

36. Országos Tudományos Diákköri Konferencia
Had- és Rendészettudományi Szekció
Budapest, 2023. április 18-20.
Nemzeti Közszolgálati Egyetem
Rendészettudományi Kar

Képfeldolgozáson alapuló hazugságvizsgáló rendszer

Jelige: *hazugságvizsgáló*

Pályamunka lezárásának időpontja: 2022. 11

2023.

Tartalomjegyzék

1. ABSZTRAKT	5
2. ABSTRACT	6
3. BEVEZETÉS	7
4. IRODALOMKUTATÁS	9
4.1. HASONLÓ RENDSZEREK ELEMZÉSE	9
4.1.1. <i>Deception Detection in Videos</i> [1]	9
4.1.2. <i>Facial Micro-Expression Detection in Hi-Speed Video Based on Facial Action Coding System</i> [2] ..	11
4.1.3. Összehasonlítás	14
4.2. MOTION PROFILES FOR DECEPTION DETECTION USING VISUAL CUES[6]	15
4.2.1. <i>A tanulmány ismertetése</i>	15
4.2.2. <i>Fej- és kézfoltkövetés</i>	15
4.2.3. <i>Arckövetés</i>	15
4.2.4. <i>Tesztelések</i>	17
4.2.5. <i>Értékelés</i>	17
4.3. MULTIMODAL DECEPTION DETECTION USING AUTOMATICALLY EXTRACTED ACOUSTIC, VISUAL, AND LEXICAL FEATURES[7]	18
4.3.1. <i>A tanulmány ismertetése</i>	18
4.3.2. <i>The Box of Lies</i> gyűjtemény	18
4.3.3. <i>Metodológia</i>	19
4.3.4. <i>Tesztek és eredmények</i>	20
4.4. HUMAN BEHAVIOR AND DECEPTION DETECTION[8]	21
4.4.1. <i>Tudományos áttekintés – a hazugság viselkedésben megnyilvánuló jelei</i>	21
4.4.2. <i>Tudományos áttekintés – képességek a hazugság észleléséhez</i>	22
4.5. DECEPTION DETECTION: STATE OF THE ART AND FUTURE PROSPECTS[9]	23
4.5.1. <i>A hazugság felismerése</i>	23
4.5.2. <i>Ellenőrizhetőségi megközelítés</i>	23
4.5.3. <i>A megtévesztés pszichofiziológiai kimutatása</i>	24
4.6. ÖSSZEGZÉS	25
4.6.1. <i>A videó rögzítésének elvárásai</i>	25
4.6.2. <i>A kérdések feltételének módja</i>	26
4.6.3. <i>Videó előkészítése a feldolgozásra</i>	26
4.6.4. <i>Releváns Action Unitok</i>	27
5. KÖVETELMÉNY SPECIFIKÁCIÓ	28
5.1. WEBES FELHASZNÁLÓI INTERFÉSZ	28
5.2. VIDEÓFELDOLGOZÓ ALKALMAZÁS	28
6. TERVEZÉS	29
6.1. FELHASZNÁLANDÓ TECHNOLÓGIÁK	29
6.1.1. <i>Webes interfész, videókezelés</i>	29
6.1.2. <i>Az arcfelismeréshez felhasználandó technológia</i>	29
6.1.3. <i>Keretrendszer kiválasztása</i>	30
6.1.4. <i>A videófeldolgozáshoz felhasználandó technológia</i>	30

6.2.	RENDSZERTERV	32
6.2.1.	<i>Webes felhasználói interfész</i>	32
6.2.2.	<i>Videófeldolgozó alkalmazás</i>	34
6.2.3.	<i>Komponensek</i>	35
6.3.	FUNKCIÓLISTA	36
6.4.	FEJLESZTÉS TERVEZETT MENETE	37
6.4.1.	<i>Backend</i>	37
6.4.2.	<i>Frontend</i>	37
6.4.3.	<i>Utómunkálatok</i>	38
6.4.4.	<i>A megvalósítás üteme</i>	38
7.	FEJLESZTÉS	39
7.1.	A FEJLESZTÉS CÉLJA	39
7.2.	A RENDSZER HASZNÁLATÁNAK ELŐKÖVETELMÉNYEI	39
7.3.	ÜZLETI LOGIKA	39
7.3.1.	<i>Action Unit detektálás</i>	39
7.3.2.	<i>Kezdeti implementáció Python nyelven</i>	39
7.3.3.	<i>Implementálás C++ nyelven</i>	41
7.4.	WEBES FELHASZNÁLÓI INTERFÉSZ	45
7.4.1.	<i>Fejlesztési környezet megválasztása</i>	45
7.4.2.	<i>Megvalósított funkciók</i>	45
7.5.	KONTÉNERIZÁCIÓ	48
8.	TESZTELÉS	49
8.1.	UNIT TESTING	49
8.1.1.	<i>Tesztvideó előállítása</i>	49
8.1.2.	<i>A teszteléssel szembeni elvárások</i>	50
8.1.3.	<i>A tesztelés technikai megvalósítása</i>	50
8.1.4.	<i>A Unit Tesztek eredményei</i>	51
8.2.	MANUÁLIS TESZTELÉS	53
8.2.1.	<i>Tesztelési módszertan</i>	53
8.2.2.	<i>Tesztek eredményei</i>	54
9.	ÉRTÉKELÉS	57
9.1.	BACKEND ALKALMAZÁS	57
9.2.	FRONTEND ALKALMAZÁS	57
9.2.1.	<i>További lehetőségek</i>	57
9.3.	TESZTELÉSEK	58
9.4.	TAPASZTALATOK ÖSSZEGZÉSE	58
10.	TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK	59
10.1.	HAMIS REAKCIÓK KISZŰRÉSE	59
10.2.	TÖBB SZEMÉLY VIZSGÁLATA EGYSZERRE	59
10.3.	AUTOMATIKUS KÉPFELJAVÍTÁS	59
10.4.	EGYÉB FAKTOROK VIZSGÁLATA	60
	IRODALOMJEGYZÉK	61

ÁBRAJEGYZÉK.....	62
TÁBLÁZATJEGYZÉK	63

1. Absztrakt

A szakdolgozat témája egy olyan rendszer megalkotása, amely képes az emberi mimikai reakciók alapján hazugságra utaló jeleket megállapítani. Egy átlagos verbális kommunikáció során 20 fő arcizmunk dolgozik összhangban, miközben kifejezzük magunkat. A mimikai reakcióink óhatatlanul is visszajelzést adnak az aktuális érzelmi állapotunkról: bizonyos alapvető érzelmeket, mint például az öröm és a szomorúság, egyértelműen fel lehet ismerni. Ezek a beszéd során megjelenő reakciók, vagy másnéven mikro-kifejezések, különböző időintervallumokra jelenhetnek meg: pár tizedmásodperctől egészen akár néhány másodpercig is. Ebből fakadóan ezeket nem csak emberi szemmel észlelni, hanem befolyásolni is rendkívül nehéz feladat. Gondoljunk csak egy pókerjátszmára: külön tehetség kell az úgynevezett póker arc kifejlesztéséhez, ami nem más célt szolgál, minthogy elrejtse a „nemkívánatos” érzelmeket.

Hazugság során az ember általában a normálistól eltérő érzelmi állapotba kerül. Ennek számos oka lehet, a mások – jellemzően hátsó szándékok által vezérelt – átveréséből fakadó izgalomtól kezdve, a lebukástól való félelemtől át egészen a büntudatig. Ilyenkor a legfőbb cél a hiteles benyomás keltése és a gyanú elkerülése, legtöbbször meg nem történt események kreálásával és/vagy azok oly módon történő leírásával, amely lehetővé teszi a többféle értelmezést. Akárhogy is történik a hazugság megalkotása, az minden esetben valamilyen szintű mentális erőfeszítést igényel, mely során érzelmek generálódnak. Mivel érzelmeink legfőbb kivetülési területe az arcunk, ezért a mimikai reakciók vizuális vizsgálatával jó eséllyel találhatók hazugságra utaló jelek. Ezek megtalálásához össze kell hasonlítani a vizsgálandó személy normál körülmények közötti reakcióit egy esetleges hazugság esetén fellépő reakciókkal.

Létezik egy úgynevezett Facial Action Coding System, ami egy anatómiai alapokon nyugvó, széleskörű leíró rendszer, mely a mimikai reakciókban résztvevő arcizmok mozgásait osztályozza és csoportosítja, ezzel létrehozva az úgynevezett Action Unit-okat. Bizonyos kutatásokból kiderül, hogy hazugság során mely Action Unit-ok észlelhetők jellemzőbben és nagyobb intenzitással az arcon. A dolgozat célja, hogy egy videóanyag segítségével megállapítsa a vizsgálandó személy normál körülmények között fellépő reakcióit, összehasonlítsa azt egy potenciális hazugság során fellépő reakciókkal, miközben olyan reakciók jelenlétét keresi, amelyek hazugságra utalhatnak.

Az önkéntesek segítségével elvégzett manuális tesztek során minden résztvevő kapott 5 kontroll kérdést, 5 kérdést, melyekre fals pozitív, valamint újabb 5 kérdést, melyre őszinte, igaz válaszokat kellett adniuk. A teszten az elvárt eredmények születtek: hazugság során a rendszer jelezte az erre utaló jeleket, igazmondás során pedig nem léptek fel hazugságra utaló jelek. A rendszer számos továbbfejlesztési lehetőséggel rendelkezik, azonban kijelenthető, hogy a tesztelés során szerzett tapasztalatok alapján biztató módon képes a hazugságra utaló jelek felderítésére.

2. Abstract

The topic of this thesis is to create a system that can detect signs of lies based on human facial reactions. During an average verbal communication, our 20 main facial muscles work in unison as we express ourselves. Our facial expressions inevitably provide feedback on our current emotional state: we can clearly recognize certain basic emotions, such as joy and sadness. These reactions – which are also called as micro-expressions – appearing during speech for different time intervals: from a few tenths of a second to a few seconds. Consequently, it is an extremely challenging task not only to perceive them with the human eye, but also to influence them. Think of a poker game: it takes a special talent to develop a so-called poker face, which serves no other purpose than to hide "undesirable" emotions.

When lying, a person usually gets into a different emotional state than normal. There can be many reasons for this, from the excitement of deceiving others (typically driven by ulterior motives), through the fear of being revealed to the feeling of guilt. In such cases, the main goal is to create an authentic impression and avoid suspicion, mostly by creating events that did not happen and/or describing them in a way that allows multiple interpretations. No matter how the lie is created, it always requires an unavoidable mental effort, during which emotions are generated. Since our face is the main projection area of our emotions, there is a good chance that signs of lies can be found by visual inspection of facial expressions. To find them, it is necessary to compare the reactions of the subject under normal circumstances with the reactions occurring in the event of a possible lie.

There is a so-called Facial Action Coding System, which is an anatomically based, broad descriptive system that classifies and groups the movements of the facial muscles involved in facial expressions, thus creating the so-called Action Units. Certain research reveals which Action Units can be detected more typically and with greater intensity on the face during a lie. The purpose of this thesis is to use a video material to determine the reactions of the subject under normal circumstances and compare them with the reactions occurring during a potential lie, while looking for the presence of reactions that may indicate a lie.

During the manual tests performed with the help of volunteers, all participants were given 5 control questions, 5 questions for which they had to give false positive, and another 5 questions for which they had to give honest, true answers. The tests showed the expected results: in case of false positive answers the system indicated the signs of lying, while no signs of lying occurred when telling the truth. The system has many possibilities for further development; however, it can be stated that, based on the experience gained during testing, it is promisingly capable of detecting signs of lies.

3. Bevezetés

Jelen dolgozat a vizuális hazugságfelismerést a mimikai reakciók vizsgálatára szűkíti.

Számos szakirodalom és tanulmány foglalkozik azzal, hogy milyen módszerek és lehetőségek léteznek egy esetleges hazugság jeleinek az észlelésére az arcizommozgások vizsgálatával. A verbális kommunikáció során körülbelül 20 különböző arcizom dolgozik annak érdekében, hogy kifejezhessük magunkat. Ezekből az izommozgásokból épülnek fel az úgynevezett mikro-kifejezések. Ezen mikro-kifejezések mindegyike különböző ideig jelenhet meg az arcunkon: néhány ezredmásodperctől egészen akár néhány másodpercig is tarthatnak. Ezeket a kifejezéseket általában nagyon nehéz – vagy nem is lehetséges – észrevenni és befolyásolni. Nem véletlen, hogy bizonyos szituációkban, például egy pókerjátszmában, különleges tehetség és sok gyakorlás kell ahhoz, hogy az embernek jó pókerarca legyen, aminek segítségével megakadályozható vagy befolyásolható az érzelmek kivetülése. A mimikai reakciók rendszeresen megjelennek az arcunkon a különböző mindennapi tevékenységeink során, és általában tükrözik az aktuális érzelmi állapotunkat.

Létezik ezen a területen egy jól működő, az emberi anatómián alapuló leíró rendszer, ami nem más, mint a Facial Action Coding System [FACS]. Ez az arckifejezéseket az izommozgás egyes összetevőire, úgynevezett Action Unit-okra [AU] bontja.

Ezt a rendszert elsősorattal használják az arcfelismerés iránt érdeklődő animátorok és informatikusok, de sokszor alkalmazzák különféle tudományos környezetekben kutatásokhoz.

Az Action Unit-ok alatt gyakorlatilag az arcizmok mozgásainak a kategorizálását értjük. Minden AU egy konkrétan felismerhető izommozgáshoz vagy azok egy csoportjához egy számot rendel hozzá. Az alábbi táblázatban néhány példa látható.

Azonosító	Leírás	Arcizmok
12	Ajkak széleinek széthúzása	Zygomatic Major
43	Szemek becsukása	Levator Palpebrae Superioris
45	Pislogás	Levator Palpebrae and Contraction of Orbicularis Oculi, Pars Palpebralis

3.1. táblázat: néhány Action Unit példaként

Az elsődleges cél, hogy egy megadott videóban felismerjük a vizsgálandó személy arcát, és a FACS alapján a releváns¹ Action Unit-ok meghatározását követően következtetést vonhassunk le azt illetően, hogy a vizsgált személy reakcióiból megállapíthatóak-e, és amennyiben igen, akkor milyen valószínűséggel állapíthatók meg a különböző hazugságra utaló jelek.

Mint az már fentebb is említésre került, a mimikai reakcióink az érzelmeink legfőbb vizuális jelei, melyek önkéntelenül is megjelennek rajtunk. Mivel hazugság során általában az emberek képtelenek teljesen higgadtak maradni, az igazság eltitkolásával járó mentális erőfeszítés

¹ A hazugság során releváns Action Unitok kiválasztása a következő források segítségével történt: [10], [11], [12]

legtöbbször kivált valamilyen érzelmi reakciót – még akkor is, ha az az emberi szem számára nem feltétlen egyértelműen észlelhető.

Léteznek tanulmányok, amelyek többek között azt vizsgálják, hogy hazugság során milyen arckifejezések, azaz Action Unit-ok jelennek meg nagy valószínűséggel. Fontos azonban ezen a ponton megjegyezni, hogy ez még nem jelenti azt, hogy ezek kizárólag hazugság során lennének jellemzőek. Vannak ugyanis olyan emberek, akik esetében a mimikai kifejezések alapjáraton intenzívebbek, míg akadnak olyanok is, akiknél pedig kevésbé. Éppen ezért, minden elemzésnél szükséges, hogy a hazugságvizsgálatot megelőzően megfelelő információkkal rendelkezünk azzal kapcsolatban, hogy az adott személy esetén melyek azok az Action Unitok, amik megjelenhetnek, és azok milyen intenzitással jelennek meg.

A projekt célja, hogy a releváns Action Unit-ok irányába elindulva egy olyan rendszert hozzon létre, amely képes egy viszonylag pontos predikciót adni azzal kapcsolatban, hogy a vizsgált személy igazat mondott-e vagy pedig nem.

4. Irodalomkutatás

4.1. Hasonló rendszerek elemzése

4.1.1. Deception Detection in Videos[1]

A projekt videóanyagok feldolgozásával és elemzésével szűri ki a mimikai mikro-kifejezéseket, amik alapján megállapítja, hogy ezek a jelek szándékosan megtévesztő magatartásra utalnak-e. A predikció eredményének pontosítása érdekében felhasználják a videó alatt lévő hangsvot, és az alany hangszínének elemzését összevetik a képfeldolgozás eredményével.

A projekt célja, hogy egy úgynevezett mozgásdinamikai elemzés segítségével ismerje fel az arc mikro-kifejezéseit. Ezen jelek felismerését egy kétlépcsős megvalósítással oldja meg: alacsony szinten egy „Dense Trajectories” (sűrűségi pálya) nevű eljárással megállapítják az arc mozgását és ezek bizonyos változásait, egy magas szintű megoldással pedig betanítanak egy arc mikro-kifejezés felismerőt ezekkel az alacsony szintű megoldásokkal, és ezek megállapításai lesznek a mérvadók. A projekt mindemellett az eredmény pontosítása érdekében hang és szöveganalízist is felhasznál.

Mozgás jellemzői

A bemeneti forrás egy videó, melyben az alany igaz vagy hamis állításokat tartalmazó kijelentéseket tesz. A videóban lévő arc pozíciója változhat, tehát nem feltétlen néz egyenesen a kamerába. A képfeldolgozás IDT (Improved Dense Trajectory) segítségével történik. Ez kiszámítja a helyi jellemzőket az egymást követő képkockákban, és RANSAC eljárás segítségével megbecsüli a kamera mozgását. Miután a kamera mozgásának hatása kiszűrésre került, a módszer több térbeli egységben sűrűn mintavételezi a jellemzőpontokat, majd korlátozott számú képkockán keresztül követi őket, hogy megakadályozza az elsodródást. Jelen esetben az MBH leíró egy jó választás, mivel az MBH optikai áramlási deriváltakat számít ki és a mozgás deriváltjait rögzíti, nem pedig elsőrendű mozgásinformációkat. Mivel a mikro-kifejezéseket kimutatása a cél, a leírónak a mozgás változásait kell képviselnie, amelyet az MBH rögzít.

Arcjellemző kódolás

Mivel minden videó más-más jellemvonásokat hordoz magában, ezért Fisher Vektor alkalmazásával változó számú tulajdonságot rendel hozzá egy rögzített hosszúságú vektorhoz. A Fisher Vektor kombinálja a generatív és diszkriminatív modellek előnyeit. Széleskörben alkalmazott a gépi látásos feladatokban, mint például a képfelismerés, tevékenység felismerés stb.

Arcon megjelenő mikro-kifejezések előrejelzése

Az arckifejezések fontos szerepet játszanak a megtévesztő viselkedés előrejelzésében. Az öt leggyakoribb megtévesztésre utaló mimikai jel a homlokráncolás, szemöldök emelés, ajak sarokának felfelé húzása, ajkak kinyúlása és a fej oldalra történő fordítása. Az adatbázisuk

minden egyes videóját rövid, fix időtartamú videoklipekre osztják, és ezeket a klipeket mikro-kifejezések szerinti címkékkel jelölik meg. Az elemzés során ezek az osztályozott videoklipek fognak segítséget nyújtani.

Hazugság felismerése

Az alacsony szintű jellemzők Fisher-vektoros kódolása és a videósintű pontszám-vektor négy bináris megtévesztés osztályozó betanítására szolgál. Az osztályozók közül három azokon a vizuális, hallási és szöveges csatornákon alapul, amelyekhez a GMM-ek készültek, a negyedik pedig az összevont pontvektorokat használja a mikro-expressziós detektorokhoz.

Tesztek és eredmények

Az adatbázis 121 bírósági tárgyaláson készült valós videófelvételből állt. Ebből 104 videó lett felhasználva a tesztelés során – 50 „megbízható” és 54 „gyanús”. Ezek a videók szerkesztésre szorultak, hogy a képfeldolgozás során a szoftver csak a fontos részeket lássa. A videókban csak 58 különböző személy szerepel, ami kevesebb, mint a videók száma, tehát egy személy több videóban is szerepel, és hol „megbízható”, hol pedig „gyanús” személyként van értékelve. Ez azt jelenti, hogy amennyiben egy adott személy szerepel a betanítási fázisban, akkor előfordulhat, hogy a tesztelés során az adott személyhez köthető státuszt („megbízható” vagy pedig „gyanús”) fogja automatikusan hozzárendelni a szoftver. Ennek elkerülése érdekében a kísérletek során tízszeres keresztellenőrzés történik.

Mikro-kifejezés meghatározása

Minden egyes videokliphez mintát vesznek 15 képkocka/mp képkockasebességgel. A mozgás elemzése Improved Dense Trajectories módszerrel történik, amelyet Fisher Vektor kódolással ábrázolnak. A mikro-kifejezés felismerőket egy lineáris rendszermag SVM segítségével tanítják be, a LibSVM segítségével.

