

TDK dolgozat
Kalmár György

Alacsony költségű, intelligens, mobil alapú otthoni felügyelő rendszer

Kalmár György
II. évf. mérnök informatikus Msc.

Témavezető: Dr. Balázs Péter
adjunktus
SZTE TTIK Képfeldolgozás és Számítógépes Grafika Tanszék

Absztrakt

Napjainkban a társadalmi problémák közé tartozik a nagyszámú betörés, rablás, amelyek megakadályozása, kivizsgálása nagyon nehéz feladat és az ismeretlen tettes után való hajsza is nem ritkán eredménytelenül zárul. Megvédeni otthonunkat igencsak költséges dolog lehet, hiszen a modern riasztóberendezések drágák, felszerelésükhöz, üzemeltetésükhöz sokszor szakemberre van szükség. Ugyanakkor napjainkban lassan mindenki zsebében ott lapul egy okostelefon, amely meglehetősen sok lehetőséget biztosít arra, hogy védelmi funkciót is bízunk rá. Továbbá, legtöbbünk zsebéből egyszer kikerülnek ezek a készülékek, elgyengült akkumulátor, meghibásodott, eltört kijelző okán vagy elavultnak számítanak bizonyos idő elteltével.

Kutatásom célja egy olyan alacsony költségű, mindenki számára elérhető, könnyen beüzemeltető és használható, mobil alapú védelmi berendezés megalkotása, amely képes megtanulni, majd később felismerni a védelmezett terület tulajdonosának arcát, felhasználva a ma már olcsón beszerezhető új, illetve használaton kívül esett régi okostelefonokat.

A megalkotott rendszer mindösszesen egy olcsó okostelefonból áll, amely a védelmezni kívánt területet figyeli. Minimális elvárás, hogy rendelkeznie kell egy kamerával és egy mikrofonnal, illetve képesnek kell lennie Android 4.0-s vagy újabb verziót futtatni.

A kifejlesztett felügyelő rendszer jelenleg egy felhasználó arcát képes megtanulni, majd azt a későbbi megjelenés esetén felismerni. Mindennek alapja az eszköz kamerájának folyamatos üzemeltetésével a védelmezett terület pásztázása. Időközben a rendszer valós időben képes emberi alak keresésére az elkészült képeken. Amennyiben ez a keresés egyszer sikerrel zárult, akkor egy dinamikus arckövető eljárással, szintén valós időben képes arcot keresni és arról nagy felbontású képeket készíteni. A felügyelt tanulási fázisban ezen képeken csak a később felismerni kívánt személy szabad, hogy szerepeljen, ez a fázis egy napig tart. A második nap kezdetével a rendszer éles, riasztó üzemben kezdi meg működését. Ilyen esetben, ha embert, majd arcot talál a képen, akkor a referenciával való összehasonlítás után döntést hoz, hogy a már ismert arc szerepel-e a képen. Mindezt az arcról kivont mikrojellemzők korrelációjával éri el, amely jellemzőket a tanulás során és a felismerés során is kivon a referencia ill. a kérdéses arcról is. A döntést egy hang alapú jelszóellenőrzés is támogatja, ha a döntés kétséges kimenetelű lenne. Negatív döntés esetén a rendszer azonnal értesít egy előre megadott telefonszámot vagy e-mail címet egy MMS ill. e-mail üzenet segítségével, amely tartalmazza az ismeretlen arcról készült képet.

A képfeldolgozási (alak keresés), mesterséges intelligenciai (tanulás, döntéshozás), optimalizálási (processzor- és tárigény) és jelfeldolgozási (hang alapú jelszó felismerés) alapokon nyugvó rendszer tesztelése során megbízhatónak mondható teljesítménnyel volt képes védelmezni a kívánt területet.

Tartalomjegyzék

Absztrakt	2
Tartalomjegyzék.....	3
1. Bevezető.....	4
2. A megvalósítandó rendszer	6
2.1. A rendszerrel szemben támasztott követelmények	7
2.2. A rendszer működésének rövid áttekintése	8
3. A rendszer alapkövei.....	9
3.1. Felhasznált módszerek	9
3.2. Emberi alak detektálása.....	10
3.3. Arc és arcrészek detektálása.....	12
3.4. Kiegészítő funkciók.....	15
3.4.1. Mozgásérzékelés	15
3.4.2. Hangalapú jelszóellenőrzés	17
3.4.3. Dinamikus arckövetés	20
3.4.4. Saját kamera kezelő osztály	21
4. A rendszer működési fázisainak vizsgálata.....	21
4.1. Tanuló fázis	21
4.2. Felügyelő fázis	23
5. A rendszer tesztelése	28
6. Elért eredmények, összefoglalás	31
Irodalomjegyzék.....	32

1. Bevezető

Napjaink egyik kiemelkedő társadalmi problémája a sok betörés, lopás, rablás és ezek megakadályozásának elérése. A rendvédelmi szervek sokszor tehetetlenek az ismeretlen tettes elleni hajsza során. A Központi Statisztikai Hivatal adatai szerint, az elmúlt években messze a vagyon elleni károkozásért indított perből volt a legtöbb, 250 000 körül évente, amely nagy számhoz jelentős mértékben járul hozzá az otthoni betörések száma. Ezen bűncselekmények megelőzése illetve nyomra vezetése igencsak nehéz feladat és a közeljövőben nem is mutatkozik rá esély, hogy ez megváltozna. Habár elérhetőek ma már széles körben otthonilag használható riasztóberendezések, ezek beszerelése sokszor túl költséges egy család számára. További hátrányai is vannak a ma használatos védelmi rendszereknek. Először is, hogy csak egyetlen egy ismert, rövid kódon alapulnak. Ezen kód ismeretében bárki hatástalaníthatja a rendszert. Az is előfordulhat, hogy amennyiben csak kis időre hagyjuk el otthonunkat, vagy éppenséggel a hátsó udvarban tevékenykedünk, nem aktivizáljuk a riasztót, hiszen ekkor nem is számítunk veszélyre. A másik hátrányuk tehát a kényelmetlenség, amely tény igenis nagy szerepet kap sajnos a mai Ember megítélésében. Hazaérve, hangos riasztás közepette szorulunk rá, hogy gyorsan beírjuk a kódot, hogy ne hogy téves riasztást adjunk ki. Jobb - és drágább - esetben ez a riasztás eljut egy központi szervhez, amely talán elindít valamilyen gyors vizsgálatot, hogy valóban betörésről van-e szó. Összefoglalva tehát, elmondhatjuk, hogy a mai riasztóberendezések többnyire drágák, és mégis alacsony tudásúak, kényelmetlenek.

