darab szövegszegmenshez, akkor a megszokott alul-közép pozícióban maradt.

Az eredmények kiértékeléséhez 19 önkéntest kértek fel, alapvető angoltudással. A videókat 3 féle feliratozási eljárással is lejátszották nekik. Tradicionális, statikus feliratozással, beszélőt közvetlenül követő feliratozással és a tanulmányban kidolgozott összetettebb technikával. Az eredmények kimutatták, hogy szignifikánsan sikerült csökkenteni a szemfixációs időt az új eljárással. Illetve a tesztalanyok 6 videóból 4 esetében egyértelműen kényelmesebbnek és használhatóbbnak ítélték meg a kidolgozott eljárást, míg a maradék két esetben nem éreztek nagyobb javulást az eddig használt technikákhoz képest.

A forrást [11] nagyon fontosnak tartom az én munkám szempontjából, ugyanis a tanulmány íróinak sikerült olyan új feliratozási eljárást kidolgozni, ami emberi mérések alapján javította a felhasználói élményt amellett, hogy a videók lényeges elemei sem kerültek kitakarásra. A technika igen nagy hátránya, hogy használatához először fel kell mérni a nézők fókuszpontjait az egyes képkockákon, ami gyakorlatilag használhatatlanná teszi a tömeges alkalmazást. Saját munkámmal megoldást szeretném nyújtani erre a problémára. Célom, hogy ne kelljen emberi erőforrást “pazarolni” arra, hogy a képkockák lényegesebb részeit meghatározzuk.

Érdekes, hogy egy három évvel előbb, 2014-ben kiadott cikk, a *Speaker-following Video Subtitles* ugyanúgy a beszélő közelében elhelyezett felirat problematikát taglalja [1] azonban ekkor még nem foglalkoztak a ROI meghatározásával, így hogyha egy beszélő közvetlen közelében más fontos tartalom van, azt könnyen kitakarja a felirat. A tanulmány nagy hangsúlyt helyez a *Speaker Detection*-re, amely az én szempontomból is érdekes, bár munkám során nem a beszélő pontos beazonosítása a cél.

A kutatás szerint a legelterjedtebb módszer beszélő meghatározására a szájmozgást azonosító algoritmusok, de ezek sokszor nem elég pontosak. Ennek a javítására dolgoztak ki két módszert: a *center contribution*-t és *length consistency*-t. A *center contribution* esetén azzal a feltételezéssel számolnak, hogy az aktuális beszélő valószínűleg közelebb helyezkedik el a kép középpontjához, ezért kérdéses esetekben inkább a középperre helyezkedő beszélő az aktív. *Length consistency*-nél pedig azt vizsgálják, hogy az aktívnak jelölt személy mennyi ideig szerepel a képen a kapcsolódó

felirat szegmenshez képest. Minél nagyobb arányban található meg a képen annál valószínűbb, hogy tőle származik a hang. Ezenkívül *Audio-Visual Synchrony* [14] eljárást használtak a pontosabb eredmények eléréséhez, melynek lényege, hogy a képi és hangi mozgás ritmikája összehasonlítható és egy szinkronizációs számmal leírható. A detekciós algoritmusokat sorba kötötték és akkor fogadtak el valakit aktív beszélőnek, ha minden a négy algoritmusnak megfelelt.

A forrás a videókon történő feliratpozicionáláshoz adott számomra ötletet. A megoldásuknál a felirat szorosan az aktív beszélőhöz közel helyezkedett el, ami sok zavaró esethez vezetett. Például mikor a beszélő a kép egyik oldaláról a másikra mozog, a hozzá közeli szegmens kicsúszhat a képből, illetve a néző számára is zavaró folyamatosan mozgó feliratot olvasni. Így inkább rövidebb részekre bontották a szegmensemset és több részletben jelenítették meg.

A gyors jelentváltásoknál előfordulhat, hogy a szegmensek egymástól elég távol helyezkednének el egymás után, ami zavaró szemmozgást eredményez. Ezt pedig úgy oldották meg, hogy az új feliratot inkább az előző helyére tették, mintsem egy távolabbi helyre. Illetve az is azt a problémát is kezelték, mikor a beszélő a jelenetváltásoknak köszönhetően kikerül a képből. Ilyenkor inkább alaphelyzetbe (alul, középen) hagyták a szegmensemset.

Részletekbe merülések nélkül elmondható, hogy a 219 tesztalany, akikkel értékelték a megoldást, egyértelmű javulást érzékeltek a korábbi feliratozási eljárásokhoz képest. A cikkben leírt módszer a beszélő helyzetére optimalizál, ezért nagyon hasznos olyan környezetben, mikor sok párbeszéd található a videóban. Egy sportadásnál (1.2. ábra) ez inkább zavaró lenne, mintsem hasznos.

3 Tervezés

A munkálatokat két főbb részre osztottam. Először képeken történő feliratelhelyezésre, utána a videókra optimalizált megoldásra. A két feladat problematikájában is különbözik. Képeknél elsődlegesen az a feladat, hogy felismerjük a képen megjelenő fontosabb elemeket és megfelelő pozíciót keressünk a feliratszegmensnek. Videóknál pedig az összhangra is kell figyelni. Ahogy az egymás utáni képek jelennek meg a videón a felirat hajlamos lenne az „ugráslásra”. A különböző jelenetek és kameraállások miatt minden más a legjobb pozíció. Tehát ennél a résznél arra kell figyelni, hogy a különböző szegmenspozíciók ne okozzanak kellemetlenséget a felhasználó számára. Ezek alapján a két fő feladat a következő:

1. Képeken feliratszegmens okos elhelyezése
2. Videókon feliratszegmens pozíció optimalizálása

A két különálló feladatot a diplomamunka két részének megfelelő bontásban végzem el, így jelen tervezés is az első részfeladatra terjed ki.

Képek információhordozó ereje sokkal több az ember számára, mint amit az egyszerű számítógépek pixelek tárolásával kezelní tudnak. Ahhoz, hogy egy-egy kép vizuális tartalmát jobban megértsék a számítógépek, különböző detekciós eljárásokra van szükségünk. A detektálható tulajdonságok és feldolgozási lehetőségeik igen fontosak a megvalósítandó rendszer számára, ezért elsőként ezeket ismertetem.