Konklúzió

Adott egy rendszer a rejtett megtévesztés automatikus felismerésére, amely egy videó kombinált vizuális információit használja. Kimutatták, hogy a megtévesztés a személy identitásától függetlenül megjósolható. A magas szintű és az alacsony szintű vizuális jellemzőket egyaránt használó digitális rendszer lényegesen jobban képes megjósolni a megtévesztést az emberhez képest. Hangalapú és szöveges információ nyújtásával a hazugság előrejelzése tovább javítható.

4.1.2. Facial Micro-Expression Detection in Hi-Speed Video Based on Facial Action Coding System[2]

A projekt egy új, gépi látáson alapuló eljárást javasol a mikro-kifejezések felismerésére és mérésére. Ennek a módszernek a lényege egy olyan leírón alapul, amely egyesíti a maszkok, hisztogramok és a térbeli-időbeli gradiens vektorok előzetes feldolgozását. A bemutatott 3D gradiens hisztogram leíró képes felismerni és mérni az arcbőr felületének gyors és finom változásainak időzítési jellemzőit. Ezt a módszert kifejezetten nagysebességű, 200 kép/mp-es kamerával rögzített videók elemzésére tervezték.

A mikro-kifejezések felismerését és jellemzését számos technikai kérdés nehezíti, ugyanis ezek időtartama 1/3 és 1/25 másodperc között váltakozik, és mindezt egy kis izomösszehúzódnás okozza, amely csak meglehetősen kevés változást idéz elő a bőr textúrájában.

A javasolt megoldás egy „flow approach” eljárás, amelyet kifejezetten 200 kép/mp-es videók elemzésére terveztek. Ez 3D gradienseken (tér-időbeli leírókon) alapul, amelyek a következő képkockák közötti változásokat mérik. A leíró egyesíti a maszkok, hisztogramok és a 3D gradiens vektorok előfeldolgozását, biztosítva ezzel a mikro-kifejezések osztályozását, valamint a mozgás jellemzését. Az időjellemzők közvetlenül a leíró értékekből mérhetők.

Az eljárás lépései

A videóban szereplő arc különböző régiókra történő bontása, majd azok kigyűjtése képkockákként. Az arcrégiókat a Facial Action Coding System (FACS) segítségével határozza meg. Az egyes régiók mozgását egy 3D-gradiens orientációs hisztogram leíróval írják le, amely részleges derivált vektorokon alapul.

Kapcsolódó témák

1. Automatikus arckifejezés felismerő rendszerek

Az általános arckifejezés felismerő rendszerek általában három lépésből állnak: az arc felismerése, az arcról történő információk leolvasása és ábrázolása, és végül az arckifejezés felismerése. Az arcfelismerés az arc automatikus észleléséből és követéséből áll, bizonyos esetekben pedig az arc irányának meghatározása is ennek a lépésnek a része.

Az arcadatok kinyerése és az kifejezés-elemzés ábrázolása szempontjából két fő megközelítés létezik: geometriai jellemzőkön alapuló, valamint megjelenésen alapuló módszerek.

A geometriai jellemzőkön alapuló módszerek az arckifejezéseket az arcjellemző pontokon (például szájzugok, szemzugok, szemöldökök szélei stb.) alapuló idomok hozzávetőleges rekonstrukciójával reprezentálják. Az ezeket a jellemzőpontokat tároló jellemzővektor az arc geometriáját ábrázolja. Például 20 különböző arcvonás nyomon követése részletszűrővel jó eredményeket ad több arckifejezés esetében, azonban néhány finom mozgás, például a mikro-kifejezések, amelyek csak a bőrfelszín változásai által figyelhetők meg, nem mutathatók ki.

A megjelenésen alapuló módszer esetében képszűrőket (pl. ún. „Gaber wavelets”) alkalmaznak az arc egészére vagy egy bizonyos részére, mely segítségével lehetővé válik a

spontán arcmozgás elemzése. Újabban a Gábor-szűrőket felváltották a haar-szerű megoldások, amelyek hasonló felismerési arányt nyújtanak, de kevesebb számítási terhelés mellett.

Mindkét módszer azonban a videó képkockáinként történő elemzésén alapul, anélkül, hogy figyelembe venné a képkockák közötti mozgást. Ezenkívül ennek a megközelítésnek az arcfelület elemzéséhez való alkalmazása nagy adatbázist igényel több mint 600 ezer Gábor-szűrő betanításához.

2. Aktív Megjelenési Modell és Aktív Forma Modell

A közelmúltban a Kanade csoport fejlett kutatási eredményeket közölt az Aktív Megjelenési Modellről (Active Appearance Model, továbbiakban: AAM). Az AAM ötvözi a textúra és az alak elemzését, azonban két hátránya van, ami alkalmatlanná teszi a vizsgált kutatási környezetben:

- I. ehhez a megközelítéshez átfogó adatbázisra van szükség, nagy mennyiségű, kézzel címkézett ponttal az arc területén, valamint
- II. az arckifejezések követésének pontossága jelentősen csökken, ha olyan arcokra alkalmazzák őket, amelyek nem voltak a tanítási készletben.

Hasonló megközelítés az Aktív Forma Modell (Active Shape Model, továbbiakban: ASM), amely egyesíti a ponteloszlási modellt és az egyes pontok helyi megjelenési mintázatát. Bár kevésbé pontos, mint az AAM, robusztusabb eredményt nyújt azokon az arcokon, amelyek nem voltak a képzési készletben. Míg az AAM / ASM bizonyítottan jó teljesítményt mutat a reguláris kifejezések kimutatásában, ahol az arc struktúrája jelentős deformációval rendelkezik, nem alkalmasak finom és alacsony izomösszehúzódnások detektálására, például a mikro-kifejezések során. Példaképpen, a mikro-expressziós mosoly csak az arcok feletti bőrfelületen lévő néhány ránc miatt figyelhető meg.

Azokban az esetekben, amikor a textúraváltozások jelentősebbek, mint az alakváltozások, a flow megközelítések jobb teljesítményt nyújtanak, mint az olyan sablon megközelítések, mint az AAM és az ASM. Emiatt a 3D-leíró megközelítés alkalmasabbnak tekinthető a mikro-kifejezés detektálására.

3D gradiens hisztogram leíró

A 3D gradiens hisztogramok leíróinak használata újszerű megközelítés, melynek során nagysebességű kamerát használnak a mikro-kifejezés detektálására. Jelen esetben a 200 FPS megfelelő kompromisszumot mutat a zaj, a pontosság és a kamera ára között.

Általánosságban a mikro-kifejezések bizonyos arcbőr-textúrák iránybéli deformációi, a kifejezéstelenségtől (semleges állapot) kezdve a kifejezés megjelenítéséig (összehúzódnásig), majd ismét a folyamatosan semleges állapotig. A bőr változásainak meghatározott irányú észlelésével észlelhetjük és azonosíthatjuk a megfelelő kifejezést. Ilyen módon a 3D gradiens hisztogram alkalmas az arcbőr tér-időbeli változásainak leírására. A kiszámított leíró értékek arra szolgálnak, hogy osztályozzák a változtatások AU (Action Unit) címkéit.

A javasolt algoritmus három fő fázist tartalmaz:

1. tizenkét arcrégió kiválasztása,

2. egy előfeldolgozási lépés, amely magában foglalja a régió normalizálását, az informatív pixelek simítását és maszkolását, valamint
3. a 3D gradiens orientáció hisztogram leírójának kiszámítása minden régió esetében

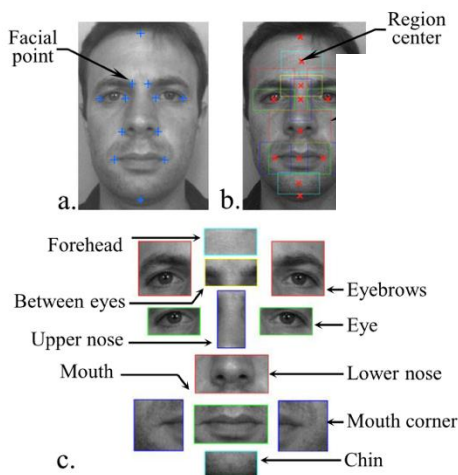
Arcrégiók kiválasztása

Az arc teljes izomzatának és mozgásának egyidejű megbecsülése nehéz és költséges számítási feladat. Ezenkívül, mivel a mikro-kifejezések megjelenése bizonyos arcrégiók megfigyelésével észlelhető, az arc elemzése 12 arcrégió segítségével történik.

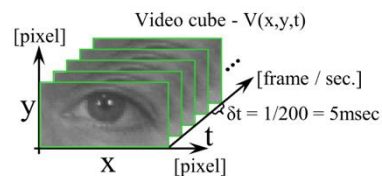
A Facial Action Coding System (FACS) az arckifejezéseket 46 komponens mozgására bontja le (ezek az úgynevezett Action Unit-ok; AU), melyekből kiválasztásra kerülnek a mikro-kifejezések szempontjából legreprezentatívabb arcrégiók.

Annak érdekében, hogy pontosan meg lehessen becsülni az egyes arcizmok mozgását, az egyes arcrégiók úgy kerülnek meghatározásra, hogy megjelenésüket csak korlátozott számú izom befolyásolja. Ezért az osztályozási lépés során minden régió korlátozott számú osztályt fog megkülönböztetni. A régiók kiválasztása manuálisan történik az alábbiak alapján:

- A bemeneti videó első képkockáján 12 arcpont van megjelölve (4.1. ábra / a.).
- A szem átlagos méretét és az egyes régiók középpontját a kiválasztott arcpont pozíciója alapján számítják ki (4.1. ábra / b.).
- Végül a 3D képkockák kiemelése történik a megfelelő arcrégiókból (4.1. ábra / c.). A kocka x, y és t tengelyének egységei „pixelek”, „pixelek” és „képkockák”, amint az a 4.2. ábrán jól láthatóan ábrázolva van.



4.1. ábra: az arc felbontása 12 régióra



4.2. ábra: a szem-régió videókocka kivonata egy 200 FPS videószekvenciából

4.1.3. Összehasonlítás

A hasonló rendszerek összehasonlítása az alábbi szempontok alapján történik. Az egyes szempontok 1–5-ig kerülnek osztályozásra.

	Deception Detection in Videos	Facial Micro-Expression Detection in Hi-Speed Video Based on Facial Action Coding System
Elméleti megközelítés	4	5
Gyakorlati megközelítés	3	5
Tesztelési adatbázis mérete	2	4
Konkrét matematikai megközelítések	3	5
Mikro-kifejezések meghatározásának pontossága	3	5
Elemzett lehetőségek	2	4
Módszertanok meghatározása	3	5
A konkrét megoldás bemutatásának részletessége	3	5
Az eredmények részletessége	2	5
Hivatkozott források hasznossága	3	4
Összesen	28 (56%)*	47 (94%)*

4.1. táblázat: hasonló rendszerek összehasonlítása

*Az értékelések az elemzett rendszerek összehasonlításán alapulnak. Az alacsony értékek nem jelentik feltétlenül azt, hogy a vizsgált rendszer nem ad hasznos információkat a projekt számára, vagy nem segítené elő megfelelő módon a rendszertervezést.

4.2. Motion Profiles for Deception Detection Using Visual Cues[6]

4.2.1. A tanulmány ismertetése

A tanulmány egy adatvezérelt, diszkrét módszert javasol a kihallgatási interjúk során a megtévesztés automatikus felismerésére, kizárólag vizuális jelzések segítségével.

A bőrfolt-elemzés és az Active Shape Modelling együttes használatával folyamatosan nyomon követi és elemezi a kezek és a fej mozgását, valamint az arckifejezések alakulását miközben az alany válaszol a feltett kérdésekre. Ezek segítségével megállapítja a mozgásprofilokat, amelyeket összesít minden választ követően. A végső következtetés az egyes válaszok mozgásprofil-eloszlásának ábrázolásában rejlik.

Ez a séma lehetővé teszi a kikérdezett alanyok viszonylag túlkontrollált és viszonylag agitált viselkedésének ábrázolását, így segítve az igaz és megtévesztő válaszok megkülönböztetését.

4.2.2. Fej- és kézfoltkövetés

A fej- és kézfoltok helyzetének, méretének és tájolásának nyomon követésére színanalízist, sajátalak-felbontást és Kalman-szűrést alkalmaz. A 3D Look-Up-Table (LUT) helyett 2D LUT-t épít fel színárnyalat és telítettség színkomponensekkel, az arc és a kezek Hue-Saturation-Value (HSV) bőrszín-eloszlása alapján. A HSV-reprezentáció Value komponensét nem használja, hogy az ábrázolás robusztusabb legyen a megvilágításra. A LUT offline bőrszínmintákból épül fel. A rendszer meghatározza az arc- és kézszerű régiókat minden egyes képkockán a LUT segítségével, majd megjelöli az elliptikus fej- és kézfoltokra hasonlítókat. Az ezt követő sajátalak-bontás és a Kálmán-szűrés a megjelölt foltok közül folyamatosan kiszűri a legvalószínűbbeket azok alakja és korábbi pozíciója alapján. Az eredmény a 4.3. ábrán látható.

A foltok nyomon követett pozícióiból származtatott jellemzőket számít ki, amelyek célja a viselkedési izgatottság és a túlzott kontroll rögzítése a kezek relatív helyzetének, testtartási eltolódásainak és az alany testtartási nyitottságának jellemzésével.



4.3. ábra: az arc, valamint a kéz és a fej formájának lekövetése

4.2.3. Arckövetés

Az arckövetés nagy kihívást jelent, mivel az arcfelismerőnek jól kell általánosítania a korábban nem látott arcokra, és kezelnie kell a megvilágítás változásait. Ezenkívül meg kell

birkózni az olyan változásokkal, mint például a fej elforgatása, amelyek drasztikus változásokat okoznak az arc alakjában, mivel az arc egy nem lineáris síkon fekszik.

A problémát Active Shape Modell (ASM) segítségével kezeli, amely az arc alakváltozásainak statisztikai modellje. Az ASM keretrendszerben egy S arcformát N arcpont képvisel, amelyek mindegyikét (x,y) képkoordináták jellemzik, tehát $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$. A főkomponens-elemzés (PCA) alkalmazásával az arcformák összehangolt tanító halmazán egy olyan altér kerül megtanulásra, amely rögzíti az alakváltoztatás főbb módjait azáltal, hogy az alakzatok kovarianciamátrixának legmagasabb sajátértékű sajátvektorai mentén kivetíti az alakzatokat.

A szerzők ezenkívül egy darabonkénti közelítést javasolnak a nemlineáris alaksokaságra, átfedő lineáris alterek felhasználásával. Ez alapvetően azt jelenti, hogy minden egyes altérhez külön ASM-modelleket kell megtanulni, és dinamikusan váltani az altereket, ahogy a követett arc pozíciója például egy forgatás során változik. Rendszerüket valós idejű működésre hozták létre azáltal, hogy beépítették a Sum of Squared Intensity Differences (SSID) pontkövetőt, hogy nyomon kövessék a képfoltokat az egymást követő képkockákon, kis elmozdulásokat feltételezve. Ezenkívül a Bayesian Mixture of Experts módszer segítségével meg tudják becsülni a fej 3D-s pozícióját a nyomon követett arcpontok alapján. Az 4.4. ábra egy mintaképet mutat be a 79 nyomon követett arcponttal, valamint a megjósolt 3D-s fejtartással.



4.4. ábra: az arc 79 jellemző pontja, valamint ez ezek alapján megbecsült fejtartás értéke látható

Ezzel a módszerrel nyomon tudták követni a fejtartást és az arcpontok (x,y) pozícióit az elemzett videószekvenciák minden képkockájában. A nyomon követett arcpontokat az arc mikro-kifejezéseinek és aszimmetriáinak rögzítésére tervezett származtatott jellemzők kiszámítására használták. Nevezetesen ezek a következők: a száj sarokpontjai közötti szög változása, a száj középpontja és minden sarokpontja közötti szög változása, a száj területének változása, a belső és külső bal és jobb szemöldök elmozdulása. A bal/jobbszájsarok pontokat a három bal/jobbszájsarok iránypont átlagaként számítják ki. A bal/jobbszájsarok szöge az a szög, amelyet a bal/jobbszájsarok és a két oldalán lévő két arcpont alkot. Végül a belső/külső szemöldök eltolódását a négy legbelső/külső szemöldök iránypont átlagos elmozdulásának felhasználásával számítják ki. Ebből az elmozdulásból levonják a hat alsó orrcsont átlagos elmozdulását, hogy észrevegyék a fej elmozdulását, feltételezve, hogy az orr a legstabilabb arckomponens.

Az alanyok követett mozgásainak és kifejezéseinek összegzéséhez mozgásprofilok kinyerését javasolják, melyek tartalmazzak arckifejezési információkat. Ezenkívül a mozgásprofilok naplózásra vannak tervezve, hogy rögzítsék a megtévesztés észleléséhez fontos információkat, nevezetesen a csekély vagy semmilyen mértékű, valamint az extrém mozgást. Az egyes alanyok válaszában a képkockák többsége kis mennyiségű mozgást tartalmaz, tehát az alanyok ritkán tesznek szélsőséges mozdulatokat válaszuk teljes időtartama alatt.

Minden mozgás öt hisztogram csoportba van osztva, amelyek mindegyike exponenciálisan növekszik. Ezért az első csoport nagyon kis tartományt fed le (ami kevés vagy semmilyen mozgásnak felel meg), az ötödik pedig a legnagyobb tartományt (ami minden szélsőséges mozgásnak felel meg). Az adatoknak ez az új reprezentációja sikeresen képes elkülöníteni a túlkontrollált és agitációs válaszokat.

4.2.4. Tesztelések

A tesztek során az interjúalanyok különböző kérdésekre 13 egyszavas, valamint 13 hosszabb választ adtak, amelyeket képzett kérdezőbiztosok tettek fel. Az interjúalanyok bizonyos kérdésekre őszintén, másokra megtévesztően válaszoltak egy előre meghatározott sorrendben. Az emberek egyik felének előbb az őszintén megválaszolandó, a másik felének pedig előbb a megtévesztéssel megválaszolandó kérdéseket raktak fel. Az interjú során három nagysebességű digitális kamera rögzítette a kinetikus viselkedést: az egyik teljes testprofil nézetet, egy másik teljes test előlnézetet és egy harmadik frontális arcképet rögzített. Minden három kérdésből álló blokk után az interjúalanyok egy 0-tól (egyáltalán nem) 10-ig (teljesen igaz) skálán értékelték, hogy az egyes kérdésekre adott válaszuk igaz volt-e. Az interjúk általában 20-30 percesek voltak.

4.2.5. Értékelés

Minden videóinterjút elemeztek, és minden egyes képkockából kivonták a jellemző reakciókat. A teljes test elülső nézetét a foltkövető, az arc elülső nézetét pedig az ASM arckövető elemezte. Jellemzőcsatornánként 5 hisztogrammezőt használtak, egyenletes naplózónával. Ily módon az első két csoport elég széles volt ahhoz, hogy rögzítse a túlzottan szabályozott viselkedésnek megfelelő nagyon kis jellemzőértékeket, míg a többi csoport szélességét egymás után megnövelték, hogy rögzítsék az egyre nagyobb mozgásokat, amelyek megfelelnek a nyugodt és izgatott viselkedések, illetőleg 147 különálló Nearest Neighbour modellt építettek fel (egyet a 147 alany mindegyikéhez) a Leave One Out Cross Validation (LOOCV) segítségével, ahol minden interjúválaszhoz hozzárendelésre került egy, hogy teszteljék és betanítsák a modellt. A mozgásprofil NN modelljeik 81,6%-os pontosságot értek el. A különböző alanyok eltérő megtévesztő viselkedéssel és eltérő alapszintű viselkedéssel rendelkezhetnek, szemben azzal, hogy létezik egy univerzális megtévesztési jelzés vagy „küszöb”, amely mindenre érvényes, és megkülönbözteti az igazságot a megtévesztéstől.

A rendszer hátránya, hogy tanítási adatokat kell gyűjteni egy tesztalanyról a modell betanításához. Az ilyen képzési adatok beszerzése nem feltétlenül triviális olyan helyzetekben, amikor egy ilyen rendszer hasznos lehet.

4.3. Multimodal Deception Detection using Automatically Extracted Acoustic, Visual, and Lexical Features[7]

4.3.1. A tanulmány ismertetése

A megtévesztés felismerése a társalgási párbeszédben az elmúlt években nagy figyelmet kapott. Ennek ellenére a létező módszerek nagymértékben támaszkodnak az ember által megjelölt annotációkra, amelyek költségesek és potenciálisan pontatlanok. Ebben a munkában egy olyan automatizált rendszer kerül bemutatásra, amely multimodális funkciókat használ a társalgási megtévesztés észlelésére, emberi megjegyzések használata nélkül. Tanulmányozza a különböző modalitások előre jelző erejét, és kombinálja őket a jobb teljesítmény érdekében. Az openSMILE-t használja az akusztikus jellemzők kinyerésére, miután zajcsökkentési technikákat alkalmazott az eredeti hangon. Az arc jellegzetességei a vizuális modalitásból származnak. Kísérleteket végez az arckifejezés-detektorok betanításával és a Fisher vektorok alkalmazásával az arc változó hosszúságú arcpont-sorozatának kódolására.