Kevés ember van, aki fel tudja dolgozni a technika napjainkban való fejlődésének hihetetlen ütemét. Nem is sejtik, hogy mekkora műszaki csoda lapul már zsebükben. Egy okostelefon, amely ma már mindenki számára a hétköznapiak része, a kommunikáció és multimédia egész tárházát nyújtja egy tenyérnyi kis eszközben. Magas processzor teljesítmény, sok memória, kamera, mikrofon, internet hozzáférés, telekommunikációs funkciók mind-mind egy kicsiny és ma már olcsó eszközben. Sokak ezen termékeket kapcsolattartásra, játszásra, szórakozásra használják, majd egy bizonyos idő után, mikor már elavultnak gondolják, vagy gondoltatják velük mások, eldobják, elteszik a fiókba és örökre kihasználatlanok maradnak. Továbbá mi történik egy telefonnal, ha meghibásodik az akkumulátora és már csak hálózati tápról tud működni? Mi történik, ha elejtik és betörik a képernyője? Igen, a válasz minden feltett kérdésre, hogy eldobják, ezzel környezetvédelmi

kérdéseket is feszegetve. A javíttatásuk az eredeti eszköz árának 70-80%-át is elérheti. Kinek jutna eszébe ezeket otthoni felügyelő rendszerként használni..?

Kutatásom célja, egy olyan mindenki számára elérhető, olcsó, intelligens otthoni felügyelő rendszer kivitelezhetőségének a vizsgálata, amely képes megtanulni a rendszer használóját, majd későbbi ismételt megjelenés esetén felismerni azt. Ezen rendszer elsősorban az elavulttá vált, sérült okostelefonok segítségével járulna hozzá a hétköznapi emberek biztonságához.

Az elgondolás azon tény felismerésén alapult, hogy jelen van rengeteg olyan kihasználatlan eszköz, amely páratlan multimédiás és telekommunikációs funkciókat biztosít. Adódott a kérdés, hogy fel lehetne-e ezeket használni esetleg olyan célra, amely a társadalom javát szolgálná? Kamerája révén egy okostelefon folyamatosan meg tudja figyelni a környezetét, mikrofonja, hangszórója révén távolról is kapcsolatot tud teremteni a felhasználójával, az interneten akkor is üzeni tud használójának, ha az távol van tőle. Ezen lehetőségek ismeretében felvetődött a lehetőség, hogy alkalmas egy okostelefon otthoni védelmi célokra.

Ma már az okos eszközök nagyon nagy hányadán Android alapú operációs rendszer fut. Ez lehetővé teszi, hogy az egyes eszközök különbségeit a felsőbb rétegekben elrejtjük. Tehát egy elkészült, védelmi célokat biztosító alkalmazás az összes Android alapú eszközön futni képes. Ezen megfontolásból a kutatásom megvalósításához én is Android alapú eszközt választottam és az elkészült alkalmazás is Java nyelven, Androidos környezetben készítettem el.

A dolgozat következő fejezeteiben szót ejtek röviden a rendszerrel szemben támasztott követelményekről, a kutatás során felhasznált módszerek elméleti háttéréről, a megvalósított rendszer elemiről, működéséről és a kész rendszer tesztelése során gyűjtött tapasztalataimról.

2. A megvalósítandó rendszer

A kutatásom célja egy olyan rendszer megalkotása, amely felismeri a használóját, és amennyiben nem ő jelenik meg a képen, értesít egy előre megadott e-mail címet vagy telefonszámot e-mail ill. MMS üzenet formájában, amely üzenet tartalmaz egy képet az ismeretlen arcról. Az üzenet megtekintésével a használó dönthet, hogy értesíti a rendőrséget vagy sem.

Kezdeként, a jelen dolgozatban tárgyalt rendszer korlátozott abból a szempontból, hogy csak egy arc megtanulására képes. Ezen korlátozás kihatással van az alkalmazott módszerre. A mesterséges intelligencia megalkotásánál figyelembe kell venni, hogy a döntéshozás bináris jellegű, tehát megtanult arc, vagy nem az szerepel egy képen, vagy esetleg több felismerendő arcból kell választani, hogy azok közül valamelyik szerepel-e a képen vagy sem. Nyilvánvaló, hogy a második feltételezés sokkal bonyolultabb megoldást kíván és az alkalmazható módszerek komplexitása is nagyobb. Azonban ne gondoljuk azt, hogy az első felvetés már egy létező dolog. Az arcfelismerés még statikus képeken sem tökéletesen megoldott probléma. Esetünkben ehhez még hozzájárul az, hogy nem mondhatjuk meg az aktuálisan vizsgált arc birtokosának, hogy folyamatosan forduljon szembe a kamerával, hogy megfelelő képeket készíthessünk.

Egy személy felismerésénél is adódnak nehézségek. Elsősorban az intraperszonális variancia, vagyis hogy egy személy saját magához képest mennyire változatos, jelenti a fő nehézséget. Gondoljunk csak bele, mennyire más a kép (egy számítógép számára), ha nevetünk, mosolygunk, beszélünk, ásítunk, stb. Ezen tény leküzdéséhez sok képre és a döntéshozás során statisztikai alapok használatára mindenképp szükség van.