3.1 Képtulajdonságok

Ha emberi szemmel ránézünk egy képre általában könnyen megtudjuk határozni, hogy mi a kép legfontosabb, legérdekesebb része még akkor is, ha az nem éppen a középpontban helyezkedik el. Olyan helyzetek is vannak, amikor detektálni tudjuk, hogy egy területet lefedésével értelmét vesztené a tartalom, annak ellenére, hogy nem a főbb objektum a képen. Ilyenek például a híradós adások, amikor egy konstans szövegdobozban a legfrissebb híreket tudjuk elolvasni és a vizuális tartalom teljesen másról szól (3.1 ábra). Sokszor a márkaelzésekkel, logókkal is ez a helyzet, mert kitakarásuk nem szolgálja a jelölt cég érdekeit. Ebben a fejezetben összegyűjtöttem a legfontosabb képi tulajdonságokat, amiket figyelembe kell venni felirat elhelyezésénél, a detektálásukat segítő megoldásokkal együtt.



3.1 ábra: Híradós adás, ahol a felirat alapértelmezett pozíciója már meglévő szövegi tartalomra esik

Képbe égetett szöveg. A fenti képen (3.1 ábra) jól látszódik, hogy a legrosszabb megoldás a feliratot alapértelmezett pozícióján hagyni. Nem csak az eredeti szöveget nem lehetne elolvasni, de a felirat értelmezésénél is igen zavaró hátteret eredményezne. Tehát a képeken/videókon található szöveget mindenkorban figyelembe kell venni a felirat elhelyezésénél és amennyiben van rá mód, nem ráhelyezni. Szöveget felismerni *OCR* (*Optical Character Recognition*) megoldásokkal [15] lehet, de esetünkben a szövegi tartalmuk kevésbé lényeges, mintsem pontos helyzetük. Ezért érdemes kevésbé robusztusabb technikákkal is próbálkozni.

Logó és márkajelzések. Számos cég sok pénzt költ arra, hogy márkája látszódjon a különböző vizuális tartalmakon (például reklámok, sportesemények szponzorai, videoklippekben felbukkanó termékek). Felirattal való elfedésük megtagadná céljukat. A problematikát *Logo Detection*-nek hívják és használható megoldásokkal is rendelkezik [16][17].

Különböző objektumok, arcok. Szintén fontos tartalomnak számítanak a képen található objektumok. Viszont a mai *Visual* vagy *Image Recognition* technikák [18] már annyira sok objektumot képesek detektálni, hogy egyes képeken már nem is tudunk semleges területet kijelölni, ahol könnyedén elférne a felirat. Az egyik legfontosabb vizuális tartalomként az arcokat emelném ki, mert ezek eltakarása nagyon zavaró tud

lenni a legtöbb helyzetben. Ezt a problémát igyekeznek megoldani a *Face Detection* technikák [19].

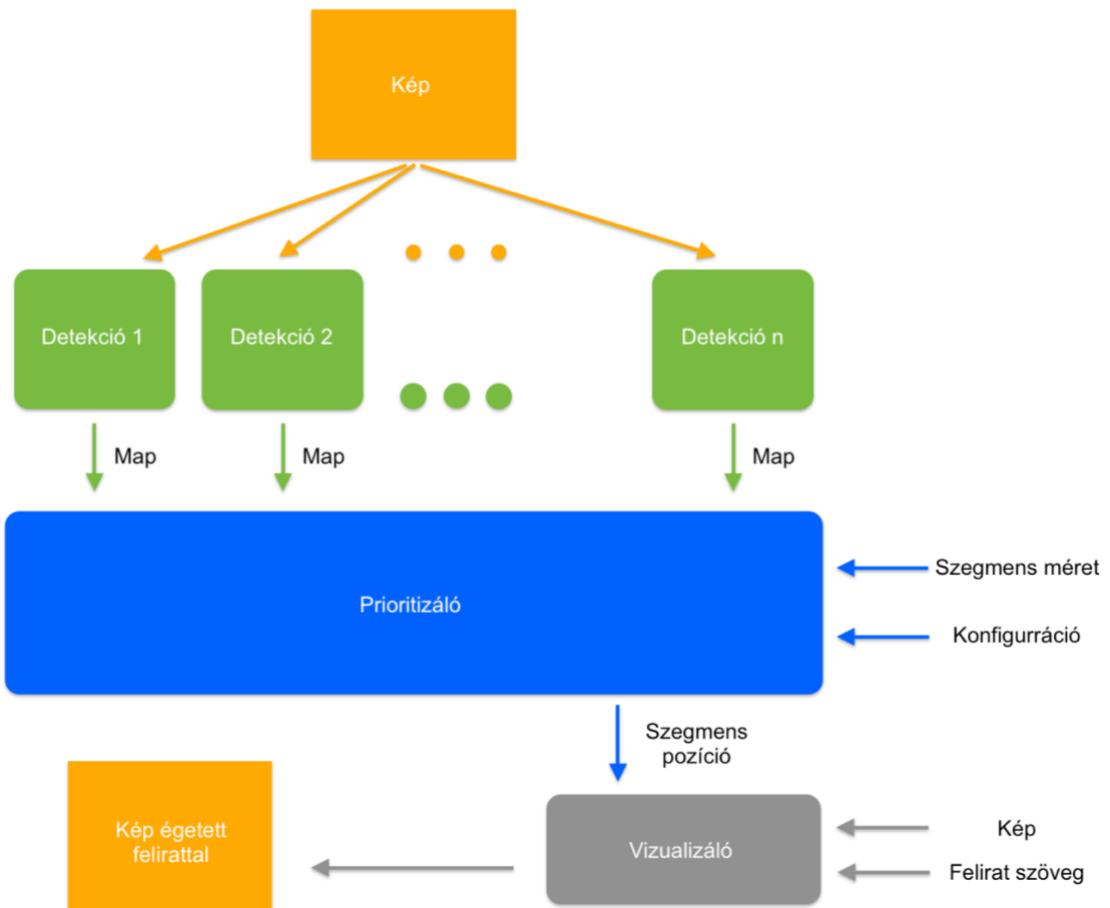
Élek száma és sűrűsége. Egy képnek fontos tulajdonsága, hogy mennyi él és milyen sűrűségen helyezkedik el rajta. Például tipikusan élgazdag terület lehet a betűket szavakat tartalmazó részek, de általában a kép fontosabb részei élesebben látszódnak ezért eleik, határvonalaiak *eldetekciójával* is érzékelhetők [20]. Tulajdonképpen élek vizsgálata jó megközelítésnek tűnhet elsőre, de előfordulhatnak olyan esetek mikor egy-egy logó, vagy fontosabb objektum elmosódottabban vagy erős közelítésben helyezkedik el a képernyőn. Kizárolag élek detektálásával nem tudnánk megtalálni a legjobb feliratpozíciót, mert az élek nem elégé sűrűek, vagy nincsenek ahhoz, hogy fontos területként találjuk meg.