4.3.2. The Box of Lies gyűjtemény

A Box of Lies adatkészletet használja a megtévesztés észlelésének multimodális megközelítéséhez. Ez a gyűjtemény a Tonight Show házigazdája, Jimmy Fallon és a „Box of Lies” nevű játékot játszó 28 vendég közötti videós beszélgetéseket tartalmazza.

Az egyik játékos valósághű leírást ad egy tárgyról, amelyet a dobozban lát, vagy megtévesztő leírást egy olyan kitalált tárgyról, amely nincs a dobozban. A másik játékos kitalálja, hogy a leírás igaz-e vagy sem. Ez a beszélgetési beállítás nagymértékben megnehezíti az igazság vagy a hazugság előrejelzését, mivel a résztvevők közötti interakciót nehéz megragadni. Ezenkívül minden játékos híresség, akik állítólag jobb színészek, mint a normál emberek, ami még nagyobb kihívást jelent a megtévesztés felismerésében. A játéknak 68 rögzített fordulója van (39 igazság és 29 hazugság) és összesen 1049 kijelentés (862 hazugság és 187 igazság). Minden kör egy új dobozt vezet be egy új tárggyal; ugyanaz a játékpár több kört is játszhat. Minden körnek egyetlen címkéje van: „Igazság” vagy „Hazugság”. Egyetlen „Hazugság” címkével ellátott körön belül azonban lehetséges, hogy néhány kijelentés valóban igaz volt, így a megnyilatkozások annotációi pontosabbak. A helyes megnyilatkozási szintű címkéket az adathalmaz gyűjtői biztosították. A gyűjtemény kiegyensúlyozatlan, az igaz és megtévesztő állítások aránya 1:4,6.

Ebben az adathalmazban kulcsfontosságú kihívások vannak az automatikus reakció-kivonásban: Az akusztikus jellemzők tekintetében a felvételeken jelentős háttérzaj van, beleértve a közönség nevetését és tapsát, valamint a háttérzenét. A vizuális jellemzők esetében a kamera szöge gyakran elmozdul, ami változást okoz a résztvevők arcának a beesési szögében a különböző képkockákban. Előfordul az is, hogy két játékos gyakran egyszerre beszél, ami megnehezíti a nyelvi elemzéshez használható kiváló minőségű automatikus beszédfelismerő (ASR) átiratok beszerzését.

4.3.3. Metodológia

A projekt három modalitás összesített vizsgálatával állapítja meg a megtévesztésre utaló jeleket: akusztikus, vizuális és lexikális. A következőkben a projekt szempontjából lényeges, vizuális jellemzők kerülnek bemutatásra.

A vizuális jellemzőket korábban már vizsgálták a megtévesztés észlelésére, de ezek gyakran fáradságos emberi megjegyzéseken alapulnak. Ebben a munkában három megközelítést vizsgáltak a vizuális jellemzők automatikus kinyerésére: (1) Fisher Vektor (FV) kódolás, (2) Lineárisan Aggregált Leírók (VLAD) kódolása és (3) Arckifejezés-érzékelés (FED). Először a dlib segítségével kinyerték az arcpontokat a videóközből. Ugyanazon személy megjelenő arcainak párosításához és egy esetlegesen megjelenő, zavaró személy kizárásához arcbeágyazást (az arc 128-dimenziós vektoros ábrázolása) használtak a két arc közötti távolság mérésére, koszinusz hasonlóság segítségével. Annak érdekében, hogy kiküszöböljék a különböző pozíciókban és különböző szögekben elhelyezkedő arcok hatását, normalizálták az egyes észlelt X arcokat. A leíró arcainak azonosítása és a normalizálás alkalmazása után a következő módszereket vizsgálták meg a vizuális információ kódolására a megtévesztés észleléséhez.

1. Fisher Vektor kódolás

A számítógépes látásban Fisher-vektort (FV) használnak a kép ábrázolására az osztályozáshoz. Egy Gaussian Mixture Model (GMM) modellezi a képből kinyert vizuális jellemzők eloszlását, majd a Fisher Vektor kódolja a GMM alatti jellemzők log-likelihood gradienseit a GMM paramétereikhez képest, így fix hosszúságú vektoros reprezentációkat hozva létre. Kimutatták, hogy az FV-k jobb teljesítményt nyújtanak a képosztályozási feladatokban a hagyományos Bag-of-Visual reprezentációknál.

2. VLAD kódolás

A VLAD egy másik számítógépes látási megközelítés, amelyet a képek sűrű vektoros ábrázolására használnak. Az FV-hez hasonlóan a VLAD kódolás különböző hosszúságú jellemzőket is leképez ugyanarra a látens térre. Az alapvető különbség az FV és a VLAD között az, hogy a VLAD K-Means klaszterezést használ a GMM helyett, és csak elsőrendű információkat tárol.

3. Facial Expression Detection (FED)

A képek a fenti két, nem felügyelt ábrázolásán kívül egy felügyelt megközelítést is megvizsgáltak a jellemzők kinyerésére: az arckifejezés észlelését. Az adathalmazban az arckifejezések 7 kategóriája van annotálva, a MUMIN multimodális kódolási séma alapján. Minden kategóriának több viselkedési értéke van, például a szemöldök lehet semleges, felfelé ívelő, összeráncolt vagy egyéb. Ezeket a kifejezéseket képzett annotátorok gondosan megjelölték az eredeti gyűjteményben.

Az egyéni arckifejezés-detektorokat Random Forest osztályozással tanították be, jellemzőkként az automatikusan azonosított arcpont-koordinátákat használva. Mind a 7

kategória esetében kifejlesztettek egy több osztályú arckifejezés osztályozót, és egy kibővített tesztkészlet segítségével értékelték a teljesítményt.

A fejmozgásokat besorolása bizonyult a legnehezebb feladatnak, és ezekben volt a legtöbb a lehetséges osztályok száma (14). Ezek az osztályozási eredmények igen ígéretesek, különösen annak a kihívásnak a fényében, hogy a kamera szöge folyamatosan változik, és így az adott alany arcát egészen más szögekből rögzítik. Figyelembe kell venni azt is, hogy az osztályozó paraméterei nem a feladathoz lettek hangolva a túlillesztés elkerülése érdekében. Az arckifejezés kimutatásának egyik kihívása ezekben az adatokban az adatosztályok súlyos egyensúlytalansága. Például a száj-ajkak osztályozásánál a „Corners down” osztálynak csak 305 példája van, míg más osztályoknak, mint például a „Retracted”, körülbelül 3000 példája van.

4.3.4. Tesztek és eredmények

Az adatok kiegyensúlyozatlanok az igazság és a hazugság osztályai között. A játékkörök 57,35%-a volt megtévesztő, és az összes elhangzott kijelentés 82,17%-a hazugságnak minősült. Ennek az az oka, hogy a játékosok választhattak, hogy hazudnak vagy igazat mondanak, és a legtöbb játékos a megtévesztés mellett döntött. Az osztály-kiegyensúlyozatlanság problémájának megoldása érdekében alul-mintavételezést alkalmaztak a hazugságnak nevezett adatoknál. Kísérleteztek olyan felül-mintavételezési stratégiákkal is, mint a Synthetic Minority Over-sampling Technique (SMOTE), de úgy találták, hogy ez rontotta a Truth osztály teljesítményét. Mivel az adatkészletben alul-mintavételt alkalmaztak, a véletlenszerű osztályozó 50%-os valószínűséggel adja meg a helyes előrejelzést.

A megtévesztés osztályozóit betanították a fentebb leírt, automatikusan kivont vizuális jellemzők segítségével. Mivel az adatkészletben az arckifejezések kézi megjegyzései is vannak, összehasonlították az automatikusan és a manuálisan kinyert jellemzők osztályozási eredményeit. A legjobb vizuális megtévesztés osztályozási teljesítményt (67%-os pontosság) az FV kódolással érték el. Ellentétben az arckifejezés-detektorokkal, amelyek a képzés során emberi megjegyzésekre támaszkodtak, az FV-kódolás és a VLAD-kódolás nem igényel emberi megjegyzéseket, így ez az eredmény fontos és hasznos a jövőbeli kutatások számára, ami arra utal, hogy az automatikus osztályozás ebben az esetben megbízhatóbb lehet, mint az emberi annotáció. Az FV-kódolás teljesítménye valamivel jobb, mint a VLAD-kódolásé, valószínűleg az FV-k másodrendű statisztikáiból származó további információk miatt.

Miután minden egyes modalitáshoz külön-külön kísérleteztek az automatikusan kinyert funkciókkal, olyan multimodális osztályozókat képeztek ki, amelyek egyesítik az egyes modalitásokhoz legjobban teljesítő jellemzőket.

Összehasonlították a multimodális megtévesztési modelljeiket a korábbi manuális funkciók eredményeivel. Az akusztikus, vizuális és nyelvi modalitások kombinálásával a legjobb, 73%-os pontosságú megtévesztés besorolási teljesítmény érhető el. Ez 8%-os abszolút javulás a kézi funkciókkal elért legjobb multimodális teljesítményhez (65%) képest. Amellett, hogy a kijelentéseket megtévesztőnek vagy igaznak minősítették, a módszert az egész játékkörön is értékelték.

4.4. Human Behavior and Deception Detection[8]

4.4.1. Tudományos áttekintés – a hazugság viselkedésben megnyilvánuló jelei

A mai napig egyetlen kutató sem dokumentált „Pinokkió választ”, vagyis egy viselkedést vagy viselkedésmintát, amely minden emberben, minden helyzetben egyértelműen hazugságra utalna. A kutatók által eddig azonosított és megvizsgált magatartásformák mindegyike a megtévesztéstől független okokból is bekövetkezhet. Általánosságban elmondható, hogy a hazugságok viselkedésből való felismerésére irányuló kutatások azt sugallják, hogy a viselkedési nyomok két széles csoportja valószínűsíthető, ha valaki hazudik: a hazug emlékezetéhez és a mondandójának megformálásához kapcsolódó, kognitív nyomok, a hazug érzelmi háttérével kapcsolatos nyomok, valamint a megtévesztéssel kapcsolatos érzések, az érzelmi nyomok.

Kognitív nyomok

A hazugság információkat rejt, gyárt vagy torz el; ez egy bizonyos mértékű mentális erőfeszítéssel jár. A hazugnak erősebben kell gondolkodnia, mint egy igazmondónak. Annak érdekében, hogy elterelje a gyanút, olyan eseményeket kell alkotnia, amelyek meg nem történtek, vagy olyan módon kell azokat leírnia, hogy lehetővé tegye ezáltal a többféle értelmezést. Ez azonban nem csak a nyíltan hazudni akarás miatt történhet, mert előfordulhat az életben olyan szituáció is, amikor a másik fél számára egy fájó, kellemetlen hírt kell közölni, és mindezt úgy, hogy a másik félben az igazság által keltett negatív érzelmek súlyát csökkenteni lehessen. A hazugság során fellépő extra mentális erőfeszítés általában hosszabb beszédlátenciával, megnövekedett beszédzavarokkal, kevésbé valószínű tartalommal, kevesebb verbális és vokális érintettséggel, kevesebb beszédd idővel, több ismételt szó és kifejezés használatában nyilvánul meg, és így tovább. A kutatások azt is kimutatták, hogy bizonyos nonverbális viselkedésmódok megváltoznak ennek a mentális erőfeszítésnek az eredményeként. Például a különböző gesztikulációk – a beszédet kísérő kéz- vagy fejmozdulatok, amelyeket sokan a beszéd részének tekintenek – csökkenni fognak hazugság esetén.

Egy másik módja annak, hogy a kogníció részt vegyen a hazugságban, az a naturalista memóriajellemzők azonosítása. A ténylegesen meg nem tapasztalt események több ambivalenciát mutatnak, kevesebb részlettel, gyengébb logikai szerkezettel: kevésbé valószínűsíthetőek, több negatív kijelentést tartalmaznak, és kevésbé vannak beágyazva a kontextusba. A hazudozók kevésbé valószínű, hogy beismerik emlékezethiányukat, kevésbé spontán korrekciójuk van, és több negatív érzelemre támaszkodó szót, valamint kevesebb ön és egyéb hivatkozást használnak. Úgy tűnik, hogy a mentális erőfeszítésekre utaló jelek inkább a hazugság átadásakor fordulnak elő, míg a memóriafelidézési nyomok inkább a hazugság tartalmában fedezhetők fel.

Fontos megjegyezni, hogy nem minden hazugság jár automatikusan szellemi erőfeszítéssel. Sokkal kevésbé megterhelő lelkileg például egy olyan zárt végű kérdés igennel vagy nemmel

történő megválaszolása, mint például: „Te pakoltál be a csomagodba?”, mint válaszolni egy nyílt végű „Mit szándékozik tenni az utazásán?” kérdésre. Sőt, egy felkészült hazudozó meggyőzőbbnek tűnhet, ha egy ténylegesen átélt eseményt helyettesít alibiként, nem pedig egy teljesen újat. Ez lehet az oka annak, hogy egy közelmúltban megjelent általános áttekintő tanulmány következetes nem homogén hatásméreteket talált ezekre a szellemi erőfeszítésekre és memóriaalapú jelzésekre az áttekintett tanulmányok során.

Érzelmi nyomok

A hazugság érzelmeket is generálhat, a mások – jellemzően hátsó szándékok által vezérelt – átveréséből fakadó izgalomtól kezdve, a lebukástól való félelemtől át egészen a büntudatig. Darwin úgy tartotta, hogy az érzelmek hajlamosak az arckifejezésekben, valamint a hangtónusokban megnyilvánulni, és ezek elég megbízhatóak lehetnek az érzelmi állapotok pontos azonosításához. A kutatások azóta kimutatták, hogy egyes kifejezésekre – pl. harag, megvetés, undor, félelem, boldogság, szomorúság/szorongás vagy meglepetés – a kultúrák szerte a bolygón felismerik és kifejezik ezeket az érzelmeket mind arcon, mind hangszínen.

Mivel hazugság fennállása esetén valós veszélyt jelenthetnek a lebukással járó következmények, ezért arra következtethetünk, hogy a hazugoknál többet fogunk látni az érzelmekre utaló jelekből, mint az igazmondóknál. Ha a hazugság úgynevezett „kegyes hazugság”, amelyet az emberek gyakran és erőfeszítés nélkül mondanak, akkor általában kevesebb érzelm van bennük. A metaanalitikai tanulmányok azt sugallják, hogy a hazudozók idegesebbnek tűnnek, mint az igazmondók, kevésbé kellemes arccal, magasabb hangfeszültséggel, magasabb hangmagassággal, nagyobb pupillatágulattal és izgatottsággal. Ha maga a hazugság érzelmekről szól – pl. elmondani valakinek, hogy nyugodtnak érzi magát, pedig valójában ideges – a kutatás azt mutatja, hogy a valóban érzett érzelmek jelei megjelennek az arcon és a hangon az eltitkolási kísérletek ellenére is, bár ezek a jelek gyakran finomak és rövidiek.

4.4.2. Tudományos áttekintés – képességek a hazugság észleléséhez

Az elmúlt 30 év kutatásai azt sugallják, hogy az átlagember statisztikailag valamivel jobb a megtévesztés felismerésében, de a gyakorlat nem ezt igazolja. Több mint 100 friss tanulmány kimutatta, hogy ha a megtévesztés véletlenszerűsége 50%-os, az átlagember körülbelül 54%-os pontossággal ismeri azt fel. Ennek a gyenge képességnek számos oka van; köztük rossz visszajelzések a mindennapi életben (azaz az ember csak az éppen elkapott hazugságokról tud); az emberek általános hajlama a hiszékenységre valamivel kapcsolatban, egészen addig, amíg az ellenkezőjét be nem bizonyítják (az „igazság elfogultsága”), és különösen annak hibás megértése, hogy hogyan is néznek ki a hazugok (azaz az emberek hazugságra utaló jelei közötti különbség a ténylegeshez képest nyomok).

4.5. Deception detection: State of the art and future prospects[9]

4.5.1. A hazugság felismerése

A tény, hogy a megtévesztés viselkedésbeli jeleinek csekély diagnosztikai értéke van, a megtévesztés kutatásában elmozduláshoz vezetett. Sok kutatót már nem érdekel a hazug által spontán módon megjelenített, megfoghatatlan megtévesztésre utaló jelek üldözése – az ilyen jelzések gyengék és ingatagok.

Ehelyett a kutatókat olyan interjústratégiák megtervezése felé orientálódnak, amelyek az igazmondók és a hazudozók közötti viselkedésbeli különbségekre irányulnak.

Az elmúlt években hatalmas munka folyt e cél érdekében, főként Vrij (Egyesült Királyság) és Granhag (Svédország) laboratóriumaiban. Ennek a kutatásnak a középpontjában az állt, hogy a bűnüldöző szervek számára olyan hazugságfelderítő eszközöket biztosítson, amelyeket a bűncselekményekkel gyanúsítottak kihallgatásakor használhatnak. Ebben az új perspektívában a hazugságvizsgáló már nem megfigyelőként jelenik meg, aki figyelmesen, de passzívan vizsgálja a küldő viselkedését, hanem mint kérdezőbiztos, aki aktívan alkalmaz sajátos kérdezési megközelítéseket, amelyek felfedhetik a megtévesztést.

Ezeknek az interjúknak az igazmondók és a hazugok közötti pszichológiai különbségeken kell alapulniuk. Például Granhag, Hartwig, Mac Giolla és Clemens (2015) azzal érvel, hogy a bűnös gyanúsítottak (hazugok) nem hajlandók információt adni a rendőrségnek, mert ez leleplezheti őket. Ebből kifolyólag olyan elkerülési stratégiákat alkalmaznak, mint például, egy megingathatatlan tűnő alibi, amikkel kapcsolatban csak homályos részleteket adnak meg, vagy eleve olyan adatokat adnak meg, amelyeket a rendőrség nem tud ellenőrizni.

Amikor a bizonyítékokkal szembesülnek, a hazudozók azt a menekülési stratégiát alkalmazzák, hogy tagadják magát a bizonyítékot. Ezzel szemben az ártatlan gyanúsítottak (igazságmondók) szívesen adnak információkat, mert úgy érzik, ez segíthet ártatlanságuk megállapításában. Amikor valamilyen terhelő bizonyítékkal szembesülnek, az igazmondók általában hajlandóak beismerni, mert úgy gondolják, hogy „az igazság ki fog derülni”, és ha nem követnek el bűncselekményt, nem ítélik el őket, mert az emberek azt kapják, amit megérdemelnek.

4.5.2. Ellenőrizhetőségi megközelítés

A bűnös gyanúsítottak hajlamát az információk visszatartására az ellenőrizhetőségi megközelítésben is kihasználják. A bűnös gyanúsítottak, akik hazudnak alibijükéről, különösen, ha a kérdező kifejezetten megkéri őket, hogy legyenek nagyon részletesek, úgy érezhetik, hogy ha kevés részletet adnak meg, azzal megtévesztőnek tűnhetnek. Ha azonban sok adatot közölnek, a rendőrség ellenőrizheti ezeket, és kiderítheti, hogy az alibi hamis. A hazugok ellenőrizhetetlen adatok megadásával oldhatják meg ezt a dilemmát.

Ezzel szemben az igazmondók több ellenőrizhető részlettel szolgálnak, mint a hazugok. Ilyen ellenőrizhető részletek például a rendőrség által kihallgatott más személyekkel vagy azok jelenlétében végzett tevékenységek leírását, valamint olyan helyszínekre, területekre való utalást, ahol a gyanúsított úgy véli, hogy vannak térfigyelő kamerák. Ez magában foglalja azt

is, hogy beszámolnak arról, ha olyan tevékenységeket végeztek, amelyeket rendszeresen elektronikusan rögzítenek (pl. hitelkártya használata). A kutatások alátámasztották azt az elképzelést, hogy a hazugok kevesebb ellenőrizhető részletet szolgáltatnak, mint az igazmondók. Érdekes módon ez a megközelítés immunis az ellenintézkedésekre; Amíg a hazugok tudatában is vannak annak, hogy ellenőrizhető részleteket kell közölniük, addig viszont csak az igazmondók közölnek ilyeneket. Egy tanulmány kimutatta, hogy ha a gyanúsítottakat arra utasították, hogy igazolható részleteket adjanak meg, az az igazmondók körében megnövekedett, de a hazugok körében nem. Így az a kifejezett kérés, hogy ellenőrizhető részletek szerepeljenek a beszámolóban, így növelve a különbséget a hazudozók és az igazmondók között, ezzel fokozva ennek a technikának a megkülönböztető erejét.