Nehezítő tényező továbbá, hogy esetünkben a felismerendő személyek mozognak, így csak homályos képeket készíthetünk, ha figyelembe vesszük, hogy az okostelefonok kamerája nem a legkifinomultabb optikai rendszer és nem is a gyors képkészítésre optimalizált.

Összefoglalva a következő célok elérését tűztem ki: a rendszer fusson egy olcsó új, vagy használt okostelefonon, kezeléséhez, beüzemeléséhez ne legyen szükség szaktudásra, a képkészítés és intraperszonális variancia által okozott sok bizonytalanság ellenére próbálja meg minél hatékonyabban felismerni a használóját.

2.1. A rendszerrel szemben támasztott követelmények

A rendszer, mivel felügyelő rendszerről van szó, sok speciális funkcióval szembeni követelményt is ki kell, hogy elégítsen. A fő funkciókat az alábbiakban részletezem.

- a rendszer képes legyen egy olcsó okostelefonon is futni, minimális elvárás, hogy az rendelkezzen egy hátoldali kamerával, egy mikrofonnal, és képes legyen Android 4.0-s vagy újabb verziót futtatni
- a rendszer képes legyen egy arc megtanulására, ezt egy egy napos, felügyelt tanítási fázisban kell, hogy megtegye
- a tanítási fázis után képes legyen egy megjelenő arcról eldönteni, hogy az a megtanult arc (személy) vagy sem
- mindezen felismerő funkciókat a lehető leghatékonyabban és legmegbízhatóbban végezze
- minimalizálja a fals pozitív döntések számát, hiszen ez azt jelentené, hogy ismeretlen arcot ismertnek feltételezne
- ha az arc felismerése fizikai korlátok miatt nem is hajtható végre, akkor is próbáljon meg döntést hozni az illető kilétéről

Ezen alapvető funkciók nagyon általánosak és iránymutatóak a rendszer felépítését illetően. Azon elvárás, hogy rendelkezzen kamerával és mikrofonnal egy okostelefon csak a rend kedvéért szerepel itt, ezek ma már alapvető tartozékai egy ilyen eszköznek. Az Android 4.0-s verzió iránti elvárás sem mondható szigorúnak. A 4.0-s verzió viszonylag nagy ugrást jelentett az előző, széles körben elterjedt 2.3-as verzióhoz képest. Jónéhány éve már csak újabb verziót futtató telefonok jelennek meg.

A második követelmény, miszerint egy nap alatt meg kell tanulni egy arcot, azt jelenti, hogy ezen a napon csak a később felismerni kívánt felhasználó szabad, hogy a kamera elé kerüljön. Ha biztosítjuk, hogy csak ő jelenik meg a kamera előtt, akkor a készített képeket fel lehet címkézni ismert arcokról készült képnek, tehát a tanítás felügyelt módon végezhető.

A harmadik követelmény a rendszer fő funkciójáról szól, a védelmi feladatot ellátó tevékenységről.

A negyedik elvárás első fele, miszerint a felismerő funkciókat minél hatékonyabban kell elvégezni, azt jelenti, hogy olyan algoritmusokra van szükség, amelyek gyorsan futnak, nem terhelik túlságosan az eszköz memóriáját és processzorát. A második rész, miszerint ezt

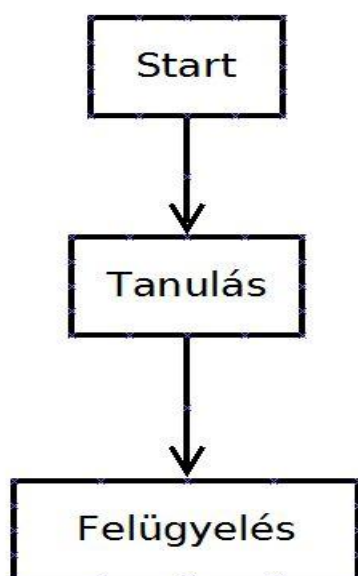
megbízhatóan kell elvégezni, arra utal, hogy a futó szoftver meghibásodás, leállás nélkül tudjon folyamatos üzemben védelmi funkciót ellátni.

Az ötödik elvárás egy, később mérőszámként is használható jellemzővel szemben támaszt elvárásokat. A rendszer működése szempontjából az egyik legfontosabb, hogy olyan arcra, amely ismeretlen, soha ne mondja azt, hogy ismert. Az ellenkező eset, a fals negatív elfogadható, következménye mindössze annyi, hogy kapunk egy e-mailt vagy MMS-t a saját arcunkkal csatolmányként.

A hatodik, és egyben utolsó elvárás kiegészítő funkció iránti követelményt szab meg. Abban az esetben, ha az arcról lehetetlen használható képet készíteni, pl.: rossz fényviszonyok, gyors mozgás, eltakart arc miatt, akkor a felügyelő rendszer ezt érzékelve, más interakciót használva próbáljon meg mégis valamiféle döntést hozni arról, hogy szükséges-e riasztást kiadni. Jelen esetben ez egy hangalapú jelszóellenőrzés formájában valósul meg.

A követelmények megismerése után, rátérek a rendszer működésének konkrét tárgyalására. Megvizsgálva minden lépésben a támogató elméleti hátteret, a fennálló problémákat, a megvalósítás módszerét és az egyéb javítási lehetőségeket.

2.2. A rendszer működésének rövid áttekintése



1. ábra: A rendszer működésének rövid folyamatábrája

A rendszer beüzemelésének kezdete, hogy az adott okostelefonra fel kell telepíteni az alkalmazást. Elindítva a programot, az egyből egy tanulási fázisba fog. Ezen fázis során tanulja meg a később felismerni kívánt arcot, illetve a hangalapú jelszó azonosításhoz a referencia jelszót. A tanulás végeztével a rendszer egyből riasztó, felügyelő üzembe lép. Ez a működés az alkalmazás leállításáig tart. Ha kívánatos, a rendszer működhet úgy is, hogy leállítás, majd ismételt bekapcsolás során megvizsgálja, hogy rendelkezik-e már a megtanulni kívánt adatokkal, és amennyiben igen, a tanulási fázis ki is hagyható. Ez abban az esetben hasznos, ha a telefont otthonlét esetén másra is használni szeretnénk, és távozáskor visszatesszük a helyére, hogy védelmi eszközként működhessen.