3.2 Rendszerarchitektúra

A négy tulajdonság alapvetően határozza meg a tervezendő rendszer szövegillesztési algoritmusát. Szerettem volna olyan rendszert létrehozni, amely nyitott más tulajdonságok későbbi figyelembevételére és ezek integrálása könnyedén elvégezhető. A detekciós algoritmusokat futtató környezetet különálló *komponenseknek* tekintettem, melyek feladata a képekből történő információ kinyerés és ezek alapján egy megfeleltetési ábra, *map* létrehozása. Például a szövegdetektáló algoritmusnak feladata felismerni a szöveget a képen és a hozzá tartozó eredményt könnyen feldolgozható formára hozni egy olyan fájlal, ami rámutat a megtalált részletekre.

A komponensek eredménye, azaz a *map*-ek egy *prioritizáló* komponens által kerülnek feldolgozásra, amely egy konfiguráció és a szegmens méretének megfelelően keresi meg a legjobb pozíciót. A konfiguráció írja le, hogy milyen tulajdonságokat érdemes figyelembe venni. A szegmensmérőt pedig azt, hogy mekkora helyet keresünk a képen. A prioritizáló kimenetként egy pozíciót határoz meg, amelyet egy vizualizációs eszköz könnyedén fel tud dolgozni és ezek alapján a képre égetni.

Az architektúra tervhez készítettem egy sematikus ábrát (3.2. ábra). Látható, hogy a detekciós komponensek megkapják a képet és egy *Map* objektumot adnak oda a Prioritizálónak. A Prioritizáló az említett bemenetek alapján kimentként adja a felirat koordinátákat, amelyből a *Visualizer* készíti el a véleges képet.



3.2. ábra: Feliratszegmens pozicionáló rendszer architektúra

4 Megvalósítás

A detektáló komponenseket egységebe zártan közelítettem meg. Mindegyik bemenete egy kép és kimenete egy *map*, ami leírja a fontosnak tartott részeket, egy fekete-fehér szürkeárnyalatos kép formájában. A komponensek bemutatása prioritási sorrendben olvasható. Legfontosabb az éldetektor, mert univerzálisabban használható a többinél a képek típusától függetlenül.

4.1 Éldetekció

Egy kép lényegi tartalmát jól összefoglalják a látható objektumok élei. Az éleket önmagukban vizsgálva is már egy jó közelítést kaphatunk az ajánlott feliratpozícióhoz. Emberi szemmel akár csak az élek rajzolatából is felismerjük, hogy egy márkat, szöveget, vagy más objektumot ábrázol az eredeti kép. Tehát ez a technika egy általánosításnak tekinthető a másik háromhoz képest. Viszont bizonyos esetekben kizárolagos használata nem kívánt eredményhez vezetne. Például egy olyan képen, ahol a lényeges tartalom egyszerűbb mintázatú és kevesebb élt tartalmaz, mint a háttér vagy más képterület, nem kívánt eredményre vezetne a használata.

Többfajta algoritmus is létezik éldetekciós problémamegoldásra. Ilyenek például a *Sobel*, *Prewitt*, *Laplace* és a *Canny* detektorok [21]. *John F. Canny* által kidolgozott eljárás képes az élek erőteljes elhatárolására, így két színnel (fehér-fekete) leírható az adott éldetektált kép. A többi megoldás különböző árnyalatokat ad végeredményül, így használatukat nem láttam előnyösnek.

A Canny edge detektor számításai közben több állapottal rendelkezi és képes érzékelni élek erősségeinek széles skáláját. Részletes bemutatás nélkül elmondható, hogy az állapotok különböző képtranszformációs műveletek határait jelölik (például zaj csökkentés, gradiens meghatározás stb.). Továbbá kiemelendő, hogy az algoritmus szürkeárnyalatos képeken működik, ezért előzetes átskálázást kell végezni színes képeken [22].

Mivel szerettem volna mielőbb látni ötletem eredményességét, ezért az algoritmus alkalmazásával elkészítettem egy éldetekción alapuló feliratelhelyező programot, amely képes volt a feliratot azonosító szövegdobozt pozicionálni az éldetektált képen. A komponensbe zárást és az elvárt ki- és bemeneteknek megfelelő működést csak később

végeztem el. Működés során a képekből először szürkeárnyalatos változat készült, majd az élek detektálása következett. Az így kapott eredményen egy élminimumot kereső implementációt készítettem, amely képes detektálni azt a kép középpontjához legközelebb elhelyezkedő területet, amely a felirat szövegdobozának méretével rendelkezik és a legkevesebb élt tartalmazza.



4.1. ábra: Éldetekció alapján készített feliratpozíció ajánlás

A kapcsolódó ábrán (4.1. ábra) jól látható az elkészített program vizualizált eredménye. A már bemutatott ábrához (3.1 ábra) hasonló kiindulási képből egy letisztított, éleket tartalmazó kép készült, majd egy fehér szövegdoboz került elhelyezésre a legkevesebb élt tartalmazó részen. Az ábrán (4.1. ábra) pontosan látszódik a kizárolagos éldetekció használat hátránya. Ugyanis a képen szereplő arc fejtetője tartalmazta a legkevésbé sűrűbb területet, így a szövegdoboz rólóg a személy fejére, ami egy videó nézése közben igen zavaró lenne. Ez a probléma könnyen javítható lenne, ha a megtalált élekhez fontosságot is tudnánk társítani egy arc, vagy objektumdetektálóval, amely terveim között szerepelt.