4.5.3. A megtévesztés pszichofiziológiai kimutatása

Ősidők óta törekednek arra az emberek, hogy a gyanúsított fiziológiai reakcióiból kimutassák a megtévesztést. Trovillo (1939) elmagyarázza, hogy a görög orvos, Erasztratosz (Kr. e. 300-250) hogyan tudott rájönni, hogy Antiochus szíriai herceg titokban szerelmes volt fiatal mostohaanyjába, Stratonice-be: megfigyelte a pulzusát.

Néhány régi hazugságérzékelési módszer azon a feltételezésen alapult, hogy a hazugság félelmet vált ki, és ez egy bizonyos fiziológia szintre emelte a korai vizsgálatokat. Például az ókori Kínában és Indiában a bűnözés gyanúsítottjainak rizst kellett rágniuk; ha nem tudták kiköpní, bűnösnek számítottak. Ez a megpróbáltatás azt a megfigyelést tükrözi, hogy a nagy stressz csökkenti a nyáleválasztást.

A pszichofiziológiai hazugságvizsgálat az 1920-as években kapott nagyobb figyelmet. Abban az időben az Egyesült Államokban a rendőrség professzionalizálódása felé irányuló reformok légköre uralkodott, amely a tudományos módszerek és eljárások bűnüldöző szervek általi elfogadását jelentette.

Ebben az összefüggésben Marston, Larson és Keeler újításokat vezetett be a gyanúsított szívfrekvenciájának, vérnyomásának, légzésének és a bőr vezető képességének rögzítésére a kihallgatás során, hogy felmérjék a gyanúsítottat. Ez volt a poligrafikus hazugságvizsgálat kezdete. Újabban az elektroencefalográfiát és a funkcionális mágneses rezonancia képalkotást (fMRI) is tesztelték hazugságvizsgálati eljárásként.

4.6. Összegzés

Áttekintve a hasonló rendszereket, azok működésének elveit és a megvalósítás szempontjait, valamint figyelembe véve a kapcsolódó pszichológiai tanulmányokat, megállapítható, hogy a projekt megvalósítása nem ütközik különösebb akadályba. A kutatómunkám során talált dokumentációknak és leírásoknak a segítségével nagyjából behatárolható az irány, amibe az esetlegesen felmerülő problémák esetén elindulni célszerű.

4.6.1. A videó rögzítésének elvárásai

Az első lépés a vizsgálandó videó rögzítése. A videó elkészítésének módja, körülményei nagyon fontos tényezők. Az alábbi felsorolásban található a videó elkészítésének legfontosabb szempontjai:

- A fényviszonyok legyenek megfelelőek. Fontos, hogy az arc árnyékolás nélkül kellően meg legyen világítva.
- A kép lehetőleg a vizsgálandó személy arcával legyen minél inkább kitöltve. Minél kevesebb felesleges, zavaró tényező van a háttérben, annál jobban elősegíthető a feldolgozás optimalizálása, és minél jobban látható a vizsgálandó arc, annál pontosabb eredményt kaphatunk.
- A vizsgálandó személy nézzen egyenesen a kamerába, és tartsa a pozícióját. Az arc a válaszadás közben ne forduljon el a kamera felől, mindig egyenesen a kamerába nézzen.
- Fontos a nyílt végű kérdések alkalmazása, amelyekre nem feltétlenül lehet egyszerű igennel vagy nemmel válaszolni. Azokra a kérdésekre, amelyekre igennel vagy nemmel lehet válaszolni, szintén egész mondatban történő válaszokat legyenek elvárva. Például, ha a kérdés az, hogy „Te vagy Gipsz Jakab?“, akkor a helyes válasz valami ilyesmi: „Igen, az én nevem Gipsz Jakab”. A lényeg, hogy lehetőleg kerüljük az egyszavas, érzelemmentes válaszokat. Ugyanakkor ügyeljünk arra, hogy a válaszadásnak nem szükséges tovább tartania néhány másodpercnél.
- Mivel az arcon sok mikro-kifejezés csak rövid ideig jelenik meg, némelyik az emberi szem számára nem vagy csak nehezen észlelhető, ezért fontos a sűrű mintavétel. Minél nagyobb a videó felbontása és minél több képkockát rögzítünk másodpercenként, annál pontosabb eredményeket kaphatunk. A minimális ajánlott FPS-érték 30 körül van.

4.6.2. A kérdések feltételének módja

Minden ember különböző, ezért nagy valószínűséggel egy adott szituációra mindenki más-más reakciókat adhat, jelen esetben például különböző gyakorisággal és intenzitással különböző mikro-kifejezéseket jeleníthet meg. Ezért fontos, hogy minden egyes vizsgált személynél normalizáljuk a reakciókat.

A poligráfes tesztekhez hasonlóan itt is szükség van bizonyos kontrollkérdésekre ahhoz, hogy megkapjuk a vizsgálandó személy normál körülmények közötti alapértékeit. Emiatt a kérdéseket a következőképpen célszerű feltenni:

- Először a kontroll kérdéseket kell feltenni. Ezek mind olyan kérdések, amelyekre őszinte és igaz válaszokat várunk. A legcélravezetőbb, ha a kérdező is tisztában van a válaszok hitelességével. Több kontroll kérdés jobban hozzájárul a vizsgálandó személy alapreakcióinak a felméréséhez. Legalább öt ilyen kérdés feltétele javasolt egy tesztelés során.
- Az ellenőrző kérdések után jöhetnek azok a kérdések, amelyekre az adott válasz őszinteségét ellenőrizni szeretnénk.

4.6.3. Videó előkészítése a feldolgozásra

Még a legprofesszionálisabban rögzített videókban is vannak olyan részek, amelyekre nincs szükség az elemzéshez. Fontos, hogy az elemzéshez kizárólag az alany éppen az aktuális kérdésre adott reakcióira van szükség. Minden más, ami nem tartalmaz egy közvetlen reakciót egy adott kérdésekre, lényegtelen a vizsgálat szempontjából.

Nem szükséges, sőt, kifejezetten félrevezető eredményeket produkálhat, ha azok a részek is az elemzés részét alkotnák, amikor például az alany csak hallgat a kérdésre, és közben változik az arckifejezése, vagy ha bármilyen zavaró tényező előfordul két válaszadás között – például egy tüsszentés –, ami bármilyen módon megzavarja a vizsgálandó személyt.

A fentiek alapján az elemzés megkezdése előtt meg kell jelölnünk a videóban azokat az időintervallumokat, amelyekben az alany közvetlenül reagál egy kérdésre. Nagyon fontos az intervallumok minél pontosabb megadása, ugyanis itt akár a tizedmásodpercek is sok mindent befolyásolhatnak. Természetesen a kontroll kérdésekre adott reakciókat külön kell megjelölni a vizsgálandó kérdésekre adott reakcióktól.

4.6.4. Releváns Action Unitok

Míg az egyszerűbb érzelmek (öröm, szomorúság, zavarodottság stb.) egyértelműen leírhatók Action Unitok bizonyos kombinációival, a hazugság sajnos más eset. Az irodalomkutatás során azonban kiderült, hogy mely AU-k jelennek meg nagyobb valószínűséggel a hazugság során. Ezek számai a következők: 1, 2, 4, 5, 6, 7, 10, 12, 14, 23 és 28. A vizsgálat során ezek lesznek a szignifikáns Action Unitok. Fontos azonban megjegyezni, hogy a tény, miszerint ezek az AU-k jellemzőbbek a hazugság során, nem feltétlenül jelenti azt, hogy jelenlétük egyértelműen hazugságra utal.

AU	Description	Facial muscle	Example
1	Inner Brow Raiser	<i>Frontalis, pars medialis</i>	
2	Outer Brow Raiser	<i>Frontalis, pars lateralis</i>	
4	Brow Lowerer	<i>Corrugator et Depressor supercilii</i>	
5	Upper Lid Raiser	<i>Levator palpebrae superioris</i>	
6	Cheek Raiser	<i>Orbicularis oculi, pars orbitalis</i>	
7	Lid Tightener	<i>Orbicularis oculi, pars palpebralis</i>	
10	Upper Lip Raiser	<i>Levator labii superioris</i>	
12	Lip Corner Puller	<i>Zygomaticus major</i>	
14	Dimpler	<i>Buccinator</i>	
23	Lip Tightener	<i>Orbicularis oris</i>	
28	Lip Suck	<i>Orbicularis oris</i>	

4.2. táblázat: a hazugság esetén jellemző Action Unitok

5. Követelmény specifikáció

5.1. Webes felhasználói interfész

A következőkben bővebben meghatározásra kerül a webes felhasználói interfész nyújtotta funkcionalitás. Itt lesz lehetőség az elemzendő videó feltöltésére, az analízis szempontjából történő előbeállításra és az eredmény megtekintésére.

A felhasználó először egy videófeltöltő felületre érkezik. Az analízisnek vannak bizonyos minimumkövetelményei, melyeknek javasolt megfelelni az elemzés pontosságának elősegítése érdekében. Ezek részletesen megtalálhatók az előző fejezetben.

A videó kiválasztását követően a rendszer egy videószerkesztő felületen megjeleníti a beállítási lehetőségeket a felhasználó számára. Itt lehet megjelölni a videó vizsgálandó részeit, amik azok az intervallumok lesznek, melyekben a vizsgálandó személy reakciói találhatók.

Miután a videó megfelelő beállításai megtörténtek, megkezdődhet a feldolgozás. Ezt egy külön alkalmazás fogja elvégezni, amit a megfelelő API elérési pontok meghívásával lehet vezérelni. A feldolgozás kezdetekor megállapításra kerül a feldolgozás aktuális állapota, amit a webes felület megjelenít a felhasználónak, és bizonyos időközönként frissít.

A feldolgozást követően az eredmények egy külön megjelenítő felületen láthatók. Itt a felhasználó nagy részletességgel, grafikonok formájában, interaktív módon megtekintheti a részeredményeket, amelyek a *kontroll* és a *vizsgálandó* kérdésekre adott reakciók összehasonlításait, valamint a két kérdéscsoport reakcióinak részletes eredményeit tartalmazza. A végeredmény szintén itt látható.

Az eredmények exportálása JSON formátumba történik. Ezt a későbbi megtekintés céljából lehet majd importálni az eredménymegjelenítő felület segítségével.

5.2. Videófeldolgozó alkalmazás

A videófeldolgozó alkalmazás elérhetővé teszi a megfelelő API elérési pontokat, hogy azokon keresztül bárholnan könnyedén lehessen vizsgálatokat kezdeményezni, valamint eredményeket lekérdezni.

Az alkalmazás a megfelelő könyvtárak segítségével megvalósítja a videófeldolgozáshoz szükséges függőségeket, segédkomponenseket. A program feladata kizárólag egy adott videó feldolgozása a videó mellé megadott paraméterek alapján, a részenként történő elemzés és ezen részeredmények összegzése után pedig a megfelelő eredmények elérhetővé tétele egy széleskörben értelmezhető formában, JSON formátumban.

Ezek alapján tehát az alkalmazás bemenete egy elérési út a kívánt videóhoz, a kimenete pedig egy olyan adatszerkezet, ami egy előre meghatározott struktúra szerint tartalmazza a szükséges részeredményeket és az azokból következő végeredményt. Az eredmények úgy kerülnek összeállításra, hogy azok segítségével a végső következtetés alakulása könnyen lekövethető, valamint hiba, esetleg pontatlan eredmény esetén könnyen felismerhetők legyenek az ezt okozó tényezők.

6. Tervezés

6.1. Felhasználandó technológiák

6.1.1. Webes interfész, videókezelés

A webes interfészen keresztül van lehetőség az elemzendő videók feltöltésére, valamint a videók az elemzés szempontjából szükséges információk megadására.

Mivel itt komolyabb backend műveletek elvégzésére nincsen szükség, ezért a technológia megválasztásánál az elsődleges szempont a felhasználói élmény fokozása és az egyszerűség. Ennek a megvalósítására a Next.js az egyik legjobb lehetőség, ami egy viszonylag új, modern keretrendszer a React-es technológiához, ugyanis mindazok mellett, hogy rendelkezik annak előnyeivel, mint például – a teljesség igénye nélkül – az újrahasznosítható, dinamikus komponensek, a hatékonyan felépíthető, modern UI könyvtárak széles választéka, és a dinamikus állapotkezelés, lehetővé teszi a szerveroldali renderelést (SSR), melynek köszönhetően egy-egy oldal betöltésének a során a szükséges információk szervertől való lekérdezése sokkal egyszerűbb, a kliensek terhelése pedig jelentősen csökkenthető az jól optimalizált statikus oldalbetöltésekkel. A fejlesztés során nagy segítséget nyújt a Hot Module Replacement (HMR) funkció, amely még jobban elősegíti a fejlesztést azáltal, hogy minden módosítást követően az eredmény azonnal láthatóvá válik a rendszerben, ráadásul úgy, hogy csak a módosított modulok kerülnek automatikusan újratöltésre. TypeScript segítségével a kód tovább optimalizálható, valamint könnyedén megvalósítható az objektum orientált programozás, ami nagy mértékben hozzájárul egy hatékony, átlátható, könnyen karbantartható, bővíthető és újrafelhasználható kód létrejöttéhez.

Összességében a Next.js egy hatékony, stabil és gyors megoldást kínál a webes interfész által támasztott elvárásoknak való megfelelés szempontjából. A keretrendszer rendelkezik a megfelelő, modern technológiai megoldásokkal a kívánt felhasználói élmény eléréséhez.

6.1.2. Az arcfelismeréshez felhasználandó technológia

Az arcok felismeréséhez ma már létezik számos nagyon jó megoldás. Ezek alapvetően két csoportra bonthatók: felhő alapú és helyi keretrendszerek.

A felhő alapú szolgáltatások igénybevételének az előnye, hogy webes API kommunikáció segítségével egyszerűen és konkrétan el lehet érni a különböző funkciókat, maga a logika, és a működéshez szükséges adatbázis folyamatosan frissül, bővül. Így a fejlesztés során nem kell foglalkozni a feldolgozó szoftver részeinek, funkcióinak esetleges elavulásával, az új funkciók használatának alkalmazásával, mivel az API hívások adottak, a mögöttes logika időközbeni változása nem érinti annak felhasználást. Hátránya viszont, hogy csak bizonyos megadott funkciókat érhetünk el vele, a személyre szabhatóság alacsonyabb szinten nem feltétlen megoldható.

Ezzel szemben a helyi, keretrendszer alapú megoldások során amellet, hogy a keretrendszer frissítésével, új funkciók bejövételével és a régiek elavulásával nekünk kell foglalkoznunk, ezeket magunknak kell implementálnunk, a rendszerünk mégis jobban személyre szabható, az

egyes képfeldolgozási algoritmusok, működési elvek alacsonyabb szinten is hozzáférhetők és adott esetben módosíthatók.

Mivel a projekt esetében szükség van az egyes arcfelismeréssel kapcsolatos funkciók egyedi igények alapján történő módosítása, valamint a felhő alapú szolgáltatások árazása meglehetősen költséges, ezért a projekt során a helyi, keretrendszer alapú megoldások lesznek felhasználva.

6.1.3. Keretrendszer kiválasztása

Az OpenCV egy olyan könyvtár, amelynek segítségével valós idejű gépi látást igénylő alkalmazásokat fejleszthetünk. Főként a képfeldolgozásra, a videó rögzítésére és elemzésére összpontosít, beleértve az olyan funkciókat, mint az arcfelismerés és az objektumfelismerés. Az OpenCV manapság az egyik legnépszerűbb képelemző csomag, ami natív nVidia CUDA támogatással is rendelkezik. Meglehetősen széleskörű lehetőséget biztosít a képek alapos feldolgozásához, beleértve arcok felismerését, bizonyos arcpontok detektálását és számos hasonló művelet elvégzését, amikre a projekt során szükség van. Eredetileg C++ nyelven fejlesztették, de mára lehetőséget nyújt Python és Java alapokon történő fejlesztésre.

A választás okához nagy mértékben hozzájárult – amellet, hogy a könyvtár használatával kapcsolatosan széleskörű tapasztalatokra volt lehetőségem szert tenni a tanulmányaim során – az a tényező, hogy a projekt nyílt forráskódú, melynek köszönhetően a szoftver közösségi támogatottsága meglehetősen nagy, nyelvi elemei egyszerűek, valamint a dokumentációja is megfelelően érthető és jól kidolgozott. Mivel a könyvtárat olyan nagy cégek is használják a saját rendszereik fejlesztéséhez, mint az IBM és a Google, ezért a projekt folyamatos fejlődése túlmutat az egyszerű fejlesztői-felhasználói érdekeken.

6.1.4. A videófeldolgozáshoz felhasználandó technológia

A videó feldolgozásánál a programozási nyelv kiválasztásának a tekintetében az egyik legfontosabb szempont a sebesség. Minél hosszabb egy videó, annál több adatot gyűjthetünk ki belőle, ennél fogva végeredményként is egy pontosabb értékeket kaphatunk. Egy másodpercenként 30 képkockát tartalmazó, 5 perces videó esetében nagyjából 9000 képkockát kell feldolgozni. Egy-egy képkocka feldolgozása egy komplex, időigényesebb művelet, kiváltképp nagyobb felbontású videók esetében. A feldolgozást a webes interfészen keresztül lehet kezdeményezni, az érdemi munkát pedig – a lehető legnagyobb sebesség elérése érdekében – egy alacsony szintű programozási környezetben célszerű lebonyolítani, ahol maga a környezet a lehető legalacsonyabb overhaddel jár, az erőforrásokhoz való közvetlen hozzáférés biztosítható, és a memóriakezelés is a lehető leghatékonyabban eszközölhető.

A jól ismert C++ programozási nyelv megfelelő lehet a probléma megoldására, ugyanis a legjobb lehetőségek közé tartozik, ha gyorsaságról és hardverközeliségről van szó. A memóriakezelés is teljes mértékben támogatott, a fejlesztés során lehetőség van a legapróbb részletekbe menően meghatározni az erőforrások kezelésének a módját. A C++ támogatja az objektum orientált programozást, a jelenlegi, 20-as verziója pedig számos hatékony, jól optimalizált algoritmust és modern, beépített megoldást tartalmaz, amik a lehető leghatékonyabb lehetőségeket hivatottak nyújtani a fejlesztők számára. Application

Programming Interface (API) segítségével könnyű szerrel valósítható meg vele egy webes backend, ezáltal kifejezetten alkalmas a projekt során történő felhasználásra.

Nem elhanyagolható szempont továbbá, hogy a legtöbb képfeldolgozást segítő keretrendszer valamilyen módon mesterséges intelligenciát, gépi tanulást, gépi látási technikákat alkalmaz az analízis során, ezért a legtöbb ilyen technológia támogatja a Python nyelvet. A Python programozási nyelv számos mesterséges intelligenciával, gépi tanulással, gépi látással kapcsolatos könyvtárat tartalmaz, és használata ezen területeken meglehetősen közkedvelt. Ennek számos oka is van, ezek között a legjelentősebbek például a nem feltétlen szoftverfejlesztők számára is könnyen megérthető szintaktikai elemek és a komplex matematikai műveleteket végző funkciók egyszerű nyelvi elemekkel történő megvalósíthatósága.

Ahogy a fentiekben meghatározásra került, a projekt során az OpenCV könyvtár használata lesz segítségül a fejlesztésben. Mivel a könyvtár mind a C++, mind pedig a Python nyelvet támogatja, ezért a tényleges programozási nyelv, ami a projekt megvalósítására használva lesz, az a fejlesztés megkezdésének korai fázisában kerül megállapításra. A választás során egymással szembeállításra kerül a C++ által nyújtott teljesítmény, valamint a Python által a területen elérhető számos könyvtár és eszköz, valamint az egyszerűbb és kényelmesebben kezelhető szintaktikai megoldások. A döntés meghozatalában ezen két tényezőnek az osztályozása és összehasonlítása lesz a mérvadó (a végleges döntés a 7. fejezetben kerül tárgyalásra).

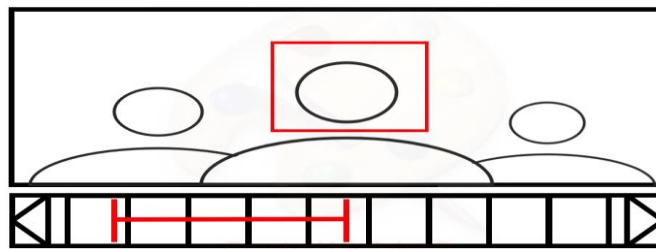
6.2. Rendszerterv

6.2.1. Webes felhasználói interfész

A webes interfész nem végez sem túl bonyolult, sem pedig túl sok műveletet. Gyakorlatilag csak előkészíti a videót az elemzéshez, kommunikál a feldolgozó alkalmazással, valamint megjeleníti az eredményeket. A főbb esetek, amiket kezelnie kell, azok az alábbi bekezdésekben kerülnek részletezésre. A weboldal Next.js technológia segítségével kerül felépítésre, és az egyes funkciók során az objektum orientált paradigma megvalósítását TypeScript segítségével oldja meg.