3. A rendszer alapkövei

3.1. Felhasznált módszerek

Az elkészítendő program egyik fő sajátossága, hogy okostelefonon, mégpedig Android operációs rendszerrel rendelkező eszközön fog futni. Nyilvánvaló, hogy amennyiben ilyen mértékű képfeldolgozási, gépi tanulási feladatokba akarunk belevágni, érdemes egy, már jól megírt, tesztelt és támogatott, megbízható elméleti alapokon nyugvó algoritmusokat alkalmazó rendszert használni. Jelen esetben, az alkalmazás felhasználja az OpenCV (Open Source Computer Vision) [1] nevű könyvtár által nyújtott funkciókat. Ez rengeteg képfeldolgozási algoritmust kínál. Egyik nagy előnye, hogy bár C++-ban íródott, Java interfésszel is rendelkezik. Továbbá becsomagolták Androidos környezetben használható formába is. A Java-s környezetben meghívott függvények, egy JNI (Java Native Interface) csatoláson keresztül natív kód futását teszik lehetővé. Maga a program Java nyelven íródik, amit az Android virtuális gépe futtat majd. A műveletigényes, bonyolult képfeldolgozási algoritmusok a viszonylag lassabb Java környezettől elkülönülten egy gyors, C++ kódból fordított natív kód futtatásával kerülnek végrehajtásra. Ezen kódok több GPU-ra is optimalizáltak, amelyek meglétét azonban nem lehet feltételezni minden okostelefonban, tehát ennek kihasználhatóságára nem építettem. Az OpenCV4Android, amely az OpenCV Androidhoz készített változata szinte minden eljárást tartalmaz, amit az OpenCV is, az arcfelismeréshez (nem arc detektáláshoz) szükséges funkciókat például kihagyták a csomagolás során. Egyébként sem valószínű, hogy a kitűzött cél eléréséhez megfelelőek lennének ezek az egyébként jól bevált módszerek [8]. Tanításukhoz rengeteg képre van szükség, amelyet egyszerre várnak el a memóriában, Ez nem mindig kivitelezhető. Például, ha megfelelő megbízhatósággal szeretnénk használni ezeket a módszereket, akkor a tanítás során több, mint 500, arcról készült képet kellene a memóriában tárolni és még minimum ennyi negatív példát tartalmazót is, vagyis összesen 1000 képet. 100x100-as képmérettel, 1 darab, 8 bites csatornával számolva is ez 10MB. Ma már lehetséges, hogy feltételezhetjük, hogy ennyi memória szabadon rendelkezésre áll, ezért a kutatásom folytatásaként majd ezen módszereket is szeretném megvizsgálni. Jelen OpenCV4Android kiadásban ezen arcfelismerő algoritmusok nem is érhetők el, illetve a céloom egy kevés képből is működni képes rendszer megalkotása. Talán ebben később segítséget nyújt majd az Android is, amely maga is tartalmaz arcfelismerő és detektáló algoritmusokat. Ezeket jelen kutatásban is teszteltem, de sebességük és pontosságuk sajnos elmaradt az OpenCV által nyújtott lehetőségektől.

3.2. Emberi alak detektálása

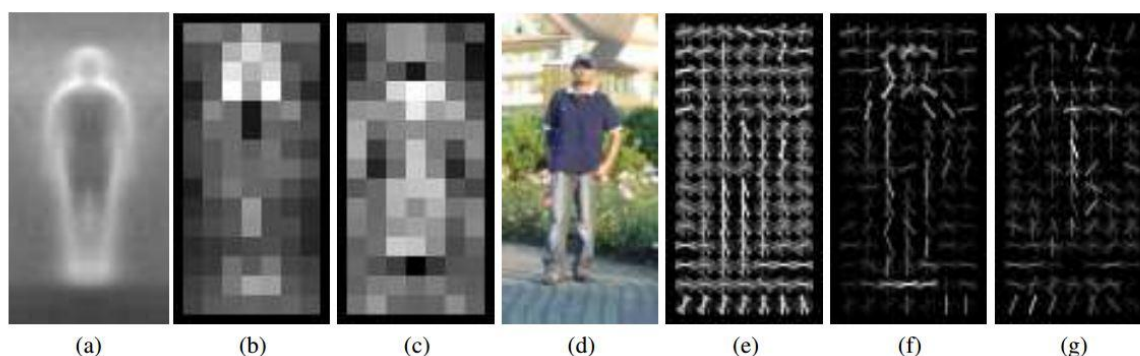
Az emberi alak megbízható és gyors felismerése az egyik, és talán a legfontosabb alappillére a felügyelő rendszernek. Ennek köszönhető, hogy a képen megjelenő személy detektálásra kerül, így a rendszer értesül arról, ha a megfigyelt térségben valaki tartózkodik. Alapvető szükséglet ez az arcfelismerés helyes működéséhez is, hiszen arcot csak abban az esetben érdemes keresni, ha emberi alak is szerepel a képen. Továbbá akkor is jelen lesz ember formájú objektum, ha az arc külső, véletlen vagy szándékosan befolyásolt tényezők (pl.: álarc) folytán nem is detektálható.