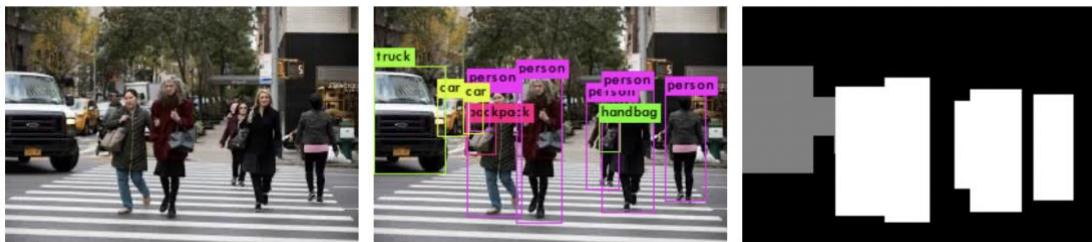
4.2 Objektumdetektálás

Feliratszegmensek okos elhelyezéséhez meg kell értenünk, hogy mi, vagy milyen objektumok szerepelnek a képen. Objektumdetekciót képeken már nagyfokú hatékonysággal lehet folytatni számítógépes technikákkal. Számos szoftver és algoritmus áll rendelkezésre, melyek képesek ezt a problémát megoldani. A legismertebb techinkák neurális hálózat alapúak, ezért betanítást és sok optimalizálást igényelnek. Mivel mesterséges intelligencia alapú megoldást szerettem volna nyújtani a feladatomra, ezért

mindenépp egy hálózat alapú algoritmusban gondolkoztam. Választásom a YOLO (*You only Look Once*) nevezetű architektúrára esett, mert számos felmérésen kiemelkedő helyen szerepelt és elsősorban ezt ajánlják objektumfelismerésre [24]. További nagy pozitívuma, hogy valós idejű működésre is képes, ami videók feldolgozásánál igen hasznos tud lenni [25].

YOLO használatánál lehetőségünk van a hálózat újra tanítására különlegesebb adathalmazokhoz, de alkalmazhatunk már előre feltanított modelleket is. Az én problémámat szerettem volna minél széleskörűbben megoldani és nem egy adott videótípusra optimalizálni, ezért úgy döntöttem, hogy nem merülök el egy megfelelő adathalmaz keresésében és felcímkezésében. Az alapértelmezett paraméterezés jól használható általánosabb objektumok felismeréséhez (pl.: emberek, autók, tárgyak), melyet alkalmASNak véltem a problémámhoz. Más beállításokkal bizonyos témaKörben hatékonyabb, de általánosságban pontatlanabb megoldás készült volna. Ebbe az irányba akkor érdemes tovább vizsgálódni, ha már tudjuk előre, hogy a feliratozandó videó milyen vizuális témaKört érint, így a tartalom szempontjából értékes objektumok hatékonyabban lokalizálhatók. Erre az opcióra továbbfejlesztési lehetőségeként tekintettem, nem pedig megvalósítandó részfeladatként az említett általános felismerési tulajdonságnak köszönhetően.

Az elkészült programom a fentiek alapján a képeket áttranszformálta, egy olyan megjelenítésbe, amely leírja, hogy a pixelek milyen fontosságúak a feliratra nézve. Tehát a bemeneti képen először lefuttattam az objektum felismerést, majd a megtalált objektumokat osztályoztam a fontosságuk alapján. Az emberi alakoknak adtam alapértelmezetten a legnagyobb prioritást, ezért ők kapták a legvilágosabb színezetet. A többi objektumot egy szinttel kevésbé fontosnak klasszifikáltam, ezért a kimeneti *map*-en ezek már szürke színnel szerepelnek. A folyamat eredményét szemlélteti a 4.2. ábra.



4.2. ábra: Objektumdetekciós eredmények felhasználása

A program bemenete egy egyszerű kép (4.2. ábra bal oldala), amin a YOLO segítségével objektumdetekció történik (középső kép). Az eredmény egy szürkeárnyalatú megfeleltetés (jobb oldal), amely feketének jelöli az objektumok szempontjából lényegtelennek tartott részeket és fehérrel a legfontosabbakat. A kettő közötti árnyalatok pedig a köztes észrevételeket. A technikának az az előnye, hogy így emberi szemmel és számítógéppel is könnyen emészthető a felismerés eredménye és jól leírja az objektumok egymás közötti relációját. A fontosabb tárgyak kitakarhatják a kevésbé fontosabbakat, ezért lehetséges, hogy a középső képen zöldön látszódó *handbag* a végeredményben már nem tűnik fel. Az eredményt felhasználva a komponenseket integráló program sokkal könnyebben megtudja határozni, hogy melyik objektumot a legkisebb probléma kitakarni, amennyiben nincs másra lehetőség.

4.3 Karakterlokalizáció

Ahogy az eddigi ábrákon is látszódott a karakterek és megjelenített szövegek fontos szerepet játszanak videókon és képeken. Feliratot helyezni egy már meglévő szövegre vagy feliratra különösen zavaró mert, nem csak a meglévő karaktereket teszi olvashatatlanná, a ráhelyezett szegmens elolvasása is nehézkessé válik. Karakterek felismerésével az *OCR* problematika foglalkozik, mely válaszolhatna arra a kérdésre, hogy hol helyezkednek el a szövegek a képen, de számítási ideje és komplexitása bőven meghaladja a számomra szükségeset. Az én esetben nem fontos, hogy a fellelhető szövegek mit írnak le, vagy milyen nyelven láthatóak, egyedül csak a helyzetük lényeges. Karakterek lokalizációjára, felismerés nélkül lényegesen gyorsabb és egyszerűbb megoldások állnak rendelkezésre, amelyek elegendők célomra.

2017-ben Xinyu Zhou és csapata publikálta a röviden *EAST* nevezetű neurális háló architektúrát, amely gyorsan képes a karakterek lokalizációjára és könnyen felhasználható más problémákra is [26]. Én az általuk készített modellt használtam fel a szövedetektálásra alkalmas programban [27].

A megírt program feladata hasonlóan az objektumdetektálóhoz (4.2 fejezet) csupán annyi volt, hogy kijelölje a fontosabb részeket a képen. Ebben az esetben nem láttam értelmét prioritásos sorrend felállításának, mert a szöveg mérete, helyzete nem feltétlen írja le fontosságát.