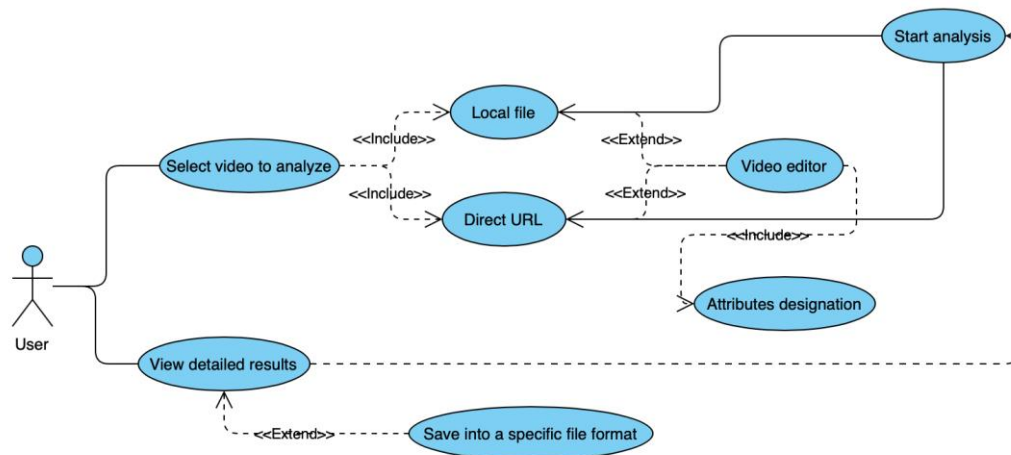
Első lépésben meg kell valósítani a videó webszerverre való feltöltésének a lehetőségét. Ehhez csak egy egyszerű HTML form és minimális szerver oldali validáció szükséges.

Ezt követően a feltöltött videó megjelenítésre kerül egy szerkesztői nézetben. A szerkesztői nézet lehetővé teszi a videó különböző intervallumainak megjelölését (6.1. ábra). Ez azért fontos, mert a feldolgozás során csak az ezekbe tartozó képkockák kerülnek elemzésre. Egy-egy intervallum tartalmazza a vizsgálandó személy egy adott kérdésre kapott reakcióját. Ezeket az intervallumokat két egységbe kell sorolni: az egyik a *kontroll* kérdésekre adott reakciókat, a másik pedig a *vizsgálandó* kérdésekre adott reakciókat tartalmazza.



6.1. ábra: a videó szerkesztésének vázlata. Látható a vizsgálni kívánt személy, valamint az időszávon történő szakaszmegjelölés.

Miután ezek a beállítások sikeresen megtörténtek, a videó továbbításra kerülhet a feldolgozó alkalmazás felé.



6.2. ábra: a videófeldolgozás folyamata nagy vonalakban a feldolgozás megkezdésétől az eredmények megtekintéséig.

A feldolgozó programmal való kommunikáció API hívásokon keresztül történik. A vizsgálat megkezdését követően az állapot és az eredmények lekérdezése a videót azonosító MD5 hash segítségével történik. Az oldal bizonyos időközönként lekérdezi a feldolgozás aktuális állapotát, és megjeleníti azt a felhasználó számára egy progress bar segítségével.

A feldolgozást követően megtörténik az eredmények lekérdezése, melyet a szerver JSON formátumban ad át, ami egy külön oldalon kerül megjelenítésre a felhasználó számára. Itt a felhasználó nagy részletességgel, grafikonok formájában, interaktív módon megtekintheti a részeredményeket, amelyek a *kontroll* és a *vizsgálandó* kérdésekre adott reakciók összehasonlításait, valamint a két kérdéscsoport reakcióinak részletes eredményeit tartalmazzák. A végső következtetés szintén itt látható.

Az eredmények exportálása JSON formátumba történik. Ezt a későbbi megtekintés céljából lehet majd importálni az eredménymegjelenítő felület segítségével.

A fentiekben megfogalmazott funkcionálisok természetesen egymástól függetlenül megírt, függetlenül működő komponensek, amik kizárólag a saját feladataik ellátásáért felelnek. A főbb komponensek az alábbiakban kerülnek bemutatásra:

- A **központi kezelő komponens** fogja meghívni az éppen szükséges komponensek funkcióit. Ez felel a felhasználó felé történő kommunikációért is, amennyiben arra szükség van.
- A **fájlkezelő komponens** felel a videófájl feltöltéséért, törléséért, valamint az elemezni kívánt videó kiválasztásáért.
- A **videószerkesztő komponens** egy megadott videó alapján összeállítja a szerkesztői felületet, amely lehetőséget biztosít az elemzendő intervallumok kiválasztásához.
- A **messenger komponens** feladata a kommunikáció végrehajtása a backenddel. Többek között ez felel az elemzendő videó továbbításáért, egy adott feldolgozás aktuális állapotának a lekérdezéséért, valamint a feldolgozást követően a feldolgozott videó és az eredmény JSON lekérdezéséért.
- Az **eredmény megjelenítő komponens** a kapott eredményeket átalakítja a felhasználó számára olvasható és értelmezhető formába, majd lehetővé teszi az eredmény mentését JSON formátumba. A részegységenként megkapott mikor-kifejezések gyűjteménye alapján megjeleníti, hogy a jelenlévő értékek milyen mértékben utalhatnak hazugság jeleire. A felhasználó a részletes eredményeket interaktív, grafikonos formában tekintheti meg.

6.2.2. Videófeldolgozó alkalmazás

A videófeldolgozó alkalmazás hivatott elvégezni a dolgok lényegi részét. Itt történik a videók elemzése és az eredmény előállítása. A program az objektum orientált paradigma alapelvei szerint kerül megvalósításra. A különböző funkciókat megvalósító komponensek az alábbi bekezdésekben kerülnek részletezésre.

Szükség lesz egy **figyelő egységre**, ami folyamatosan figyeli a beérkező API hívásokat.

- Amennyiben érkezik egy feldolgozási kérés, egyből indít neki egy saját szálát, majd visszaadja az elindított folyamat egyedi azonosítóját.
- Amennyiben nem feldolgozási, hanem állapotlekérési kérés érkezik, akkor az állapotlekérő egységet meghívva lekéri a kívánt folyamat aktuális állapotát, ami egy 0-tól 1-ig terjedő értéket vehet fel.

Az **központi egység** a különböző komponensek meghívásával levezényli a feldolgozást. Az analízis itt kezdődik, majd az eredmények kiértékelésével itt ér véget.

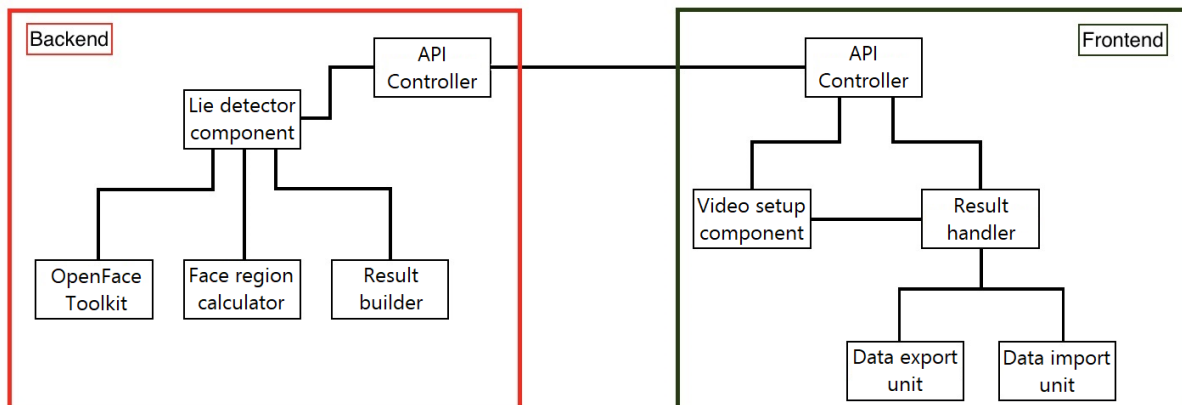
A **feldolgozó egységnek** két főbb feladata van:

- A videó egészére kiterjedően megkeresi és összesíti a videóban szereplő vizsgálandó személy arcizom mozgásaival kapcsolatos jellemző értékeket, melyeket egy-egy Action Unit formájában értelmez. Az eredményeket a videó MD5 azonosítójával címkézve eltárolja egy CSV fájlban, majd az értékeket egy mátrix formájában, képkockánként visszaadja a feldolgozó egységnek.
- A korábbiakban meghatározott módon, az OpenCV keretrendszer segítségével megkezdje a kapott képkockák feldolgozását. Ennek a feladata, hogy képkockánként megkeresse az elemzendő arcot, és meghatározza az arcjellemző pontokat, valamint azok segítségével a megfelelő arcrégiókat. Minden képkockán megjeleníti, hogy mely vizsgálandó csoportba tartozik, jelzi a jelen lévő Action Unit-ok aktuális intenzitás értékeit, valamint megjelöli az azokat tartalmazó arcrégiókat.

Az **eredmény kiértékelő egység** részegységeként eltárolja az értékeléseket. Miután az összes részegység eredménye beérkezett, kiszámításra kerül egy átfogó érték a videó egészére vetítve. Az összesített osztályozást követően megkezdődik az eredmény összeállítása egy olyan adatszerkezetbe, ami tartalmazza az összes információt a feldolgozással kapcsolatban, és ami felhasználásával részletes kimutatás készíthető és a feldolgozás modellezhető. Eltárolásra kerülnek például a képkocka-csoportonként, azon belül arcrégióként meghatározott mikro-kifejezések és azok külön értékelése, valamint az azok alapján kiszámított, összesített értékelés. Mindezek után az elkészült adatszerkezet bármikor lekérdezhető a későbbiekben, egy API híváson keresztül.

6.2.3. Komponensek

Az alábbi diagramon (6.3. ábra) látható nagy vonalakban a kliens és szerver oldali kommunikáció, valamint azokon belül az egyes főbb komponensek szerepe.



6.3. ábra: a kliens és szerver oldal, valamint azok főbb komponenseinek kapcsolata.

Backend

- **API Controller:** interfész az alkalmazás vezérléséhez.
- **Lie detector component:** a hazugságvizsgálatot levezénylő komponens.
- **OpenFace:** Action Unit-ok meghatározásáért felelős komponens.
- **Video marker component:** az arcjellemző pontok, az arcrégiók és a feldolgozás során kapott információk megjelölése a videó egyes képkockáin.
- **Result builder component:** az eredmények összeállításáért felelős komponens.

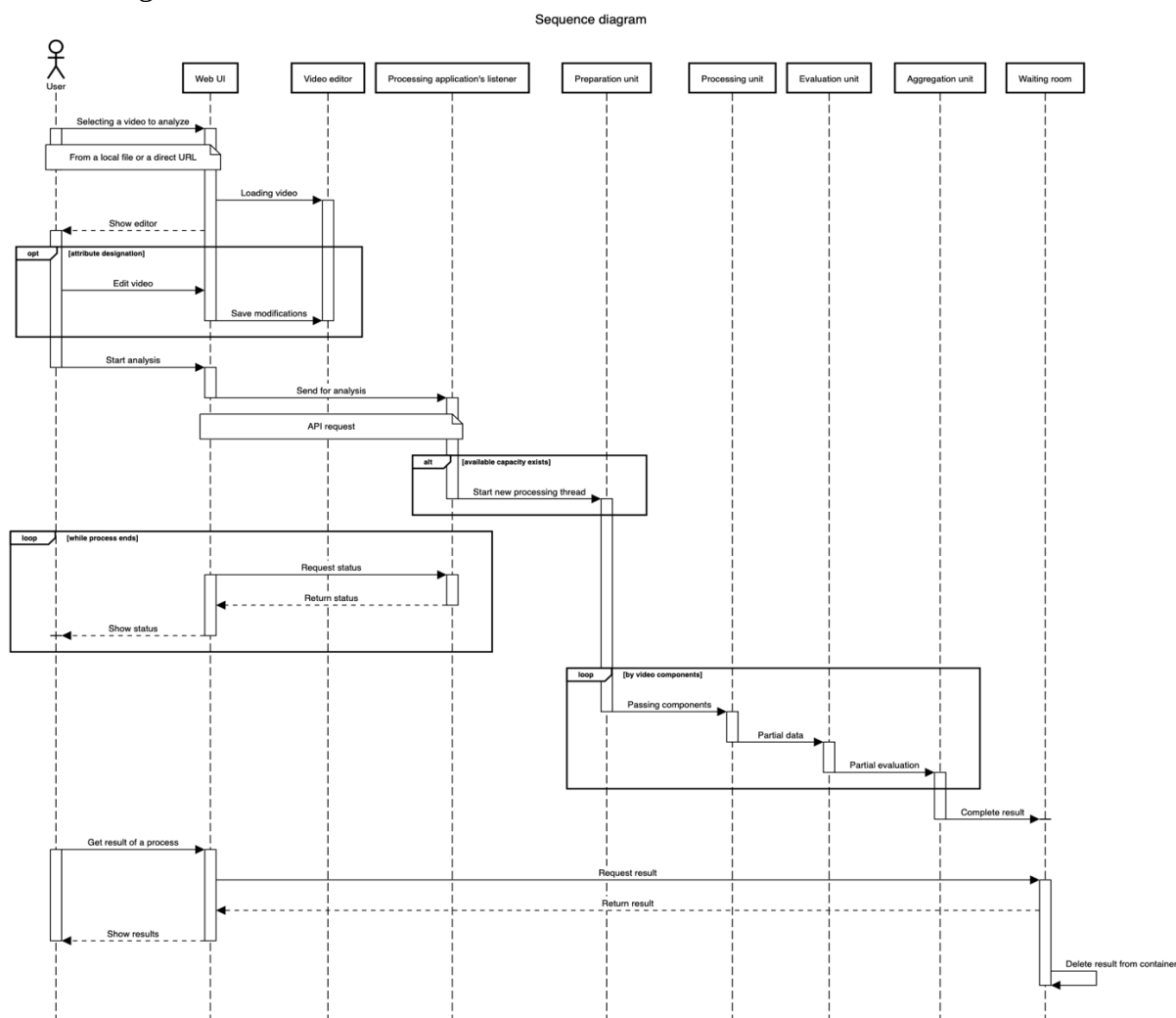
Frontend

- **API Controller:** kommunikáció a backenddel, kérések közvetítése.
- **Video Editor Component:** a videóval kapcsolatos szükséges információk meghatározása a felhasználó által.
- **Result processor:** a feldolgozás eredménye alapján a végkonklúzió meghatározása megjelenítése.
 - **Data export unit:** eredmények exportálása.
 - **Data import unit:** korábbi eredmények importálása azok újbóli megjelenítéséért.

6.3. Funkciólista

A rendszer az alábbiakban ismertetett funkciókkal rendelkezik:

- **Feltöltött videó szerkesztése:** az elemezni kívánt videó megadása, és a releváns szekciók megjelölésének lehetősége.
- **Eredmények megjelenítése:** a feldolgozás végeztével az eredmények automatikus megjelenítése a kiértékelő oldalon a felhasználó számára.
 - A kiértékelő oldalon grafikonos formában történik az adatok vizuális megjelenítése, a lehető legnagyobb részletességgel.
- **Eredmények exportálása:** a későbbi újrafelhasználás érdekében az eredmények mentése.
- **Eredmények importálása:** régebbi feldolgozás eredményének betöltése a kiértékelő oldalon, ahol az eredmények megtekintésén kívül lehetőség van a szakaszok újra hangolását követően ismételt elemzés futtatására.



6.4. ábra: szekvencia diagram a program egyes lépéseiről a felhasználó szempontjából.

6.4. Fejlesztés tervezett menete

A fejlesztés git alapú verziókezelő rendszer segítségével történik. A backend és a frontend rész is ugyanabba a repositoryba kerül.

A fejlesztés első fázisában a backend rész kerül kidolgozásra.

6.4.1. Backend

Az első lépés a kiinduló project létrehozása, a szükséges dependenciák importálásával, melyek közül a legfontosabbak a következők (a 6.1-es bekezdés alapján):

- **OpenCV:** valós idejű, optimalizált képfeldolgozáson és gépi látáson alapuló library.
- **OpenFace:** egy korszerű eszköz, mely többek között lehetőséget biztosít az Action Unit-ok meghatározására.

A függőséket inicializálását követően megtörténik a feldolgozásért felelős komponensek felépítése. A feldolgozás első lépése az Action Unit-ok meghatározása egy bemeneti videófájl esetében. Mivel a vizsgálat során csak azokra az Action Unit-okra lesz szükség, amelyek – az előzetes irodalomkutatások alapján – nagyobb valószínűséggel fordulnak elő hazugság esetén, így csak azok fogják az elemzés részét képezni.

A felhasználó felé történő vizuális visszajelzés érdekében a bemeneti videón képkockánkként feltüntetésre kerülnek az adott képkockára vonatkozó információk, amelyek a fellelhető releváns AU-k intenzitásának a mértékében, valamint az arc jellemző pontjainak meghatározásában merülnek ki. Kiemelésre kerül továbbá az az arcrégió, amelyen belül az AU megjelenik.

A rendszer tesztelése előre elkészített mintavideókkal fog történni, amelyek esetében előre ismert az elvárt kimenet. A tesztelés azután lesz esedékes, hogy a frontend részben megvalósult az eredményfeldolgozó komponens.

6.4.2. Frontend

Az első lépés itt is a kiinduló Next.js project létrehozása a TypeScript modullal.

Az első megvalósítandó komponens a videó feltöltését és vizsgálat előtti beállítását fogja biztosítani. Itt lesz lehetősége a felhasználónak megjelölni a videóban a releváns szakaszokat, melyek a következők:

- Azon szakaszok, melyekben a vizsgálandó alany a kontrollkérdésekre igazolhatóan őszintén válaszol.
- Azon szakaszok, melyekben a vizsgálandó alany azokra a kérdésekre válaszol, melyekre a hazugságvizsgálatot végre szeretnénk hajtani.

A megjelölt videószakaszok után az API Controller biztosítja a videó továbbítását a backend feldolgozó egysége felé. Ameddig a feldolgozás nem történik meg sikeresen, addig megadott időközönként lekérdezésre kerül annak állapota, ami megjelenik a felhasználó számára. Abban az esetben, ha a feldolgozás sikeres, az eredmény JSON formában lekérésre kerül.

Az eredmény feldolgozó komponens feladata, hogy a nyers eredményeket megjelenítse a felhasználó számára grafikus módon, átlátható formában, a szükséges magyarázatokkal ellátva. Ugyanitt lehetőséget van az eredmények exportálására a későbbi felhasználáshoz, valamint

korábbi eredmények importálására és azok alapján – az esetleges paraméterek módosítása után – egy megismételt feldolgozásra.

6.4.3. Utómunkálatok

Miután a szoftver működőképes és sikeresen megvalósítja az előírt funkciókat, megtörténik a bemeneti adatok validálása kliens és szerveroldalon egyaránt.

Szükséges továbbá a véglegesítés előtt egy kódellenőrzés, amely magába foglalja a szükséges refaktorálást, ami a kód tisztaságát és karbantarthatóságát biztosítja.

Utolsó lépésként pedig megtörténik a rendszer konténerizálása, amely lehetővé teszi a gyors és könnyű beüzemelést és használatot a továbbiakban.

6.4.4. A megvalósítás üteme

A projekt megvalósítása heti sprintekben történik, tehát egy részegység fejlesztésére hozzávetőlegesen 1 hét áll rendelkezésre. Így a projekt előreláthatólag kb. 10-11 hét alatt fog elkészülni, ami ideális, ugyanis az esetleges nem várt eseményekből fakadó csúszás kényelmes hibahatáron belül tud maradni, valamint így elegendő idő fog rendelkezésre állni a manuális tesztelésekre is.