A szakirodalom kutatása során sok módszerrel találkoztam, amelyek mindegyike igencsak jó eredményeket ért el ezen a téren. Közülük kiemelkedőnek éreztem a 2005-ben, Navneet Dalal és Bill Triggs által publikált HOG leíróval működő módszert. [2] A HOG, (*Histograms of Oriented Gradients*) irányított gradiensek hisztogramjából álló leíró vektor. Lényegében az eljárás arról szól, hogy a beérkező képet cellákra osztjuk. A cellákon normalizálást hajtunk végre. Minden cellában kiszámoljuk a gradiensek irányát és nagyságát. A nagyság és kvantált irány alapján az egyes pixelek hozzájárulnak egy irányított gradiens hisztogram megalkotásához, ahol a hozzájárulás mértéke az adott irányhoz a gradiens nagyságának a mértékétől függ. A cellákon túl a képet nagyobb, átlapolódó régiókra is felosztjuk, amelyekben kontraszt normalizálást hajtunk végre, így megvilágítástól kevésbé függő HOG leírókat kapunk. A teljes képre kiterjedően összegyűjtjük az egyes cellák HOG leíróit és egy vektorra fűzzük őket össze, ez lesz a kép HOG leíró vektora. Ezeket a vektorokat egy SVM-be (*Support Vector Machine*) irányítjuk, amelynek lényege, hogy egy optimális hipersíkot találjon egy adott dimenziós térben, hogy a megtanulni kívánt osztályt (clustert) a lehető legjobban elkülönítse a többi adattól. Nyilvánvaló, hogy először felügyelt tanítás szükséges, mivel az SVM-nek meg kell mondani, hogy melyek azok a HOG leíró vektorok, amelyek embert tartalmazó képről származnak és azt is, hogy melyekhez nem tartozik ember. A 2. ábrán a szerzők által készített blokkvázlaton kísérhetjük nyomon a rendszer működésének menetét.[2]



2. ábra: A HOG jellemzőkön alapuló rendszer blokkvázlata [2]

A 3. ábrán megfigyelhetjük az algoritmus működését egy tesztképen.



3. ábra: A HOG alapú, emberi alak keresést megvalósító algoritmus egy tesztképen: (a) a tanítás folyamán a rendszer által elsajátított átlagos gradiens kép; (b) minden „pixel” a maximális pozitív SVM súlyt mutatja az adott blokkban (látható a váll és fej kiemelkedő szerepe); (c) mint (b), csak a negatív súlyokra; (d) egy tesztkép; (e) a számolt HOG leírók; (f) a HOG leíró súlyozva a (b) eredményével; (g) mint (f), csak (c)-vel súlyozva; [2]

A teljesség kedvéért megemlítem, hogy ennek a módszernek az eredményessége győzött meg arról, hogy a kutatás folyamán ezt szeretném használni emberi alak felismerésre. A megvalósíthatóság utáni kutatásban találtam meg az OpenCV-t, amelynek Androidos változata is tartalmazza a HOG alapú emberi alak felismerést. Sokat elmond, hogy egy 2005-ben publikált módszer már egy ilyen horderejű képfeldolgozási könyvtárban helyet kap, mint az egyetlen, kizárólag emberi alak felismerő algoritmus.

A megvalósított rendszer az OpenCV-ben meglévő, HOG leíróra támaszkodó eljárást használja emberi alak felismerésre. [3] Azonban ennek megfelelő használata nem olyan egyszerű. Habár az algoritmus igen megbízható, nagy méretű képekre, főleg gyengébb processzorteljesítmény esetén nagyon lassan fut le, végrehajtása akár több másodpercig is tarthat. Mivel elvárás, hogy az ember detektálás gyorsan, számszerűsítve nagyjából 10fps (*frame per second*) sebességgel fusson, ezért szükség volt egy saját kamerakezelő írására, amelyet majd a későbbiekben részletezek. A lényege az, hogy a feldolgozó már hardver szintről is kis méretű képeket kapjon és az alkalmazás futása során ez a méret tetszőlegesen módosítható legyen. Mivel a bejövő kép eleve kevés pixelt tartalmaz, ezért az embert detektáló algoritmust már ezen futtatva elértem a kívánt sebességet. A megbízhatóságáról annyit érdemes megemlíteni, hogy csak az emberi detektálást elemezve, nem történt olyan eset, hogy a kamera előtt álló személy nem váltott volna ki jelzést, tehát a fals negatív észlelések száma nem kimutatható. A fals pozitív találatok száma kevésbé jó eredményt mutat, előfordul, hogy nincs ember a képen, de mégis jelez a rendszer (főleg kabátok, pólók jelenléténél kifejezetten). A végső tesztelés során azonban ez sem mondható jellemzőnek, a

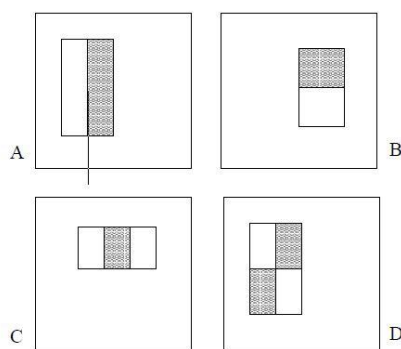
teljes rendszer vizsgálata során szinte sosem fordult elő, hogy nem tényleges emberi alakra jelezzen. A tesztelés során készített néhány tesztkép a 4. ábrán figyelhető meg.