A szövedetektáló algoritmus úgy működik, hogy egy konfidenca számot rendel a felismert képrészletekhez. Azt, hogy mit detektál karkatereknek a konfidenzia

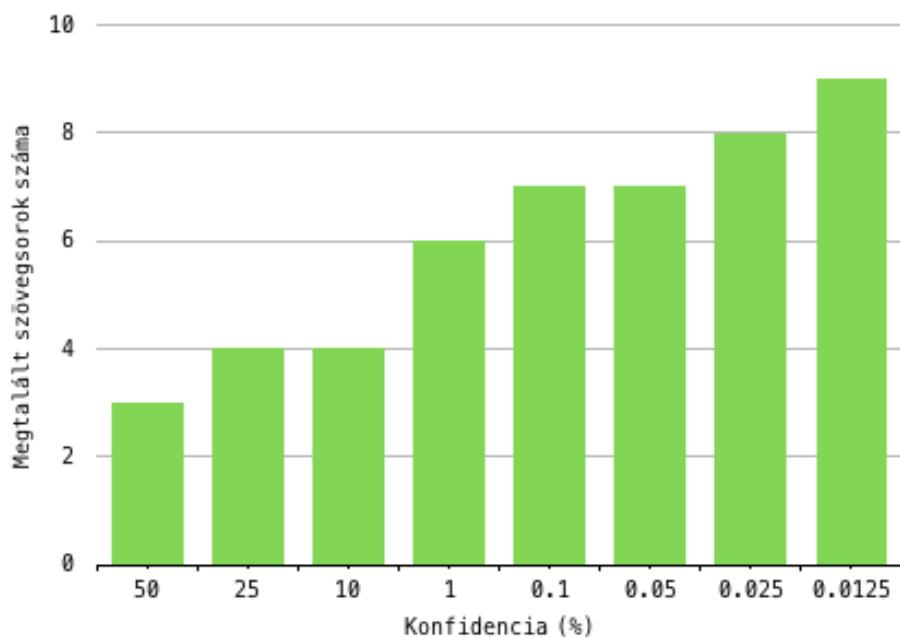
állításával tudjuk befolyásolni. Az érték finomhangolásához többek között a már bemutatott két képet is felhasználtam (1.2. ábra és 3.1 ábra). Először 50%-os konfidenciaszint felett tekintettem egy képrészletet szövegnek, ami a híradós képnél (3.1 ábra) teljesen megfelelő volt. Megtalálta az algoritmus az alul-felül látható karkatereket. De az autóverseny részletnél látható szövegeket kis pontossággal találta meg (4.3. ábra, felső kép).



4.3. ábra: Karakterdetektálás a mintaképeken 50%-os és 0.01%-os konfidenciaszinttel

50%-os konfidenciával látható, hogy a képen szereplő baloldali listán csak három nevet tudott detektálni. A százalékot szisztematikusan csökkentettem, melyet az alábbi diagrammon lehet szemmel követni (4.4. ábra). A diagramm függőleges tengelyén a fenti

ábra (4.3. ábra) bal oldalán található 20 elemű névsor felismert szövegsorainak számát látjuk. Vízszintes tengely a vizsgált konfidencia százalékokat azonosítja. 50%-nál a megtalált sorok száma három, míg 0.0125%-nál kilenc, de itt már tévesen is szövegként detektált egy nem releváns képrészletet (4.3. ábra, alsó kép). Sajnos látható, hogy ilyen alacsony érték mellett sem vett észre minden szöveget. A baloldali névsor felét így sem ismerte fel. A további csökkentés már erős negatív hatással lenne az eredményre, ezért a konfidenciát 0,025%-on hagytam, mivel ezzel az értékkel találta meg a legtöbb szöveget úgy, hogy más vizuális elemet, nem kategorizált karakternek.



4.4. ábra: Megtalált szövegsorok száma a konfidencia függvényében

A program kimente hasonló az objektumdetektálásnál (4.2 fejezet) már ismertetett formátumhoz azzal a különbséggel, hogy csak fekete és fehér színeket használtam információhordozóként. Ezzel a technikával a modulok kimenete könnyen összekapcsolható, ha képpontonként egy logikai VAGY műveletet végzünk. A keletkezett fehér színnek megfelelő egyesek (vagy nem nullások) leírják a fontos tartalmat.

4.4 Logó és márkJelzés detekció

Logók és márkJelzések detektálása nem csak feliratozási szempontból érdekes. Nagyvállalatoknak, hirdető cégeknek nagy értéket képvisel, ha a videók és TV adások során megtudják mondani, hogy egy-egy reklámkampány mennyire volt eredményes a logó és márkJelzések előfordulásának tekintetében. Illetve szükséges bizonyos

esetekben a logók kitakarása, ha a hozzá tartozó vállalat nem egyezett bele márka jelzésük feltüntetésébe.

A témakör nehézsége, hogy robosztus megoldás, ami képes több száz márkat felismerni, nagyon sok tanítómintát igényel, ráadásul a különböző felismerendő logók képeiből külön-külön. Mivel ilyen megoldás összeállítása igen nehéz és szabad felhasználású változata nem is beszerezhető, ezért a nagy hatékonyságú logódetektorról le kellett mondanom. Azonban nagy szakirodalommal rendelkezik a témakör és könnyű elmélyülni benne. Általában hálózatfelmérésről és konfigurálásról számolnak be a források [28]. Én szerettem volna egy olyan megoldást használni, ami egyszerűen beilleszthető eddigimunkámba és nem igényel hosszas környezetkialakítást. Végül a YOLO hálózatarchitektúra mellett döntöttem, melyhez találtam olyan súlyokat és konfigurációt, amely márka jelzések detektálásához készült [29]. A paraméterezés 47 márka jelzést képes felismerni melyek között szerepelnek nagy világmárkák is (pl.: Ford, Nvidia, CocaCola stb.) [30]. A következő képen (4.5. ábra) látszódik, hogy a megjelenített 7 márka közül csak 2-t ismer fel az algoritmus. Az Apple logójára is fel van tanítva a hálózat, azonban ebben a formában nem tudta beazonosítani, csak a Pepsi és a Google márkkákat. Más képeken a harapott alma jelet gond nélkül megtalálta. Mivel a tématerület igen nagy és javulást hosszas elmélyüléssel lehetne elérni ezért a gyors korrekcióra nem volt lehetőségem.