A fentiek alapján a fejlesztés tervezett menete a következőképpen néz ki:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
GIT repository létrehozása	1													
Python projekt létrehozása	1													
AU detektálás megvalósítása	2		1											
C++ modul készítése a képkockák megjelöléséhez		2		1										
Kézi tesztelés mintavideókkal			2		1									
Next.js projekt létrehozása					1	1								
Videókezelő komponens létrehozása					2		1							
Eredmény-kiértékelő komponens létrehozása						2		1						
Eredmény import / export funkciók létrehozása							1	1						
Kliens / Szerver validáció								1						
Kódáttekintés, refaktor									1					
Konténerizáció										1	1			
Tesztelés önkéntesek segítségével												4		

Szorgalmi időszak adott hetén megvalósítandó

Lehetséges csúszás

6.5. ábra: a fejlesztés tervezett időbeosztása

7. Fejlesztés

7.1. A fejlesztés célja

A projekt célja a szándékos megtévesztés jeleinek felkutatása az emberi arcon megjelenő mikro-kifejezések alapján. Ebben a munkában a mikro-kifejezések alatt a Facial Action Coding System [FACS] által meghatározott Action Unit-ok [AU] értendők. A cél a vizsgálandó személy arcán megjelenő Action Unitok megtalálása, értékelése, és ezek alapján következtetések levonása a válaszadó szándékaira vonatkozóan.

7.2. A rendszer használatának előkövetelményei

A rendszer egy videót dolgozik fel, amelyben egy jól látható pozícióban szerepel a vizsgálandó személy arca. Fontos, hogy az elemzés során csak azok a képkockák a relevánsak, amelyek a vizsgálandó személy közvetlenül és kizárólag csak az adott kérdésre adott reakcióját tartalmazzák. Ezen intervallumok megjelöléséhez a felhasználói felületen lehetőséget kell biztosítani a felhasználóknak.

A videóban kérdések 2 csoportjának az intervallumait kell külön-külön megjelölni:

- Az egyik csoportba tartoznak a *kontroll* kérdések. Ezek azok a kérdések, amelyekre őszinte, ellenőrizhetően igaz válaszokat várunk.
- A másik csoport tartalmazza a *vizsgálandó* kérdéseket. Ezeknél a kérdéseknél nem tudjuk, hogy az adott válasznak mi az igazságtartalma, a választ az elemzéssel szeretnénk megmondani.

7.3. Üzleti logika

7.3.1. Action Unit detektálás

A hatékony AU detektáláshoz ezen a területen gyakoriak a gépi tanuláson alapuló megoldások, melyek – a matematikai megközelítések mellett – rendelkeznek egy előre betanított, robosztus modellkészlettel. Ennek a segítségével pedig még precízebb módon történhet az izommozgások meghatározása. Természetesen ehhez hatalmas adatbázisra és jelentősen sokféle mintavételre van szükség. Ebben a munkában az Action Unitok meghatározásához az *OpenFace* [13] nevű könyvtár kerül felhasználásra, amely biztosítja a megfelelő környezetet a hatékony észleléshez.

7.3.2. Kezdeti implementáció Python nyelven

A programozási nyelv kiválasztása

A képfeldolgozás és a gépi tanulás közkedvelt fejlesztési környezetét a Python nevű programozási nyelv nyújtja. Ennek számos oka is van, ezek között a legjelentősebbek például a nem feltétlen szoftverfejlesztők számára is könnyen megérthető szintaktikai elemek és a komplex matematikai műveleteket végző funkciók egyszerű nyelvi elemekkel történő megvalósíthatósága. Főként ezekből az okokból kifolyólag a fejlesztés a kezdeti fázisban ebben a környezetben történt.

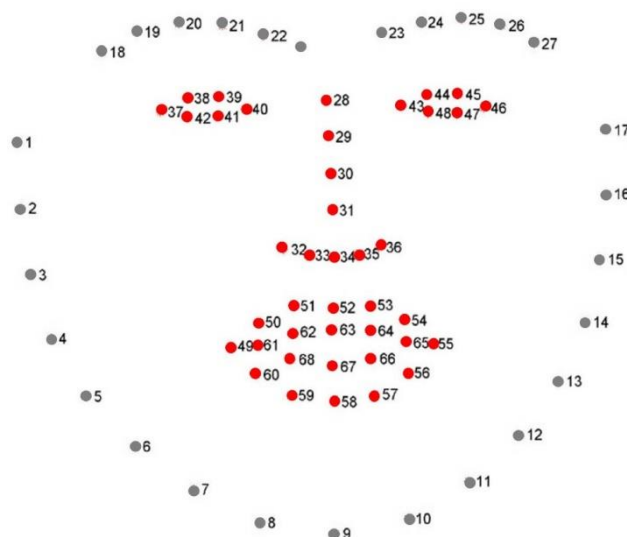
Alapvető funkcionalitás

A program egyik fontos feladata, hogy felismerje és elemezze a vizsgálandó személy arcán megjelenő izommozgásokat. Ezeket hívjuk mikro-kifejezéseknek, melyek egy-egy csoportját az úgynevezett Action Unit-ok reprezentálják. A későbbiekben ezek elemzésével történik majd a megtévesztésre utaló jelek megállapítása.

Fontos továbbá, hogy a vizsgálat részeredményeiről is átfogó képet adjunk a felhasználóknak, ezért a videókban lévő képkockákon megjelölésre kerülnek az adott képkockával kapcsolatos információk.

Ezek az információk a következők:

- az adott képkockán felismert arc jellemző pontjainak a megjelenítése 68 jellemző arcpont alapján
- azok az arcrégiók, amelyeken belül szignifikáns Action Unit megjelenhet. Az arcrégiókat határoló téglalapok határvonalai egy adott színnel vannak megjelölve. Azok a régiók, amelyeken belül az adott képkockán jelen van legalább egy szignifikáns Action Unit, a határoló téglalap határvonalai az alaptól eltérő színnel kerülnek megjelölésre
- az adott képkockán jelen lévő Action Unit-ok felsorolása, egyenként az intenzitási értékekkel együtt
- annak a megjelenítése, hogy az adott képkocka a vizsgálat szempontjából szignifikánsnak minősül-e, és amennyiben igen, melyik kérdéscsoporthoz tartozik
- az adott képkocka sorszáma



7.1. ábra: a 68 jellemző arcpont

Felmerülő problémák

Az alap funkcionalitás megvalósítását követően – Action Unit detektálás, feldolgozott képkockák megjelölése, arcrégiók meghatározása stb. –, ahogy a kezdetleges program egyre komplexebbé vált, a futásidőben egyre jelentősebb növekedés volt tapasztalható.

Ekkor kezdett el felmerülni az igény a különböző optimalizációs lehetőségek vizsgálatára. Ilyen volt például a képkockák megjelölésének a párhuzamos végrehajtása, amely a *Master-Worker* tervezési minta segítségével került megvalósításra. Ezt követték a kisebb refaktorálások, matematikai műveletek optimalizálásai, többször kiszámított eredmények eltárolása stb.

Az optimalizálási kísérletek eredménye

Az optimalizálási kísérletek annak ellenére, hogy javítottak a futásidőn, jelentős gyorsítást sajnos nem tettek lehetővé. Itt fontos továbbá azt is megjegyezni, hogy az Action Unit detektálás – mely a már korábbiakban tárgyalt, C++ környezetben implementált OpenFace nevű külső program futtatható állományának meghívásával történik – egy meglehetősen komplex, nagy számításigénnyel rendelkező műveletsorozat, ami ebből kifolyólag meglehetősen időigényes. Jelen esetben is sajnálatos módon a futásidő hossza nagy mértékben ebből következik.

További optimalizálási lehetőséget jelentene, ha a fejlesztés során az AU detektálás nem csak egy futtatható állomány meghívásával történne (mely sikeres futtatás után egy CSV dokumentumba menti az eredményeket képkockánként), hanem közvetlen függvényhívásokkal, esetleg a szükséges programkód kisebb, a jelen projekt szempontjából kedvezőbb módosításaival.

Lehetőségek a folytatásra

A fentiek alapján, tapasztalati úton bebizonyosodott, hogy az adott szempontok szerint megválasztott fejlesztési környezet mégsem bizonyult teljes mértékben megfelelőnek, így át kell gondolni, hogy a továbbiakban mi legyen az irány.

A legfontosabb szempontok az új fejlesztési környezet megválasztásához, hogy legyen egy alacsony szintű programnyelv, a gyorsaságot és a hatékony memóriakezelés lehetőségét elősegítendő, valamint biztosítson közvetlen hozzáférést az Action Unit detektáló külső komponens forráskódjához. A legkézenfekvőbb választás, ami erre a célra megfelel, és amelyben a továbbiakban az implementáció folytatódni fog, az a C++ programozási nyelv lesz.

7.3.3. Implementálás C++ nyelven

Előnyök a Python kóddal szemben

A fentiekben tárgyaltak alapján a fejlesztés C++ nyelven folytatódik. Szerencsére az eddig Python nyelven zajló fejlesztés során nem merült fel olyan mértékű függőség a programozási nyelvvel és annak környezetével kapcsolatban, ami jelentősen megnehezítené a váltást.

Az elsődleges cél az, hogy a már meglévő kód funkcionalitása megvalósításra kerüljön az új környezetben. Miután ez megtörtént, ellenőrzésre került a futásidő mérése. Ennek során

kiderült, hogy a váltás nem volt hiábavaló: hozzávetőlegesen 5-6-szoros gyorsulás tapasztalható az előző környezethez képest.

Ezt követően megtörténik a külső Action Unit detektáló komponens forráskód szinten történő felhasználása. Ezzel az eddigiektől eltérő módon már nem szükséges a futtatható állomány külön szálon történő meghívása, majd az exportált eredmények külön importálása, hanem közvetlenül a megfelelő függvények meghívásával, azok végrehajtását követően az eredmények folyamatosan eltárolhatóak a megfelelő adatszerkezetben.

Folyamatok gyorsítása

Egy videó tartalmazhat akár több, egymástól teljesen függetlenül vizsgálendő kérdéscsoportot, valamint olyan helyzet is előfordulhat, hogy ugyanazt a videót más-más szignifikáns időintervallumokkal szeretnénk vizsgálni, például amennyiben finom hangolni szeretnénk a kérdésekre érkezett reakciók intervallumait.

Fontos tulajdonsága a rendszernek, hogy minden feltöltött videónak meghatározza az MD5 hash értékét, és onnantól kezdve a videót kizárólag ez alapján az érték alapján azonosítja.

Ebből fakadóan az egyik fontos gyorsítási lehetőség, hogy a videó feltöltése előtt ellenőrzésre kerül, hogy létezik-e már korábban feltöltött videó az adott hash értékkel. Amennyiben igen, úgy a videó nem kerül ismételt feltöltésre, hanem a már meglévő változatot fogja használni a rendszer.

A másik fontos gyorsítási lehetőség, hogy minden elemzett videónak a hash értéke alapján egy külön fájlba exportálásra kerülnek az Action Unit értékei. Mivel az AU detektálás egy igencsak teljesítményigényes, hosszabb folyamat, ezért amennyiben egy már elemzett videót újbóli elemzésnek szeretnénk alávetni – csak például más paraméterekkel –, úgy megspórolhatjuk ezt a lépést egy egyszerű fájlolvasással.

API végpont

A webszerver a Crow nevű webes keretrendszer segítségével van létrehozva. A keretrendszer előnye, hogy gyors, egyszerű, és tartalmaz minden olyan funkcionalitást, amire a projekt szempontjából szükség van.

Az elkészült szolgáltatást 4 API végpont segítségével lehet használni. A 4 végpont a következő műveleteket teszi elérhetővé:

- *Feldolgozás megkezdése.*
 - Tartalmazza a feldolgozni kívánt videó elérhetőségét (a frontend szolgáltatás segítséget nyújt a kiválasztott videó feltöltéséhez), valamint a kérdésekre adott reakciók időintervallumait kérdéstípusonként külön-külön.
- *Feldolgozás állapotának lekérdezése.*
 - Visszaad egy 0 és 1 közötti értéket a feldolgozás állapotát illetően.
- *Eredmények lekérdezése JSON formátumban.*
- *Feldolgozott videó letöltése.*

Eredmények összeállítása

Az elemzés eredményei JSON formátumban kerülnek összeállításra, az előző pontban bemutatott Crow keretrendszer saját adatszerkezetének segítségével, amelynek az alapja a *std::unordered_map* típus, tehát a megfelelő JSON szerkezet felépítése egyszerűen megoldható.

A rendszer a *kontroll* és a *vizsgálándó* kérdésekre adott reakciókat külön-külön vizsgálja. Mindkét kérdéscsoport rendelkezik bizonyos szekciókkal. Minden szekció egy-egy kérdésre adott reakciót tartalmaz.

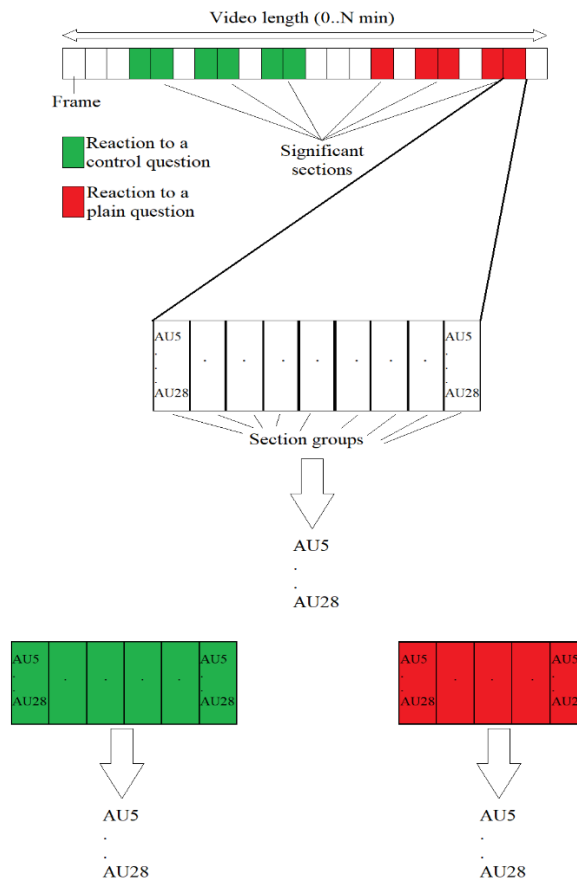
A szekciók további csoportokra vannak osztva, hogy az eredmények megtekintésekor részletekbe menően meg lehessen tekinteni az eredmények alakulását.

A szekciók csoportjaiban jelen lévő szignifikáns Action Unitok és azok intenzitás értékei összevetődnek, és azok az AU-k lesznek a relevánsak, amelyek intenzitása átlagosan eléri az előre megszabott küszöbértéket.

Ezt követően egy szinttel feljebb lépünk, és ugyanez megtörténik szekciónként is. Majd megint egy szinttel feljebb végül mindkét kérdéscsoportnál előáll egy lista, ami azokat az AU-kat tartalmazza, amelyek összesítve megfelelően relevánsak.

Ezek a hazugságra utaló Action Unitok a két kérdéscsoportban összevetésre kerülnek. Amennyiben egy vagy több AU értéke a *vizsgálándó* kérdések szakaszában jelentősen eltér a *kontroll* kérdéseknél kapott eredményektől, hazugságra utaló jelként lesznek kezelve.

A végső eredmény meghatározásánál két értéket kell figyelembe venni: a felfedezett – tehát a *kontroll* kérdéseknél jelentősen aktívabban szereplő – Action Unitok aránya az összes lehetséges Action Unithoz képest, valamint a felfedezett Action Unitok átlagos intenzitás értéke. A két érték adja a megfelelő súlyozásokkal a végeredményt.



7.2. ábra: az Action Unitok kiértékelésének a szintjei

Fejlesztési környezet értékelése

A C++ nyelvben történő implementáció összességében egy jó megoldásnak bizonyult, ugyanis az alacsony szintű megvalósításnak és az optimalizálási lehetőségeknek köszönhetően a futásidőben egy nagyon jelentős javulás érzékelhető (egy 80 másodperces, 30 FPS-el rendelkező videó esetében az arcpontok és arcrégiók megjelenítésénél ez hozzávetőlegesen 3-szoros gyorsulást eredményezett). Az erős típusosság és a programstruktúra hozzájárult ahhoz, hogy egy robusztus, átlátható módon felépített program jöjjön létre, aminek köszönhetően a hibalehetőségek is jelentősen visszaszorultak.

7.4. Webes felhasználói interfész

7.4.1. Fejlesztési környezet megválasztása

A webes felület a NextJS nevű keretrendszer segítségével kerül megvalósításra. Ez rendelkezik a React technológia előnyeivel, mint például a dinamikus állapot- és komponenskezelés, a TypeScript támogatás stb. Ezen kívül rendelkezik még olyan egyéb funkciókkal, melyeknek köszönhetően egy-egy oldal megjelenítése sokkal optimálisabban történik, továbbá lehetőséget nyújt statikus és szerver oldali oldalgenerálásra.

Az átlátható, modern és felhasználóbarát megjelenés érdekében a PrimeReact nevű UI könyvtár és azok komponensei kerülnek felhasználásra.

7.4.2. Megvalósított funkciók

Videók feltöltése

Első körben a videókezelés került megvalósításra. Ennek a célja, hogy lehetőséget nyújtson a felhasználóknak az elemzendő videók feltöltésére, listázására és törlésére. Minden feltöltött videót lehetőség van külön-külön előkészíteni a feldolgozáshoz.

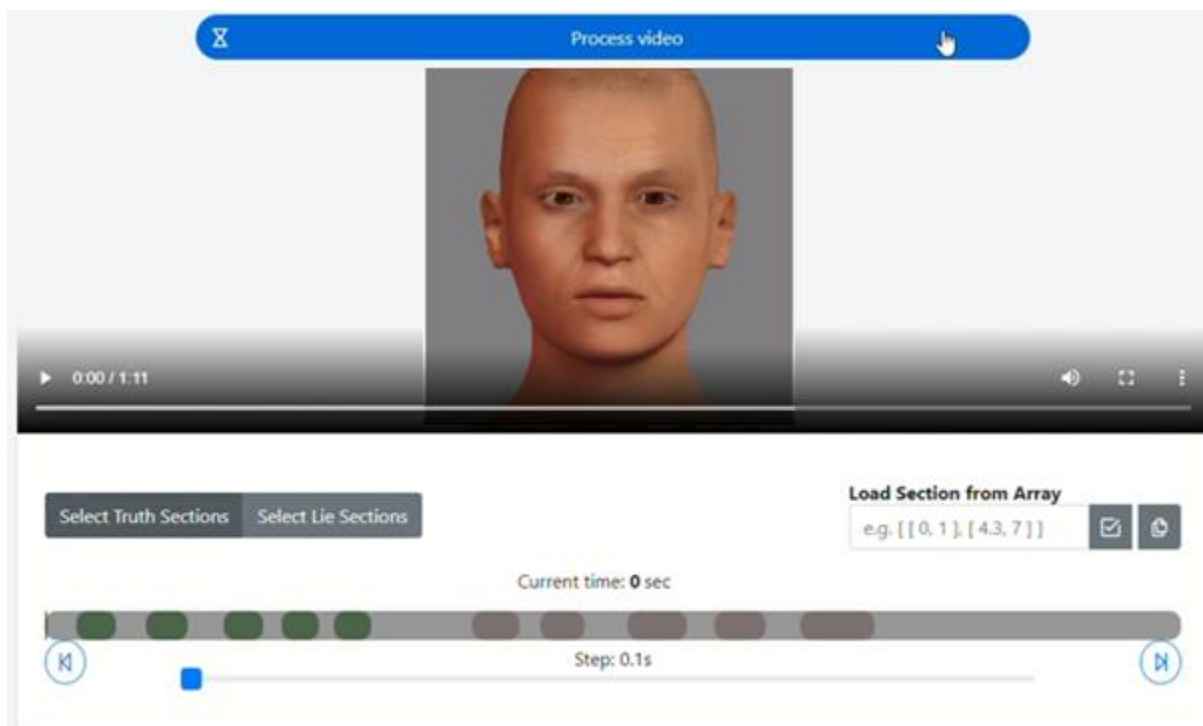
Videó előfeldolgozása

A videó előfeldolgozásához egy külön komponens készült, amely segítségével lehetőség van a videó megtekintésére, valamint a *kontroll* és a *vizsgálható* kérdésekre adott reakciókat tartalmazó szakaszok kijelölésére. Erre a célra létrehozásra került egy olyan alkomponens, amely kivetíti a videó lejátszási sávját és interaktív módon segíti a szakaszok meghatározását.

A kurzort a sáv adott pontjára mozgatva megjelenik a videó adott ponton éppen aktuális időtartama, és abban az esetben, amennyiben az adott ponton egy már kijelölt szakasz is található, megjelenik az intervallumtól-ig értéke.

A szakaszok megjelöléséhez lehetőség van az intervallum értékeinek a megadására JSON tömb formátumban. Ugyanebbe a formátumba ki lehet másolni a már megjelölt területek intervallumait.

Mivel a vizsgálat során kifejezetten fontos, hogy a megjelölt reakciók kezdete és vége a lehető legpontosabban legyen kijelölve, ezért egy további alkomponens lehetőséget nyújt a képkockák közötti aprólékos előre és hátralépésre. A lépések nagysága tetszőlegesen beállítható a megadott határértékek között.