4. ábra: Emberi alakot felismerő algoritmus tesztelése során készült tesztképek

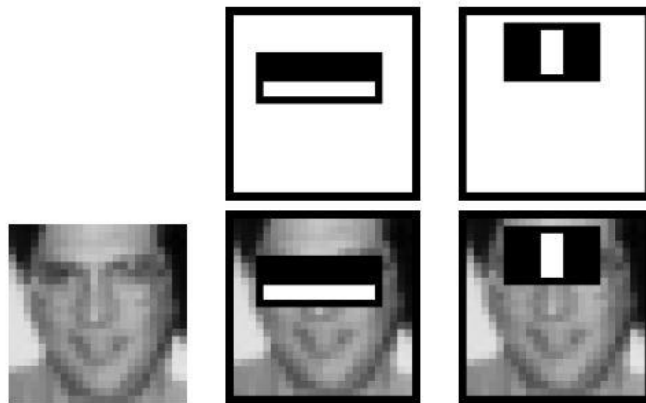
3.3. Arc és arcrészek detektálása

Az emberi alak megtalálásán túl a rendszer másik elengedhetetlen eleme az arcok megbízható és gyors detektálása, tekintve, hogy a végső cél ezek megtanulása és felismerése. Még az emberi alak többnyire egy általános formának mondható, addig az arc nagyon változatos kinézetű is lehet. Egy fehér bőrű, szemüveges, arcszőrzettel rendelkező férfi nagyon sokban különbözik, a jellemzők tekintetében, egy fekete bőrű nőtől. Ezért ez a feladat még ma sem megoldott probléma, viszont kielégítő teljesítményű eredmények természetesen születtek a témában. Az OpenCV az eddig publikált leggyorsabb és legrobustusabb arcfelismerő eljárást támogatja. Paul Viola és Michael Johns a cambridgei egyetem tudósai 2001-ben publikálták arcfelismerésre optimalizált módszerüket [4]. Ennek lényege, hogy úgynevezett Haar jellemzőket nyernek ki a képekből, majd ezekre építve gyenge osztályozókat hoznak létre, amelyeket egymás után fűznek. A beérkező képet egy új reprezentálási formába alakítják, az ún. integrál kép, amely egy adott (x,y) pozícióban megadja a pixeltől balra és felfelé lévő pixelek összegét. Ez a reprezentáció nagyon gyorsan számolható és a későbbi jellemzőkinyerés erre alapul. A használt jellemzők sem túl bonyolultak. Kijelölt régiók pixelösszegeinek a különbségeiről van szó. Három féle ilyen jellemzőt vesznek alapul (2,3,4 régiót használó), amelyeket az 5. ábrán figyelhetünk meg. [4]



5. ábra: Haar jellemzők: a fehér térrészek pixelösszegeit vonjuk ki a sötétekeiből [4]

Ezen jellemzők alapján gyenge osztályozókat hoznak létre, amelyekhez a legjobban szeparáló jellemzőt AdaBoost boosting algoritmussal választanak. A 6. ábrán pl. két jól működő jellemzőt láthatunk. [4] Megfigyelhető a sötétebb szemek és a sötét szemek közt elhelyezkedő világosabb orr egyértelműen kimagasló szerepe.



6. ábra: Az első és második kiválasztott jellemző [4]

Az algoritmusuk lényege, hogy mivel rengeteg jellemzőt kinyernek egy képből, azért először próbáljanak meg igen jól szeparálókat használni arra, hogy a lehetséges arc jelöltekről korán döntést tudjanak mondani, hogy lehet-e szó arcról vagy sem. Az egyes jellemzőkön alapuló gyenge osztályozókat egymás után fűzik (*cascade classifier*), hogy a végén erős osztályozót kapjanak (boosting). Minden eleme az osztályozó láncnak vagy elveti az aktuális jelöltet vagy továbbadja egy komplexebb osztályozónak. Azok a régiók a képről, amelyek eljutnak a lánc végéig, nagyon nagy valószínűséggel arcot tartalmaznak. A módszer előnye, hogy már a kezdeti lépésben kiszűrjük a biztosan nem arcot tartalmazó régiókat, így a magasabb számításigényű, komplexebb osztályozók csak az arra érdemes régiókra futnak le, lényegesen gyorsítva az arc detektálását. Ennek is köszönhető, hogy a publikáláskor ez volt a leggyorsabb módszer, a szerzők mérései alapján 15-ször gyorsabb, mint az addigi leggyorsabb algoritmus.

A felügyelő rendszerben én is ezt a módszert használtam, a megvalósítása elérhető az Androidos OpenCV változatban is. Előre tanított osztályozókat biztosít, amelyek nem csak arc, de sok más objektum keresésére is alkalmasak [5]. Ennek köszönhetően választottam az arcfelismerés módszeréül azt, hogy az arcról ún. mikrojellemzőket vonok ki és ezeket hasonlítom össze a később megjelenő arc jellemzőivel. A mikrojellemzők konkrétan a száj, az orr, a bal szem, a jobb szem, és a szempár, pontosabban az ezekről készült képrészeket. Detektálásukat ugyancsak Haar alapú kaszkádolt osztályozó segítségével érem el.

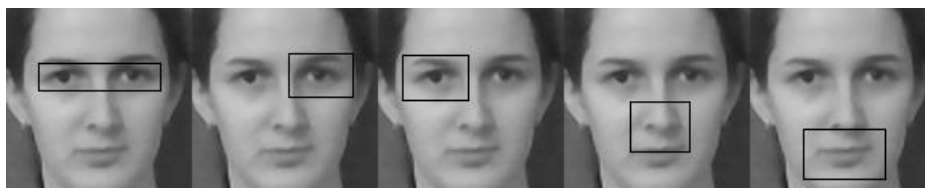
Az emberi alak felismerésénél látott függés a képmérettől itt nem annyira számottevő, azonban itt is hangsúlyt fektettem arra, hogy gyorsan, nagy felbontásban készíthessek képeket

az arcról. Ehhez egy később részletezett, dinamikus arckövető eljárást fejlesztettem ki, amely nagy felbontású képen teszi lehetővé arc detektálását több, mint 10fps sebességgel. Kiemelkedően fontos, hogy ez ténylegesen gyorsan működjön. Az érvek mellett szól, hogy kevés esélyünk van az arcról szembenálló képet készíteni, azon rövid időt kell kihasználni, amikor egy személy a mobileszköz felé fordul. Ebben a pillanatban érdemes minél több képet készíteni, hogy a hibák statisztikai alapon szűrhetőek legyenek. és majd ezeket a képeket a felvétel után feldolgozni, ha megvan egy előre adott maximális darab. A nagy felbontás is szükséges, hiszen csak akkor működhet bármiféle felismerő algoritmus is, ha rendelkezésre áll megfelelően sok részlet az arcról, amelyek a döntéshozást elősegítik. Néhány, detektálás során készült kép látható a 7. ábrán.