4.5. ábra: Logódetekció eredménye, ahol a 7 márka közül a színesen keretezetteket találta meg az algoritmus.

A hálózatot futtató algoritmusom azonos koncepció alapján lett elkészítve, mint a karakterdetekciós eljárás. A bemeneti kép eredménye egy ugyanolyan fekete-fehér megfeleltetése volt a képnak, melyből könnyen ki lehet nyerni a fontos képrészleteket algoritmikailag és emberi szemmel is.

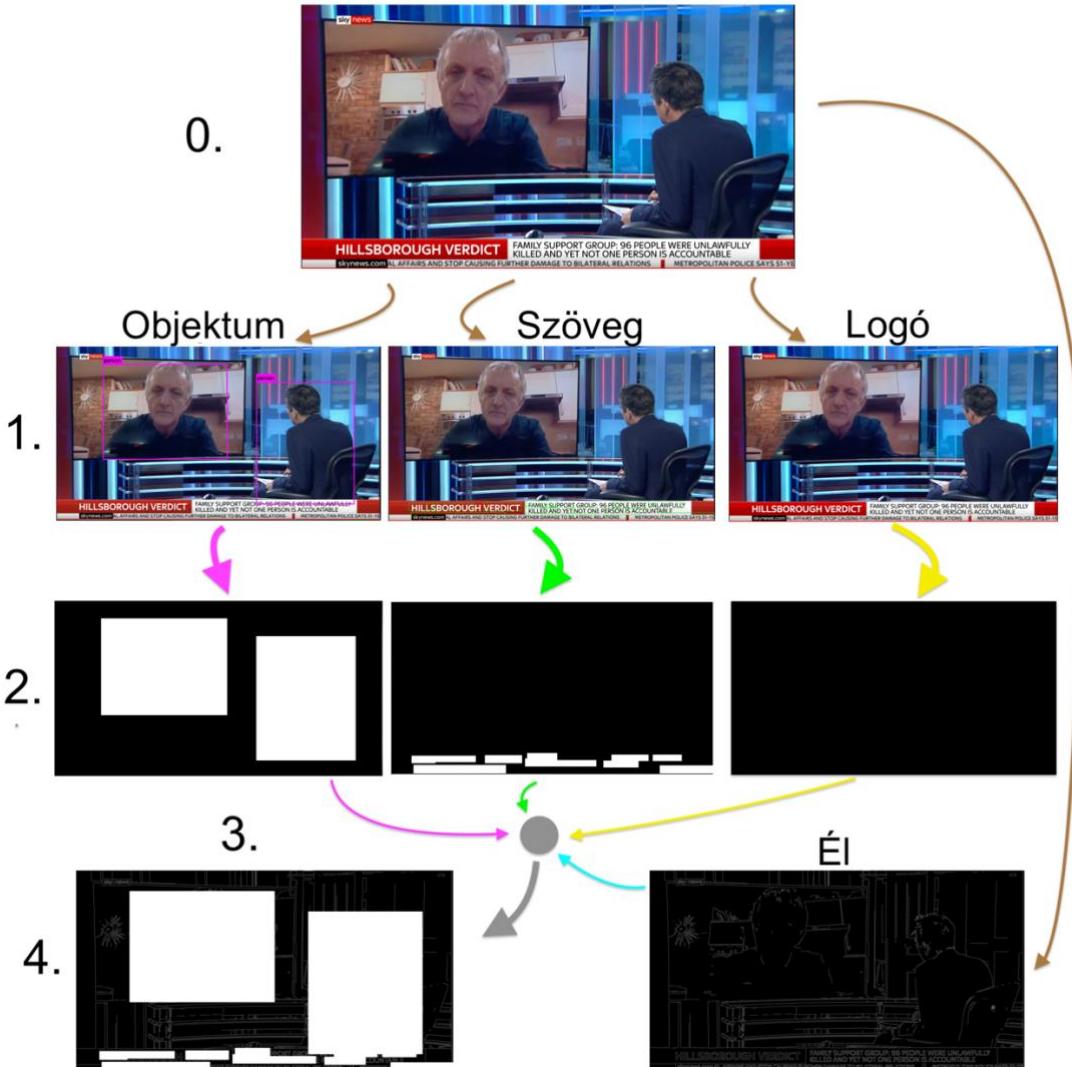
4.5 Algoritmusok felhasználása

Tehát az ismertetett képtulajdonságokat négy különféle módszerrel észleltem és dolgoztam fel. A következő lépés az eredmények felhasználása. A márka, objektum és szöveg detekciónál megtudjuk mondani, hogy felirat a felismert területekre ne kerüljön, mert észleltük a konkrét pozíciójukat és kategorizálásunk szerint nagy befolyással vannak az összesített vizuális élményre. Illetve ezekben az esetekben az eredeti képen történik a detektálás és nem egy áttranszformált verziót. Ezt a három detekciót elneveztem *kizárást* detekciónak. Éldetekciónál azonban más a helyzet. Sok esetben található annyi él a képen, hogy fedésük nélkül képtelenség lenne elhelyezni a feliratot. Valamint a számításhoz szürkeárnyalatúvá kell alakítani a képet. A megtervezett rendszer architektúrája azonban képes elfedni ezt a különbséget. Az előzetes transzformációkat komponensen belülre kell mozgatni.

Tehát a prioritizáló komponens a megkapott inputokat összesíti egy képre, melyen ezáltal megjelennek a fontos és kevésbé fontos területek az élek mellett és kialakul a *prioritizált* kép. Az így kapott eredményen egy csúszó ablak segítségével végig iteráltam. A csúszó ablak területe megfelel a megadott felirat szövegdoboz méretének. Erre a területre eső sűrűséget számoltam ki ciklikusan, ami gyakorlatilag a nem fekete képpontokhoz tartozó értékek összegének felel meg. Mivel nagyobb felbontású képeken is szerettem volna használni az algoritmust, ezért a képpontonkénti iterálás és számításvégzés hosszas ideig tartott. Az elhúzódó számítási időt, úgy küszöböltem ki, hogy az egyes iterációkhoz tartozó növekményt paraméterezhetővé tettem. Ennek köszönhetően a számítás gyorsítására van lehetőségünk, ami hasznos lehet nagyobb felbontású képeknél és videóknál.