7.3. ábra: a videó előkészítése a feldolgozáshoz

Videó feldolgozása

A feldolgozási paraméterek megadását követően a videó feldolgozásra kerülhet. A felhasználói kezdeményezést követően a frontend API hívások segítségével elküldi a feldolgozandó adatokat a backend részére, majd amennyiben válaszként megkapja, hogy a feldolgozás sikeresen megkezdődött, megjelenít egy állapotjelző elemet a videó fölött, amin nyomon követhető a feldolgozás aktuális állapota, százalékos formában. Az állapot lekérdezése a kliens által szintén API hívásokkal történik, 500 milliszekundumonként.

Amennyiben az állapot lekérdezések során az aktuális állapot eléri az „1.0” értéket, a feldolgozás a végére ért, a frontend lekérdezi az eredményeket JSON formátumban, valamint a feldolgozott videót, amin megjelölésre kerültek a megfelelő vizuális elemek. Ezek után megjelenik egy gomb, amin a felhasználó tovább navigálhat az eredmény kiértékelő oldalra.

A videófeldolgozás és az eredmények állapotmegőrzőek, tehát egy elindított feldolgozás minden esetben megjelenítésre kerül, ha például a felhasználó a feldolgozás ideje alatt egy másik oldalra navigál. Hasonló módon működik a feldolgozott videó megjelenítése: amennyiben egy videó feldolgozásra került, a videó megjelenítése automatikusan a feldolgozott videó formájában történik, külön navigálási lehetőséggel az eredmény megjelenítő oldalra.

Ha a felhasználó ismételten fel szeretné dolgozni ugyanazt a videót, lehetőség van a számára a videót visszaállítani, és a feldolgozást előlről kezdeni.

Eredmények megjelenítése

Az eredmények megjelenítésére készült komponens a szerver által visszaadott eredményeket tartalmazó JSON fájl tartalmát jeleníti meg a felhasználó számára vizuális módon, grafikonok segítségével.

A feldolgozott videó megjelenése alatt található egy opcionálisan kinyitható panel, ami tartalmazza az eredmények megjelenítésével és bizonyos részek értelmezésével kapcsolatos leírásokat.

A grafikus ábrázolás 4 külön komponens segítségével történik. Minden komponensnél, ahol szükség van rá, található egy leírást tartalmazó panel, ami opcionálisan megjeleníthető.

Az első komponens összehasonlítja a *kontroll* és a *vizsgálható* kérdésekre adott válaszok adatait: azokat az Action Unitokat – átlagos intenzitás értékekkel együtt –, amelyek az adott szakaszban szignifikánsan jelen vannak. Egy radar diagramon megjelenítésre kerül mindkét kérdéstípus összesített eredménye. Innen már könnyen kiolvashatóak az esetleges eltérések. Ez alatt ugyanezek az értékek oszlopdiagram formájában is megtekinthetők.

A második és harmadik komponens megjeleníti külön-külön oszlopdiagrammokon a két kérdéstípus esetén észlelt Action Unitokat és azok átlagos intenzitásait. Itt a konkrét kérdésekre adott reakciónként is részletesen megtekinthetők a kapott értékek.

A negyedik komponens mutatja meg az összesítést. Megjeleníti, hogy melyek azok az Action Unitok, amik szignifikáns eltéréssel vannak jelen a *vizsgálható* szakaszok esetén a *kontroll* szakaszokhoz képest. Ezek alapján pedig látható a végső következtetés azzal kapcsolatban, hogy vannak-e a vizsgált szakaszokban hazugságra utaló jelek, és ha igen, ezek milyen mértékben tapasztalhatók.

Eredmények mentése

Az eredményeket lehetőség van letölteni JSON formájában is a későbbi megtekintéshez és egy esetleges újrafeldolgozáshoz.

Amennyiben a felhasználó egy korábban lementett eredményfájlt szeretne betölteni, lehetősége van azt megtenni az eredmény megjelenítő panel külön erre a célra szolgáló felületén. Abban az esetben, ha az eredményfájlban szereplő MD5 hashnek megfelelő videó létezik a feltöltött videók között, úgy az megjelenítésre kerül, ellenkező esetben csak a grafikonos kimutatás elérhető.

A keretrendszer értékelése

A program megvalósítását követően kijelenthető, hogy a megválasztott keretrendszer képes egy modern, a felhasználó számára interaktívan kezelhető és a beépített funkcióknak köszönhetően optimálisan működő, dinamikus weboldalt megjeleníteni és kezelni.

A felhasznált UI könyvtár segítségével az alapvető webes komponensek letisztult és jól alkalmazható struktúrát kapnak.

Ennek a segítségével a nyers adatok megjelenítése grafikus formában nemcsak egyszerű és letisztult, de olyan felhasználói élményt fokozó előnyökkel is jár az interaktív kezelhetőség szempontjából, amelyek elengedhetetlenek ahhoz, hogy az interfész megfeleljen mind a felhasználói, mind a fejlesztői elvárásoknak egy modern weboldal esetében.

7.5. Konténerizáció

Annak érdekében, hogy ez elkészült programot könnyedén lehessen futtatni a Docker konténerek futtatására alkalmas operációs rendszerrel rendelkező eszközök széles spektrumán, mind a backend, mind pedig a frontend alkalmazáshoz külön készül egy-egy Docker image.

Azon kívül, hogy a konténerek tartalmazzák az adott rendszerhez szükséges függőségeket, folyamatosan naprakészen tartják a programot is.

A konténerek minden elindítással letöltik az adott program módosításait a megfelelő GitHub repository-ból, majd a buildelést követően automatikusan elindítják a webszervert.

8. Tesztelés

A program tesztelése alapvetően kétféle módon történik.

Egy automatikus Unit Test segítségével történik gyakorlatilag az egész videóelemzési folyamat tesztelése. Ennek során egyúttal a részfolyamatok is tesztelésre kerülnek, hiszen a végső eredmények előállításához több részkomponens együttes munkája szükséges. A tesztelés fő kérdése, amire a választ keresi, az nem más, minthogy az Action Unitok a megfelelő módon kerültek-e kiértékelésre és összegzésre a két kérdéstípus esetén, valamint a végső következtetés megfelel-e az előre megadott kimenetelnek. Ezen felül a feldolgozott videón megjelenő vizuális elemek és visszajelzések megfelelő jelenléte manuális ellenőrzést igényel.

A rendszer tényleges, éles helyzetben történő megfelelő működésének az ellenőrzése manuális tesztekkel lesz elvégezve. Ezeknek a teszteknek a lényege, hogy önkéntesek segítségével, optimális környezetben, a megfelelő körülmények között tesztvideók kerülnek rögzítésre, amelyek vizsgálata során ellenőrzésre kerül, hogy a rendszer a megadott videók esetén az elvárt eredményeket produkálja-e.

8.1. Unit Testing

8.1.1. Tesztvideó előállítása

Az automatikusan futtatható tesztekhez szükség van egy rövid tesztvideóra, amelyen előre megadott Action Unitok vannak megjelenítve, hogy aztán azokat a teszt során észlelhessük és jóváhagyhassuk.

Annak az érdekében, hogy ez a tesztvideó könnyen és gyorsan feldolgozható legyen, ami a Unit Test szempontjából is fontos, egy olyan videóra van szükség, ami csak és kizárólag a teszteléshez szükséges tartalmat foglalja magában, a lehető legpontosabb megjelenítéssel, a lehető legrövidebb időtartamban.

Ehhez a legideálisabb megoldás, ha egy mesterségesen előállított emberi arcot használunk a teszteléshez, ami – viszonylagos anatómiai pontosságot nyújtva – a megfelelő animációk segítségével megjeleníti az előre beállított arckifejezéseket.

A tesztvideó előállítása egy 3 dimenziós, számítógépes modellezés segítségével történik. A modellezés a „*MakeHuman Community*” nevű programmal történik, ami egy nyílt forráskódú eszköz 3D emberi karakterek létrehozására. Ennek a programnak létezik egy „*FACSHuman*” nevű kiegészítője, amely lehetőséget biztosít az adott modell arcán történő érzelmi reakciók megjelenítéséhez, méghozzá Action Unitok segítségével. Mivel a szoftver a Facial Action Coding System által definiált összes létező Action Unit beállítását lehetővé teszi meglehetősen nagy részletességgel, ezért teljesen ideális választás egy adott teszteset összeállításához.

A megfelelő arcszerkezet beállítását követően a szoftver lehetőséget biztosít a megfelelő animáció létrehozásához. Ehhez meg kell adni a generálni kívánt képkockák számát és méretét. A jelen esetben a tesztvideó létrehozásánál egy érzelmi reakció-együttes megjelenítéséhez 140 darab, 500x500-as képkocka került generálásra.



8.1. ábra: képkockák a teszteléshez modellezett arc animációkból

Mivel a vizsgálat elvárása, hogy a vizsgált videónak tartalmaznia kell legalább 1-1 reakciót mind a *kontroll*, mind pedig a *vizsgálható* szakaszban, ezért a tesztvideó a következő módon lett összeállítva:

- Tartalmaz egy olyan animációt, amelyen előre megadott Action Unitok szerepelnek. Ez lesz a *kontroll* szakasz.
- Tartalmaz egy olyan animációt, amelyen a *kontroll* szakaszban megadott Action Unitokhoz képest szignifikáns eltéréssel szerepelnek Action Unitok. Ez lesz a *vizsgálható* szakasz első esete.
- Tartalmaz egy olyan animációt, amelyen a *kontroll* szakaszban megadott Action Unitokhoz képest hasonló Action Unitok, hasonló intenzitással szerepelnek. Ez lesz a *vizsgálható* szakasz második esete.

8.1.2. A teszteléssel szembeni elvárások

A fentiek alapján a teszt kimenetelével szembeni az elvárások, hogy a tesztvideó megfelelő szakaszaiban hiánytalanul legyenek felismerve az animációban előre beállított Action Unitok, és a végeredmény is ezek alapján legyen meghatározva.

A tesztvideóban két külön *vizsgálható* szakasz is jelen van, ezért két külön tesztet vizsgálható. Az első esetben a *kontroll* szakasz és az első *vizsgálható* szakasz kerül összevetésre, melynek során jelentős eltérést kell tapasztalnunk a két szakasz között, ami a végeredményben is megjelenik. A *kontroll* szakasz és a második *vizsgálható* szakasz között az eltérés azonban nem lehet a vizsgálat szempontjából releváns, tehát a végeredményben nem jelenhetnek meg hazugságra utaló jelek. Az automatizált Unit Tesztek tehát nem a hazugságvizsgálat eredményességét, hanem a rendszer funkcionalitását vizsgálják.

8.1.3. A tesztelés technikai megvalósítása

A projekt során számos funkció megvalósításához szükség van egy Boost nevű könyvtárra. Ez egy olyan könyvtárkészlet a C++ programozási nyelvhez, amely számos hasznos megoldást és segédeszközt biztosít a fejlesztés során a matematikai eszközöktől kezdve, a többszálú feldolgozáson át a képfeldolgozáson keresztül egészen a Unit Tesztelésig.

Mivel a Boost megfelelő környezetet termet Unit Testek megvalósítására, ezért a projektben a könyvtár által szolgáltatott tesztelési lehetőségek lesznek felhasználva.

Egy tesztet modellezéséhez szükség van egy olyan osztályra, amely tartalmazza az összes olyan adatot, amelyre a tesztek során szükség lehet. Ennek a felhasználásával könnyedén előállíthatók különböző tesztesetek a megfelelő videók hozzáadásával és konfigurálásával.

```

18 struct TestClass {
19     string path;
20     string md5;
21     vector<vector<double>> truth_sections;
22     vector<vector<double>> lie_sections;
23     vector<int> expected_truth_AUs;
24     vector<int> expected_lie_AUs;
25     vector<int> expected_overall_AUs;
26
27     TestClass() {}
28
29     TestClass(
30         const string& path,
31         const vector<vector<double>>& truth_sections,
32         const vector<vector<double>>& lie_sections,
33         const vector<int>& expected_truth_AUs,
34         const vector<int>& expected_lie_AUs,
35         const vector<int>& expected_overall_AUs
36     ) {
37         this->path = path;
38         this->md5 = file_hasher::getFileHash(path);
39         this->truth_sections = truth_sections;
40         this->lie_sections = lie_sections;
41         this->expected_truth_AUs = expected_truth_AUs;
42         this->expected_lie_AUs = expected_lie_AUs;
43         this->expected_overall_AUs = expected_overall_AUs;
44     }
45 };

53 static const boost::array<TestClass, 2> TEST_FILES{
54     TestClass("files/AU_test.mp4",
55         {{0, 5.6}},
56         {{5.7, 11.2}},
57         {6, 7, 10, 12, 14},
58         {6, 10, 12, 14},
59         {12}},
60     TestClass("files/AU_test.mp4",
61         {{0, 5.6}},
62         {{11.3, 16.9}},
63         {6, 7, 10, 12, 14},
64         {6, 10, 12, 14},
65         {}))
66 };

```

8.2. ábra: a teszt osztály és a tesztesetek definiálása

8.1.4. A Unit Tesztek eredményei

Eredmények a *kontroll* szakasz és a releváns eltéréseket tartalmazó első *vizsgálendő* szakasz összehasonlítását követően:

Vizsgálat helye	Leírás	Elvárt AU-k	Eredmény
Kontroll szakasz	Az itt jelen lévő Action Unitok képzik a vizsgálat alapját, az eltérések ezekről lesznek számítva.	6, 7, 10, 12, 14	✓
Vizsgálendő szakasz	Az itt jelen lévő Action Unitok kontroll szakaszhoz viszonyított eltéréseiből áll elő a végeredmény.	6, 10, 12, 14	✓
Végeredmény	A kontroll és a vizsgálendő szakaszok közötti jelentős eltéréssel jelen lévő Action Unitok.	12	✓

8.1. táblázat: a teszt eredménye a kontroll szakasz és az első vizsgálendő szakasz esetén

A tesztet végeredménye során megállapítható, hogy a *kontroll* és a *vizsgálendő* szakaszok között jelentős eltérés tapasztalható a 12-es Action Unit formájában.

Eredmények a *kontroll* szakasz és a releváns eltéréseket nem tartalmazó második *vizsgálándó* szakasz összehasonlítását követően:

Vizsgálat helye	Leírás	Elvárt AU-k	Eredmény
Kontroll szakasz	Az itt jelen lévő Action Unitok képzik a vizsgálat alapját, az eltérések ezekről lesznek számítva.	6, 7, 10, 12, 14	✓
Vizsgálándó szakasz	Az itt jelen lévő Action Unitok kontroll szakaszhoz viszonyított eltéréseiből áll elő a végeredmény.	6, 10, 12, 14	✓
Végeredmény	A kontroll és a vizsgálándó szakaszok közötti jelentős eltéréssel jelen lévő Action Unitok.	-	✓

8.2. táblázat: a teszt eredménye a kontroll szakasz és a második vizsgálándó szakasz esetén

A tesztelés végeredménye során megállapítható, hogy a *kontroll* és a *vizsgálándó* szakaszok között jelentős eltérés nem tapasztalható semelyik Action Unit formájában.

A tesztelés során ellenőrizni kell, hogy a tesztvideó feldolgozása után az eredmények megfelelő módon előálltak-e:

Vizsgált elem	Leírás	Elvárt állapot	Eredmény
Állapotváltozó	A vizsgálat végeztével az aktuális állapotváltozó értékének kész állapotot kell visszajeleznie.	1.0	✓
Action Unitok exportálása	A vizsgálat során a videóban felfedezett Action Unitok egy CSV fájlban exportálásra kell, hogy kerüljenek.	A fájl létezik: Output/ Md5_hash/ action_units.csv	✓
Feldolgozott videó létrehozása	A vizsgálat során a megfelelő vizuális jelölésekkel ellátott videó jön létre.	A fájl létezik: Output/ Md5_hash/ Result.mp4	✓

8.3. táblázat: a teszt eredménye a feldolgozás ellenőrzésére

8.2. Manuális tesztelés

8.2.1. Tesztelési módszertan

Az automatikus tesztek mellett a program tényleges, éles környezetben történő megfelelő működésének ellenőrzése manuális teszteléssel történik. Ezekben a tesztekben olyan önkéntesek vesznek részt, akik vállalták, hogy anonim módon hozzájárulnak a róluk történő videófelvétel készítéséhez és a felvételek elemzéséhez.

A tesztek elvégzése a következő módon történik:

- 5 db *kontroll* kérdés, melyre őszinte, igaz válaszokat várunk. Az ezekre adott reakciók értékei lesznek a vizsgált alany alapértékei.
- 5 db *vizsgálandó* kérdés, melyekre válaszként hazugságot várunk.
- 5 db *vizsgálandó* kérdés, melyekre őszinte, igaz válaszokat várunk.

A tesztelés célja, hogy a *kontroll* kérdések és a *vizsgálandó* kérdések első csoportjának az együttes vizsgálatát követően a rendszer hazugságra utaló jeleket állapítson meg, míg a *kontroll* és a *vizsgálandó* kérdések második csoportja között ne találjon hazugságra utaló jeleket.

Az alábbi táblázatokban az látható, hogy az adott tesztalany esetében milyen mértékű arcizom mozgások tapasztalhatók a különböző kérdésekre adott reakciók esetén.

A táblázatok utolsó sorában lévő „Eltérés” alatt a *vizsgálandó* kérdésekre adott reakciók pozitív mértékű eltérését kell érteni a *kontroll* kérdésekre adott reakciók értékeitől. Adott pozitív irányú eltérés akkor számít szignifikáns mértékűnek, ha az értéke meghaladja az 5%-ot. Hasonló módon, egy Action Unit is csak akkor kerül beszámításra, ha az intenzitása eléri a 25%-ot.

8.2.2. Tesztek eredményei

1. Tesztalany

a. A vizsgálandó kérdésekre hazugságtartalmú válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	31.82	-	30.47	30.99	35.02	28.08	25.25	-	-
Vizsgálandó	-	-	31.36	-	25.82	28.44	31.36	29.30	27.78	28.60	-
Eltérés	-	-	-	-	-	-	-	1.21	2.53	28.60	-

8.4. táblázat: az 1-es számú tesztalany eredményei a hazugságtartalmú válaszok esetén

A táblázat alapján a vizsgálat során jelentős eltérés tapasztalható a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

A hazugságra utaló jelek sikeresen észlelve lettek.

b. A vizsgálandó kérdésekre őszinte válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	31.82	-	30.47	30.99	35.02	28.08	25.25	-	-
Vizsgálandó	-	-	26.92	-	29.14	31.07	39.46	29.09	-	-	-
Eltérés	-	-	-	-	-	0.09	4.44	1.01	-	-	-

8.5. táblázat: az 1-es számú tesztalany eredményei az őszinte válaszok esetén

A táblázat alapján a vizsgálat során nem tapasztalható jelentős eltérés a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

Hazugságra utaló jelek nem voltak tapasztalhatók jelentős mértékben.

2. Tesztalany

a. A vizsgálandó kérdésekre hazugságtartalmú válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	-	-	-	26.99	-	-	-	-	-
Vizsgálandó	-	-	-	-	30.73	32.93	-	32.32	-	-	-
Eltérés	-	-	-	-	30.73	5.931	-	32.32	-	-	-

8.6. táblázat: a 2-es számú tesztalany eredményei a hazugságtartalmú válaszok esetén

A táblázat alapján a vizsgálat során jelentős eltérés tapasztalható a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

A hazugságra utaló jelek sikeresen észlelve lettek.

b. A vizsgálandó kérdésekre őszinte válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	-	-	-	27.03	-	-	-	-	-
Vizsgálandó	-	-	-	-	-	29.91	-	-	-	-	-
Eltérés	-	-	-	-	-	2.88	-	-	-	-	-

8.7. táblázat: a 2-es számú tesztalany eredményei az őszinte válaszok esetén

A táblázat alapján a vizsgálat során nem tapasztalható jelentős eltérés a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

Hazugságra utaló jelek nem voltak tapasztalhatók jelentős mértékben.

3. Tesztalany

a. A vizsgálandó kérdésekre hazugságtartalmú válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	30.42	-	-	33.39	-	30.17	27.81	-	-
Vizsgálandó	30.88	30.46	33.01	-	-	31.03	-	29.16	27.31	-	-
Eltérés	30.88	30.46	2.59	-	-	-	-	-	-	-	-

8.8 táblázat: a 3-mas számú tesztalany eredményei a hazugságtartalmú válaszok esetén

A táblázat alapján a vizsgálat során jelentős eltérés tapasztalható a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

A hazugságra utaló jelek sikeresen észlelve lettek.

b. A vizsgálandó kérdésekre őszinte válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	30.42	-	-	33.39	-	30.17	27.81	-	-
Vizsgálandó	-	-	31.19	-	-	34.92	-	29.04	27.58	-	-
Eltérés	-	-	0.77	-	-	1.53	-	-	-	-	-

8.9. táblázat: a 3-mas számú tesztalany eredményei az őszinte válaszok esetén

A táblázat alapján a vizsgálat során nem tapasztalható jelentős eltérés a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

Hazugságra utaló jelek nem voltak tapasztalhatók jelentős mértékben.