7. ábra: Arcdetektálás során készült képek

A későbbiekben tárgyalt arcfelismerő algoritmus működéséhez szükséges, hogy az arcról mikrojellemzőket vonjunk ki. Ez gyakorlatilag a száj, orr, szempár, bal és jobb szem detektálását jelenti egy arcon. Mondani sem kell, hogy fontos a gyors futás és megbízható működés. Minden arcról készült képen el kell ezt az öt detektálási folyamatot végezni, amit a már ismertetett Haar alapú osztályozó rendszer segítségével valósítok meg. A detektálás lépéseiről készült képeket a 8. ábrán tekinthetjük meg.



8. ábra: Arcrészek detektálása során készült képek

A Haar alapú objektum detektáló algoritmus tesztelése során a következő észrevételeket tettem. Elsősorban valóban nagyon gyors, a nagy felbontású képeken sokszorta gyorsabban fut le, mint a HOG alapú módszer. Nem is érzékeny annyira a képméret változására. Fals negatív jelzések tekintetében jónak mondható. Amennyiben kamera felé forduló arc jelenik meg a képen, a jelzés szinte sosem marad el. A fals pozitív észlelések száma itt is rosszabb, sokszor adódik olyan, hogy arcra nem is hasonlító objektumokra is jelez. Ennek kivédését majd a későbbiekben részletezett dinamikus arckövető eljárás megoldja. Az arcrészek hibás detektálásra is megoldott probléma, mivel a kívánt szervek helyzete megközelítőleg ismert az arcokon, ezért nem a teljes arcról készült képet, hanem csak egy adott részét átadva ezeknek a kereséseknek a hibaarány közel 0-ra csökkenthető.

3.4. Kiegészítő funkciók

A felügyelő rendszer alapvető funkcióinak tárgyalása után a következőkben bemutatom a nem szükségszerű, de nagyon hasznos kiegészítő eljárások működését. Ezek közé tartozik a pehelysúlyú mozgásérzékelő, a hangalapú jelszófeldolgozó, a dinamikus arckövető algoritmus és a saját kamera kezelő osztály.

3.4.1. Mozgásérzékelés

Habár az emberi alak keresés nagyon hatékony a kis méretű képen, és a feldolgozási sebességet is így magasan lehet tartani, mégis indokolatlan egy olyan bonyolultságú algoritmust futtatni minden képre, ha azon nincs is mozgás. Amennyiben nincs mozgás, biztosak lehetünk benne, hogy ember nem sétált be a megfigyelt területre. Ezt kihasználva egy mozgásérzékelő algoritmus előszűrése lehetne bizonyos értelemben az emberi alak keresésnek, miszerint azt csak akkor érdemes futtatni, ha történt a képen jelentős változás. Nyilvánvaló, hogy ez az algoritmus sokkal egyszerűbb, kisebb processzor és memóriaigénnyel. A szakirodalom *background subtraction*, azaz háttér kivonás néven hivatkozik az olyan algoritmusokra, amelyek célja, hogy egy képen elkülönítsék a háttérhez és az objektumhoz (előtérhez) tartozó pixeleket. A legalapvetőbb megközelítés az, hogy egy adott képet összehasonlítunk egy referencia háttér képpel (háttér modellel), és azokat a pixeleket előtérnek, objektumnak válasszuk, amelyek értéke jelentősen eltér a referencia kép adott pixelének értékétől. A háttér modell kialakítására és az „eltérés” nagyságának meghatározására rengeteg módszer született. A legegyszerűbb eljárás az, amikor az adott képet egyszerűen kivonjuk egy régebbi képből, és ahol a különbség nagy, ott mozgást jelzünk.

Ennek hátránya azonban, hogy nem képes általánosításra, a háttér kicsiny mozgásai (pl. fényviszony változása, fényvisszaverődések okozta változások) mindig jelzést generálnak. Ettől kicsit jobb megközelítés, amikor egy adott hosszú csúszóablak fogja meghatározni, hogy melyik képekből nyerjük ki a háttér modell információt. Itt minden csúszóablakba tartozó kép adott pixelét összegezzük, majd elosztjuk a képek számával, vagyis egy átlagolást hajtunk végre minden pixelre. Ennek jelentése, hogy átlagosan mekkora volt a pixel intenzitás az elmúlt néhány darab képen. Majd az újonnan érkező képet ezzel az átlagolt háttér modellel hasonlítjuk össze, és ha ettől nagyon eltér egy pixel érték, akkor mozgást jelzünk. A legkiemelkedőbb módszerek ezen a téren valószínűségi háttér modelleket alakítanak ki. Az általam használt, OpenCV-ben megvalósított módszer ([7]) lényege, hogy minden pixel intenzitását egy tanító képhalmaz segítségével egy normális eloszlással adja meg. Vagyis minden pixelre meghatároz egy várható értéket és szórást, és az eloszlást normálisnak feltételezi (változó paraméterű Gauss görbékkel ír le minden pixelt). Innen ered a neve, *Gaussian Mixture Model*, vagyis Gauss görbék keverékéből álló modell [6]. Az OpenCV-ben szereplő verzió nagyon gyors és műveletigénye is alacsony. Fontos, hogy ennek az algoritmusnak alacsony legyen a eszközigénye, mert az idő nagy részében ez fog futni és így elkerülhető az okostelefon túlmelegedése. Az emberi alak felismerésnél tárgyalt kis méretű képen futtatva valós időben tudja érzékelni a mozgást. Habár ez nagyon jó eredmény, felesleges túlterhelés. Indokolatlan a mintavételezést 30Hz körülire venni, amikor az emberi mozgás szempontjából ez nagyon nagy érték. Ennek javításáért a mozgás érzékelő algoritmus csak 10-szer fut le egy másodpercben. Működés közben készült képeket a 9. ábrán tekinthetünk meg. Ennek bal oldalán a felül egy beérkező kép, jobb oldalán felül az ekkor történt változások (nem történt), balról alul egy újabb beérkező kép, jobb oldalán az aktuális változások a háttérhez képest. Látható az ajtó kicsiny nyitásának a hatása.