A képfeldolgozási folyamatot a lenti ábrán (4.6. ábra) vizualizáltam egy tetszőlegesen választott képpel. 0-ik lépésben a SkyNews híradó kiindulási képe látható. Első lépésben kiemelésre kerültek a beazonosított elemek (objektum, szöveg és logó). A SkyNews logója is szerepel a képen, de ezt a logódetektor sajnos nem képes beazonosítani, ezért a hozzá tartozó *map* fekete marad. Azonban a szöveg és objektum felismerő sikeresen azonosítja a látható elemeket és a második sorban vizualizálásra kerültek a detektált területek fehér lenyomatai. Harmadik lépésben, a három kizárást kép egyesítésre kerül az éldetektor eredményei mellett. Ezáltal előáll a már említett *prioritizált* kép.

Eredeti kép



4.6. ábra: Képfeldolgozás bemutatása



4.7. ábra: Az újrapozícionált felirat elhelyezve a képen

A megtervezett rendszer algoritmusa alapján készült el a fenti ábrán látható feliratozás (4.7. ábra). A definiált képtulajdonságok figyelembevételével a legkevésbé zavaróbb helyre került a felirat. Ezt zölden bekeretezve jelöltem. Közelebbről megnézve észrevehető, hogy a felirat szövegdobozának bal sarka beleesik az emberi objektumnak detektált részbe, de annak csak a jobb felső sarkát érinti. További példa feliratozott képek a 6 Függelék részben találhatók (6.1. ábra)

5 Lezárás

Lezárásként megemlítem, hogy miken lenne érdemes tovább dolgozni a megoldásomat illetően, valamint egy rövid összefoglalás adok munkámról.

5.1 Továbbfejlesztési javaslatok

Munkám jó alapot nyújt videókon történő feliratozások javítására, melyet a későbbiekben tervezek elkészíteni. Továbbfejlesztési javaslatként a detektorok finomhangolását emelném ki. Sok esetben a megtalált objektumokhoz tartozó kizárt terület nagy részt fed le a képen és ezzel megnehezíti, hogy a felirat kényelmes közelégebe kerüljön a lényegi tartalomhoz. Ezzel csökkenteni lehetne a kényelmetlen szemmozgások hosszát.

A logódetektor pontosításán is érdemes elgondolkodni, hogy több és jobb minőségen találja meg a márka jelzéseket. A feliratok képre égetése mellett hasznos lenne a feliratfájl generálásában is elmerülni, hogy a megoldás széleskörűbben is használható legyen. Erre a rendszerarchitektúra kitűnően alkalmas, hiszen a vizualizátor helyett csak egy feliratfájl létrehozására alkalmas komponenst kell megvalósítani.

Önmagában megoldásom alkalmas címek, reklámszövegek kényelmes elhelyezésére plakátokon, képeken.

5.2 Összefoglalás

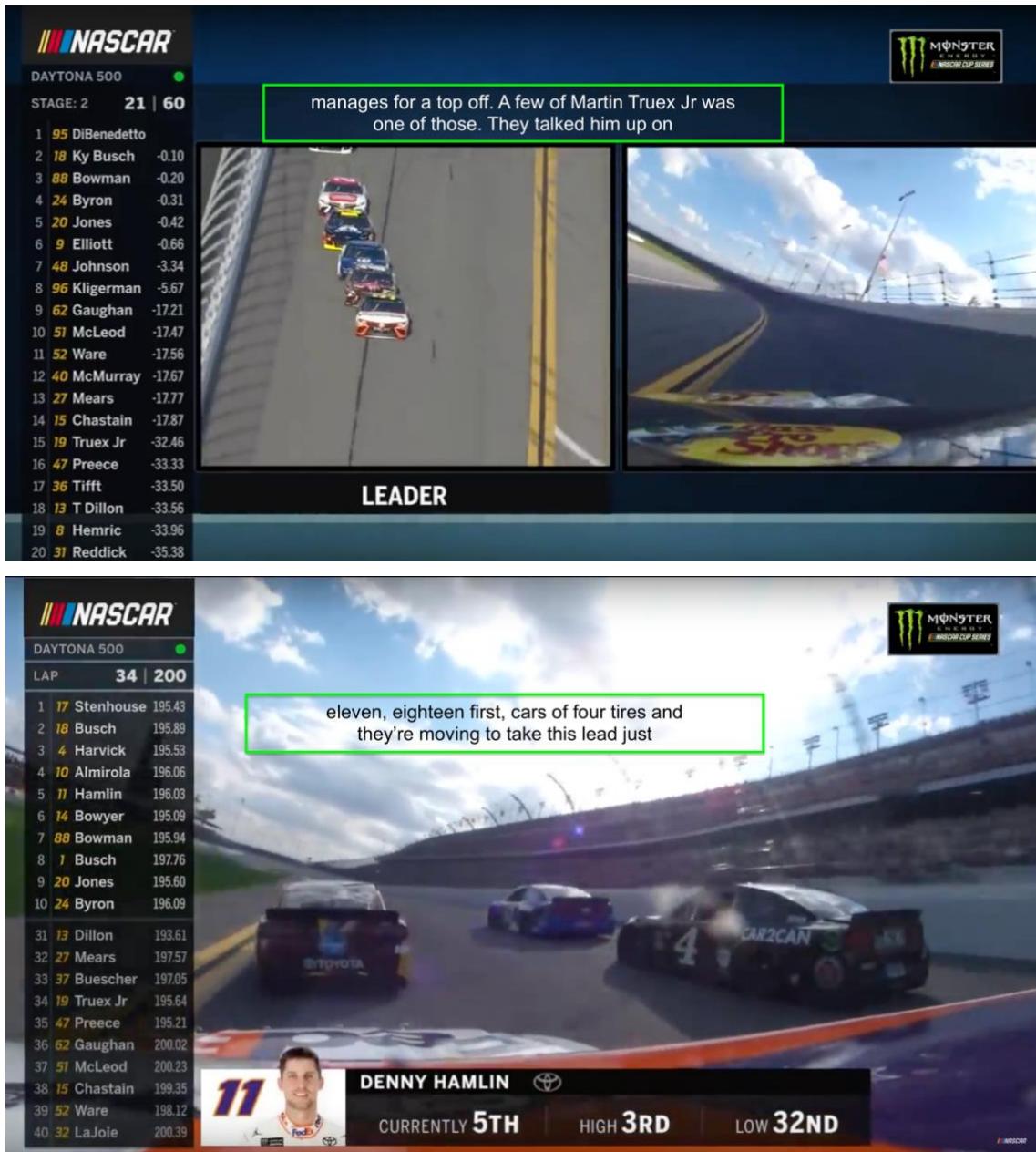
Dolgozatomban bemutattam a feliratozás fontosságát és jelenlegi automatizált megoldásokat a téma kör ismertetése mellett. Irodalomkutatást végeztem a téma körben és rámutattam a pozícionálási megközelítésekre, melyek párbeszédekkönnyebb megértését teszik lehetővé. Célom ismertetése után egy rendszerarchitektúrát terveztem, amely lehetővé teszi feliratok okos pozícionálását vizuális tartalmak alapján. Az algoritmus figyelembe veszi a megjelenő logókat, a képen már szereplő szövegeket, a különböző objektumokat és képes az élek detektálására is. A rendszer használatával elkerülhetjük, hogy az elhelyezendő szöveg zavaró helyzetbe kerüljön megadott képeken. Az implementált megoldást komponensekre bontva bemutattam és egy példán keresztül ábrázoltam a feldolgozás folyamatát. Legvégül továbbfejlesztési javaslatokat tettem.