4. Tesztalany

a. A vizsgálandó kérdésekre hazugságtartalmú válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	26.20	-	-	-	25.71	-	-	-	-
Vizsgálandó	-	-	25.98	-	-	-	32.30	34.77	-	-	-
Eltérés	-	-	-	-	-	-	6.59	34.77	-	-	-

8.10. táblázat: a 4-es számú tesztalany eredményei a hazugságtartalmú válaszok esetén

A táblázat alapján a vizsgálat során jelentős eltérés tapasztalható a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

A hazugságra utaló jelek sikeresen észlelve lettek.

b. A vizsgálandó kérdésekre őszinte válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	26.20	-	-	-	25.71	-	-	-	-
Vizsgálandó	-	-	26.51	-	-	-	-	-	-	-	-
Eltérés	-	-	0.31	-	-	-	-	-	-	-	-

8.11. táblázat: a 4-es számú tesztalany eredményei az őszinte válaszok esetén

A táblázat alapján a vizsgálat során nem tapasztalható jelentős eltérés a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

Hazugságra utaló jelek nem voltak tapasztalhatók jelentős mértékben.

5. Tesztalany

a. A vizsgálandó kérdésekre hazugságtartalmú válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	25.99	-	-	25.00	-	-	-	-	-
Vizsgálandó	-	-	26.97	-	-	28.07	35.12	-	-	-	-
Eltérés	-	-	0.98	-	-	3.07	35.12	-	-	-	-

8.12. táblázat: az 5-ös számú tesztalany eredményei a hazugságtartalmú válaszok esetén

A táblázat alapján a vizsgálat során jelentős eltérés tapasztalható a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

A hazugságra utaló jelek sikeresen észlelve lettek.

b. A vizsgálandó kérdésekre őszinte válaszok adása esetén

Csoport/AU	#1	#2	#4	#5	#6	#7	#10	#12	#14	#23	#28
Kontroll	-	-	25.99	-	-	25.00	-	-	-	-	-
Vizsgálandó	-	-	26.62	-	-	-	-	-	-	-	-
Eltérés	-	-	0.63	-	-	-	-	-	-	-	-

8.13. táblázat: az 5-ös számú tesztalany eredményei az őszinte válaszok esetén

A táblázat alapján a vizsgálat során nem tapasztalható jelentős eltérés a *kontroll* kérdésekre adott válaszokhoz képest a *vizsgálandó* kérdésekre adott válaszokban.

Hazugságra utaló jelek nem voltak tapasztalhatók jelentős mértékben.

9. Értékelés

9.1. Backend alkalmazás

A dolgozat célja az volt, hogy megalkosson egy olyan programot, amely különböző képfeldolgozási eljárások segítségével képes észlelni a hazugságra utaló jeleket egy adott vizsgálandó személy arckifejezései alapján.

A megvalósított program képes felmérni egy adott személy mimikai stílusát normális körülmények között, és összehasonlítani azt azokkal az esetekkel, melyek során lényeges eltérés tapasztalható. Ezek alapján a lényeges eltérések alapján a program képes megállapítást tenni azzal kapcsolatban, hogy az adott videóban a vizsgált személy reakcióból ki lehet-e szűrni bizonyos hazugságra utaló jeleket.

Az alkalmazás egyik fő célja az volt, hogy igyekezzen az üzleti logika felhasználhatóságát platformfüggetlen módon, különböző eszközök széles spektrumán elérhetővé tenni. Ezt sikerült elérni a Rest API elérési pontok biztosításával.

A feldolgozáshoz minimális felhasználói szintű konfiguráció szükséges, az összeállított eredmények pedig a JSON formátumnak köszönhetően könnyen értelmezhetőek és megjeleníthetőek tetszés szerint.

9.2. Frontend alkalmazás

Az alkalmazás elsődleges célja az volt, hogy egy olyan modern és interaktív felületet biztosítson a felhasználóknak, ami megfelel napjaink elvárásainak. A felület kellően letisztult és átlátható ahhoz, hogy a rendszerrel kapcsolatos előismeretek nélkül is könnyedén kezelni lehessen.

A videók feldolgozás előtti konfigurálása átlátható és egyszerű módon lett megadva, a szakaszok megjelölésének elősegítésére létrejött segédkomponensek segítségével az előfeldolgozás gyorsan és könnyedén megoldható.

Az eredmények megjelenítése a felhasználó számára átlátható, könnyen kezelhető és kellően nagy részletességgel történik. A részeredmények alakulása egyszerűen nyomon követhető az eredményjelző grafikonok segítségével, amik lehetőséget biztosítanak a felhasználó számára a megjelenített adatok szűrésére és kombinálására.

Egy-egy vizsgálat eredményeinek letöltése a későbbi feldolgozás nélküli megjelenítésért egyszerűen megtehető, a felhasználói felület jelentős része adatmegjelenítési célt szolgál.

Az alkalmazás minimális függőségekkel rendelkezik a backend szerver irányába: gyakorlatilag az API végpontokon és a fogadott adatok típusain kívül egyéb függőségek kialakítására nem került sor.

9.2.1. További lehetőségek

A backend szerver API elérési pontjainak köszönhetően a rendszer könnyedén megvalósítható bármilyen más platformon is, amely képes az alapvető HTTP protokoll szerinti kommunikációra.

Mivel a mai világban már alapvető felszereltségnek számítanak az okostelefonok, és a legtöbb ember ezek segítségével készíti a különböző multimédiás felvételeit, ezért kézenfekvő lenne például egy olyan frontend alkalmazás megvalósítása, amely a mobil eszközökön is lehetővé tenné a videók feldolgozását. Ehhez elegendő lenne pusztán egy egyszerű videó konfigurációs és egy eredménymegjelenítő felületet megvalósítani.

9.3. Tesztelések

A kész program kétféleképpen lett tesztelve.

Az automatikus Unit Teszteknek köszönhetően megállapítható, hogy a program a működését tekintve az elvártaknak megfelelően működik. Az automatizált teszteken nem valódi hazugságvizsgálat került megvalósításra, hanem a megfelelő funkcionalitás működésének ellenőrzése. A tesztek alátámasztották, hogy a program az elvi működésének eleget tesz és a különböző funkciók képesek együttesen az elvárt eredmények összeállítására és a megfelelő következtetések levonására.

A valódi hazugságvizsgálat működését manuális tesztek támasztják alá. Ezek a tesztek egyértelműen megmutatták, hogy a rendszer nem csak elvi síkon működik, de éles helyzetben is, konkrét személyek konkrét reakcióit elemezve képes a rendszer felismerni a hazugságra utaló jeleket, amennyiben vannak ilyenek. Mivel a tesztek során nem csak a hazugság, de az igazmondás is tesztelésre került, méghozzá igen kedvező eredményekkel, ezért kijelenthető, hogy tapasztalati úton a rendszer működése sikeresnek bizonyult.

9.4. Tapasztalatok összegzése

A dolgozat tervezési fázisa során többféle megvalósítási lehetőség és koncepció felmerült a program szerkezetét és funkcionalitását illetően. Miután az összes szükséges funkció és technikai elvárás megvalósításra került, beigazolódni látszott, hogy a kezdeti elképzelések egy egészen jó irányba mutattak.

Bár a kezdeti implementáció során felmerültek olyan nem várt tényezők, amelyek miatt szükséges volt a fejlesztési folyamatok többszöri újra tervezésére, összességében a munkálatok nem ütköztek olyan technikai akadályba, amelynek az elhárítása ne lett volna kisebb-nagyobb refaktorálások árán megoldható.

Az elvárás az volt, hogy egy megfelelően elkészített vizsgálandó videó alapján hazugságra utaló jeleket lehessen találni a videóban szereplő személy mimikai reakcióiból kiindulva. Mivel ezt sikerült elérni, és a manuális tesztek megerősítették a rendszer működésének sikerességét, ezért kijelenthető, hogy a kész szoftver eleget tesz a jelen dolgozatban felállított elvárásoknak.

A hazugságvizsgáló működik, ám – a hagyományos poligráfos vizsgálóhoz hasonlóan – nem átverhetetlen. Bár a projekt céljai között nem szerepelt, hogy a programnak átverhetetlennek kellene lennie, vagy bármiféle védelmet kellene nyújtania a szándékos szabotálás ellen, mégis egy jó lehetőség lenne a kérdéskör részletesebb vizsgálata a jövőbeni fejlesztéseket során. Ilyen és hasonló továbbfejlesztési lehetőségekre érdemes a jövőben sort keríteni. Néhány hasonló ötlet kifejtésre kerül a dolgozatban, ezeket lásd a következő fejezetben.

10. Továbbfejlesztési lehetőségek

10.1. Hamis reakciók kiszűrése

A jelenlegi program képes észlelni a hazugságra utaló jeleket az adott vizsgálandó személy arckifejezései alapján, képfeldolgozási módszerek segítségével. A program képes felmérni a bárki mimikai stílusát normális körülmények között, és összehasonlítani azt azokkal az esetekkel, melyek során lényeges eltérés tapasztalható, szintén normális körülmények között.

A mikro-kifejezések az adott személy érzelmi reakcióinak kivetüléseiként jönnek létre. Ezek szándékos manipulálása nem egyszerű, de nem is megoldhatatlan. Ez azonban a vizsgálat során fals adatok létrejöttéhez vezethet.

Jelenleg a rendszer nincs védve a reakciók szándékos elrejtése, befolyásolása ellen. Az egyetlen védekezési lehetőség ezekkel szemben, ha a kérdezőbiztos megpróbálja kiszűrni a vizsgált személy furcsa és egyértelműen szokatlan viselkedését a felvétel készítése közben.

Erre egy lehetséges megoldást nyújthatna, ha a rendszer maga próbálná meg kiszűrni az adott személy normális körülmények közötti viselkedésétől, reakciótól jelentősen eltérő magatartást. A határt azonban nehéz megtalálni aközött, hogy valaki valóban a hazugság okozta mentális erőfeszítés miatt produkál szokatlan, hazugságra utaló eredményeket, vagy pedig szándékosan, a vizsgálat szabotálása miatt befolyásolja a mimikai reakcióit. Szükséges lenne felismerni egy bizonyos mintát a vizsgált személy reakcióit illetően, és meghatározni egy olyan szintű eltérési küszöbértéket, amik a spontaneitást és a szándékosságot külön tudják választani egymástól. Egy-egy adott reakció során érdemes lehet ezeket az észleléseket külön jelezni, és a végeredményben a megfelelő helyeken feltüntetni.

10.2. Több személy vizsgálata egyszerre

Érdemes lehet kihasználni az arcfelismerés azon lehetőségét, hogy több, egy adott képkockán szereplő arc közül fel lehessen ismerni és meg lehessen különböztetni egymástól a különböző arcokat. Ennek a megoldásával elérhető lenne, hogy egy videóban egyszerre több ember reakcióit is meg lehessen vizsgálni.

10.3. Automatikus képfeljavítás

Amint az a korábbiakban tárgyalásra került, fontos tényezője a vizsgálatnak, hogy az arc megfelelő fényviszonyok mellett, lehetőleg árnyékolás nélkül, jól megvilágítva jelenjen meg. Annak érdekében, hogy ez a szempont ne legyen egy kritikus pontja a felvétel elkészítésének, érdemes lehet az egyes képkockákon a feldolgozás előtt bizonyos képfeljavító algoritmusokat lefuttatni, akár olyanokat is, amik kifejezetten az arcfelismerést és az arcfelismerés során a felismert arc vonásait és megjelenését képesek feljavítani.

Ezzel együtt az is egy jó optimalizálási lehetőséget jelentene, ha a képkockákon a felismert arc egy szűrő segítségével kiemelésre kerülne a környezetből, és a további műveletek csak a szükséges területen kerülnének alkalmazásra. Ennek köszönhetően el lehetne kerülni a képkockák azon részének a feldolgozását, amelyek teljesen irrelevánsok a vizsgálat szempontjából.

10.4. Egyéb faktorok vizsgálata

A hazugsággal járó mentális erőfeszítések eredményeképpen megjelenő érzelmi hangulatváltozásokat együtt járhat még a testhőmérséklet emelkedésével, valamint a hangszín bizonyos mértékű eltorzulásaival. Ezen faktorok figyelembevételével tovább javítható lehetne a hazugságra utaló jelek felismerésének a pontossága.

Irodalomjegyzék

- [1] Z. Wu, B. Singh, L. Davis, V. Subrahmanian: Deception Detection in Videos, (<https://ojs.aaai.org/index.php/AAAI/article/view/11502>), utoljára megtekintve: 2022. 10. 12.
- [2] S. Polikovsky, Y. Kameda, Y. Ohta: Facial Micro-Expression Detection in Hi-Speed Video Based on Facial Action Coding System (https://www.jstage.jst.go.jp/article/transinf/E96.D/1/E96.D_81/_article/-char/ja/), utoljára megtekintve: 2022. 10. 12.
- [3] „Facial Action Coding System [FACS]”, <https://www.paulekman.com/facial-action-coding-system/>
- [4] „FACS Action Units”, <https://www.cs.cmu.edu/~face/facs.htm>
- [5] „68 jellemző arcpontról”, <https://www.sciencedirect.com/science/article/pii/S0263224121006527>
- [6] N. Michael, M. Dilsizian, D. Metaxas, J. K. Burgoon: Motion Profiles for Deception Detection Using Visual Cues (https://link.springer.com/content/pdf/10.1007/978-3-642-15567-3_34.pdf), utoljára megtekintve: 2022. 10. 12.
- [7] J. Zhang, S. I. Levitan, J. Hirschberg: Multimodal Deception Detection using Automatically Extracted Acoustic, Visual, and Lexical Features (http://www.cs.hunter.cuny.edu/~slevitan/papers/interspeech_multimodal_deception_2020.pdf), utoljára megtekintve: 2022. 10. 12.
- [8] M. G. Frank, M. A. Menasco, M. O’Sullivan: Human Behavior and Deception Detection (<https://www.cl.cam.ac.uk/~rja14/shb10/frank2.pdf>), utoljára megtekintve: 2022. 10. 12.
- [9] J. Masip: Deception detection: State of the art and future prospects (<https://www.psicothema.com/pdf/4376.pdf>), utoljára megtekintve: 2022. 10. 12.
- [10] H. H. Thannoon, W. H. Ali, I. A. Hashim: Design and Implementation of Deception Detection System Based on Reliable Facial Expression. *Journal of Engineering and Applied Sciences*, vol. 14, no. 15, pp. 5002-5011, 2019
- [11] S. H. Abd, I. A. Hashim, A. S. Abdulhadi: Optimized Action Units Features For Efficient Design of Deception Detection System, (<https://www.ijict.edu.iq/index.php/ijict/article/view/160>), utoljára megtekintve: 2022. 10. 12.
- [12] C. M. Hurley, M. G. Frank: Executing Facial Control During Deception Situations, (<https://www.buffalo.edu/content/dam/www/news/imported/pdf/July11/FrankHurleyLyingStudy.pdf>), utoljára megtekintve: 2022. 10. 12.
- [13] T. Baltrušaitis, A. Zadeh, Y. C. Lim, L-P. Morency: OpenFace 2.0: Facial Behavior Analysis Toolkit. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2018

Ábrajegyzék

4.1. ÁBRA: AZ ARC FELBONTÁSA 12 RÉGIÓRA	13
4.2. ÁBRA: A SZEM-RÉGIÓ VIDEÓKOCKA KIVONATA EGY 200 FPS VIDEÓSZEKVENCIÁBÓL	13
4.3. ÁBRA: AZ ARC, VALAMINT A KÉZ ÉS A FEJ FORMÁJÁNAK LEKÖVETÉSE	15
4.4. ÁBRA: AZ ARC 79 JELLEMZŐ PONTJA, VALAMINT EZ EZEK ALAPJÁN MEGBECSÜLT FEJTARTÁS ÉRTÉKE LÁTHATÓ.....	16
6.1. ÁBRA: A VIDEÓ SZERKESZTÉSÉNEK VÁZLATA. LÁTHATÓ A VIZSGÁLNI KÍVÁNT SZEMÉLY, VALAMINT AZ IDŐSÁVON TÖRTÉNŐ SZAKASZMEGJELÖLÉS.	32
6.2. ÁBRA: A VIDEÓFELDOLGOZÁS FOLYAMATA NAGY VONALAKBAN A FELDOLGOZÁS MEGKEZDÉSÉTŐL AZ EREDMÉNYEK MEGTEKINTÉSÉIG.	32
6.3. ÁBRA: A KLIENS ÉS SZERVER OLDAL, VALAMINT AZOK FŐBB KOMPONENSEINEK KAPCSOLATA.	35
6.4. ÁBRA: SZEKVENCIA DIAGRAM A PROGRAM EGYES LÉPÉSEIRŐL A FELHASZNÁLÓ SZEMPONTJÁBÓL.....	36
6.5. ÁBRA: A FEJLESZTÉS TERVEZETT IDŐBEOSZTÁSA	38
7.1. ÁBRA: A 68 JELLEMZŐ ARCPONT	40
7.2. ÁBRA: AZ ACTION UNITOK KIÉRTÉKELÉSÉNEK A SZINTJEI	44
7.3. ÁBRA: A VIDEÓ ELŐKÉSZÍTÉSE A FELDOLGOZÁSHOZ	46
8.1. ÁBRA: KÉPKOCKÁK A TESZTELÉSHEZ MODELLEZETT ARC ANIMÁCIÓKBÓL.....	50
8.2. ÁBRA: A TESZT OSZTÁLY ÉS A TESZTESETEK DEFINIÁLÁSA	51

Táblázatjegyzék

3.1. TÁBLÁZAT: NÉHÁNY ACTION UNIT PÉLDAKÉNT	7
4.1. TÁBLÁZAT: HASONLÓ RENDSZEREK ÖSSZEHASONLÍTÁSA	14
4.2. TÁBLÁZAT: A HAZUGSÁG ESETÉN JELLEMZŐ ACTION UNITOK	27
8.1. TÁBLÁZAT: A TESZT EREDMÉNYE A KONTROLL SZAKASZ ÉS AZ ELSŐ VIZSGÁLANDÓ SZAKASZ ESETÉN	51
8.2. TÁBLÁZAT: A TESZT EREDMÉNYE A KONTROLL SZAKASZ ÉS A MÁSODIK VIZSGÁLANDÓ SZAKASZ ESETÉN ..	52
8.3. TÁBLÁZAT: A TESZT EREDMÉNYE A FELDOLGOZÁS ELLENŐRZÉSÉRE	52
8.4. TÁBLÁZAT: AZ 1-ES SZÁMÚ TESZTALANY EREDMÉNYEI A HAZUGSÁGTARTALMÚ VÁLASZOK ESETÉN.....	54
8.5. TÁBLÁZAT: AZ 1-ES SZÁMÚ TESZTALANY EREDMÉNYEI AZ ŐSZINTE VÁLASZOK ESETÉN	54
8.6. TÁBLÁZAT: A 2-ES SZÁMÚ TESZTALANY EREDMÉNYEI A HAZUGSÁGTARTALMÚ VÁLASZOK ESETÉN	54
8.7. TÁBLÁZAT: A 2-ES SZÁMÚ TESZTALANY EREDMÉNYEI AZ ŐSZINTE VÁLASZOK ESETÉN	54
8.8 TÁBLÁZAT: A 3-MAS SZÁMÚ TESZTALANY EREDMÉNYEI A HAZUGSÁGTARTALMÚ VÁLASZOK ESETÉN.....	55
8.9. TÁBLÁZAT: A 3-MAS SZÁMÚ TESZTALANY EREDMÉNYEI AZ ŐSZINTE VÁLASZOK ESETÉN.....	55
8.10. TÁBLÁZAT: A 4-ES SZÁMÚ TESZTALANY EREDMÉNYEI A HAZUGSÁGTARTALMÚ VÁLASZOK ESETÉN	55
8.11. TÁBLÁZAT: A 4-ES SZÁMÚ TESZTALANY EREDMÉNYEI AZ ŐSZINTE VÁLASZOK ESETÉN	55
8.12. TÁBLÁZAT: AZ 5-ÖS SZÁMÚ TESZTALANY EREDMÉNYEI A HAZUGSÁGTARTALMÚ VÁLASZOK ESETÉN	56
8.13. TÁBLÁZAT: AZ 5-ÖS SZÁMÚ TESZTALANY EREDMÉNYEI AZ ŐSZINTE VÁLASZOK ESETÉN.....	56