Irodalomjegyzék

- [1] Yongtao Hu, Jan Kautz, Yizhou Yu, Wenoing Wang, *Speaker-following Video Subtitles*, arXiv:1407.5145v1, 2014.07, https://www.researchgate.net/publication/264123179_Speaker-Following_Video_Subtitles, (2019.11.18)
- [2] Shea Laverty: *What is Open Captioning?*, <https://www.techwalla.com/articles/what-is-open-captioning>, (2019.11.09)
- [3] Washington University: *What is the difference between open and closed captioning?* <https://www.washington.edu/doit/what-difference-between-open-and-closed-captioning>, (2019.11.18)
- [4] Google Cloud Speech-to-Text Service, <https://cloud.google.com/speech-to-text>, (2019.11.08)
- [5] Amazon Transcribe Service, <https://aws.amazon.com/transcribe/>, (2019.11.08)
- [6] IBM Watson Speech-to-Text Service, <https://www.ibm.com/cloud/watson-speech-to-text>, (2019. 11. 08)
- [7] Simon James: *5 Good Open Source Speech Recognition/Speech-to-text Systems*, <https://fosspost.org/lists/open-source-speech-recognition-speech-to-text>, (2019.11.08)
- [8] VLC Media Player, <https://www.videolan.org/vlc/index.html>, (2019.11.10)
- [9] Alberto Sabater: *Automatic Subtitle Synchronization through Machine Learning*, <https://machinelearnings.co/automatic-subtitle-synchronization-e188a9275617>, (2019.11.10)
- [10] Open Toegankelijke, , <https://www.opentoegankelijk.be/exhibitors>, (2019.11.18)
- [11] Wataru Akahori, Tatsunori Hirai, Shigeo Morishima: *Dynamic Subtitle Placement Considering the Region of Interest and Speaker Location*, *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 102-109, ISBN 978-989-758-227-1, <https://pdfs.semanticscholar.org/2864/cfd949e5b787b1fd22c05b5bb6450197a72e.pdf> (2019.11.06)
- [12] Apostolidis, E. And Mezaris, V., *Fast shot segmentation combining global and local visual descriptors*, *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp 6583-6587
- [13] Katti, H., Rajagopal, A. K., Kankanhalli, M., and Kalpathi, R., *Online estimation of evolving human visual interest*, *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)*, 2014
- [14] Monaci G.: *Towards real-time audiovisual speaker localization*, 2011

- [15] Wikipedia: *Comparision of optical character recognition software*, https://en.wikipedia.org/wiki/Comparison_of_optical_character_recognition_software, (2019.11.19)
- [16] Nithiroj Tripatarasit: *Logo Detection Using PyTorch*, (2018.06), <https://medium.com/diving-in-deep/logo-detection-using-pytorch-7897d4898211>, (2019.11.19)
- [17] Ivan's Software Engineering Blog: *Robust logo detection with OpenCV*, <https://ai-facets.org/robust-logo-detection-with-opencv/>, (2019.11.19)
- [18] Francois from TalkWalker: *10 best Image Recognition tools*, <https://www.talkwalker.com/blog/best-image-recognition-tools>, (2019.11.19)
- [19] Divyansh Dwivedi: *Face Detection for Beginners*, <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>, (2019.11.19)
- [20] Wikipedia: *Edge Detection*, https://en.wikipedia.org/wiki/Edge_detection, (2019.11.19)
- [21] Nika Tsankashvili: *Comparing Edge Detection Methods*, <https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>, (2019.12.11)
- [22] Sofiane Sahir: *Canny Edge Detection Step by Step in Python*, <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>, (2019.11.20)
- [23] Rodrigo Verschae, Javier Ruiz-del-Solar: *Object Detection: Current and Future Directions*, 2015
- [24] Arthur Ouaknine: *Review of Deep Learning Algorithms for Object Detection*, 2018, <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852>, (2019.11.23)
- [25] YOLO: *YOLO: Real-Time Object Detection*, <https://pjreddie.com/darknet/yolo/>, (2019.11.23)
- [26] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiram He, Jiajun Liang: *EAST: An Efficient and Accurate Scene Text Detextor*, arXiv:1704.03155, 2017.04.11, <https://arxiv.org/abs/1704.03155>, (2019.11.24)
- [27] Adrian Rosebrock: *OpenCV Text Detection (EAST text detector)*, 2018.08.20, <https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/>, (2019.11.24)
- [28] Ankur Singh: *Logo detection in Images using SSD*, 2018.07.11, <https://towardsdatascience.com/logo-detection-in-images-using-ssd-bcd3732e1776>, (2019.11.26)
- [29] Akarshzingade, *Logo-Detection-YOLOv2 GitHub Repository*: <https://github.com/akarshzingade/Logo-Detection-YOLOv2>, (2019.11.26)
- [30] *YOLOv2 logódetekcióval felismerhető márkák*: <https://github.com/akarshzingade/Logo-Detection-YOLOv2/blob/master/obj.names>, (2019.11.26)

6 Függelék



6.1. ábra: Feliratozott képek egy Nascar videóból