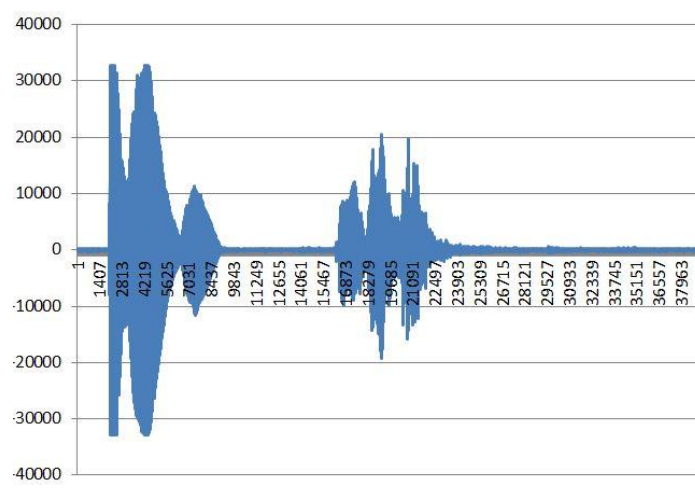
9. ábra: Mozgás érzékelés során készült képek.

3.4.2. Hangalapú jelszóellenőrzés

Az eddig tárgyalt funkciók mind azért lettek „hadba állítva”, hogy lehetővé tegyék a felügyelő rendszer számára, hogy érzékelje az emberi jelenlétet és megpróbálhassa a személy arcát felismerni. Mi van akkor, ha az arc detektálásának fizikai okokból eleve lehetetlen? Ez természetesen számos okból előfordulhat: rossz fényviszonyok, eltakart vagy a kamerával csak háttal forduló arc, gyors mozgás, amely miatt lehetetlen éles képet készíteni. Ilyen esetekben is valahogy meg kellene arról bizonyosodni, hogy a detektált emberi alak ismert archoz tartozik-e. Az nyilvánvaló, hogy maga az emberi alak sok esetben detektálva lesz, attól függetlenül, hogy az arc nem. Ezen alkalmakkor csak más interakció segítségével lehet meggyőződni arról, hogy az ismert arc tulajdonosáról van-e szó.

A probléma megoldásához a hang alapú interakciót választottam. Az azonosítás egy előre megtanult jelszó ismételt kimondásával valósul meg. A tanulási fázis elején a felhasználó egy tetszőleges jelszót mondhat bele a rendszerbe, amit az felvesz és megjegyez. Ha a későbbi éles működés során az arc felismerése sikertelen lenne valamilyen okból kifolyólag, akkor a rendszer egy hangjelzés segítségével „megkéri” a felügyelt területen tartózkodó személyt, hogy ismételje el a megtanult jelszót. Ezt is felvételezi, majd egy alább ismertetett algoritmussal kinyeri a beszéd részét a felvételi időnek, majd a referencia és az aktuális döntéshez tartozó felvételeknek a korrelációja alapján döntést hoz, hogy ugyanaz a szó hangzott-e el. Ha igen, akkor az azonosítás sikeres, egyébként nem, és a rendszert értesít egy előre megadott telefonszámot vagy e-mail címet MMS ill. e-mail formájában.

Ahhoz, hogy az azonosítás megvalósulhasson, szükség van egy algoritmusra, amely a felvett jelből kinyeri a beszédet. Ez nem egyszerű feladat. A felvett jel nagyon zajos. Egy felvett jelre láthatunk példát a 10. ábrán, ahol a „TDK” szó hangzott el.

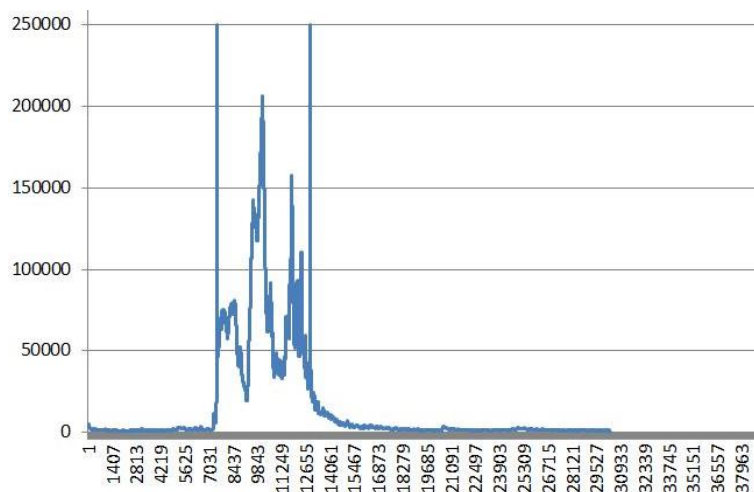


10. ábra: Felvett hang jelalakja.

Hangról lévén szó, a jelalak egy periódikus függvény, amely pozitív és negatív tartományban is vesz fel értéket, modulálva a hang erősségével. Ilyen jelek feldolgozása nehezebb, mint az egyenirányítottaké. Ezért a szűrés első lépésében a jel abszolút értékét vesszük. Ekkor a jel már csak a pozitív tartományban lesz jelen és eltűnik szinuszoid jellege. Azonban még így is sok zavar, ugrás található benne, ezeket szűrni kell. Ennek megoldásához egy csúszóablakos, súlyozott, átlagoló szűrőt használtam:

$$f(x) = \frac{1}{2n+1} \sum_{i=-n}^n (n - |i| + 1) * y(x+i), x = n, \dots, L-n$$

Ahol $f(x)$ a szűrt jel az x pozícióban, $n=N/2$, ahol N a csúszóablak szélessége. L az eredeti jel hossza, $y(x)$ pedig az értéke az x pozícióban. A 10. ábrán látható jelalak elején megfigyelhető kiemelkedések a rendszer által kiadott hang miatt jelennek meg, ami mindig ugyanolyan hosszú, tehát eldobható a jel eleje a kiadott hang végéig. A 10. ábrán látható jel szűrt, kezdeti részt nem tartalmazó alakja a 11. ábrán figyelhető meg.



11. ábra: Szűrt jelalak.

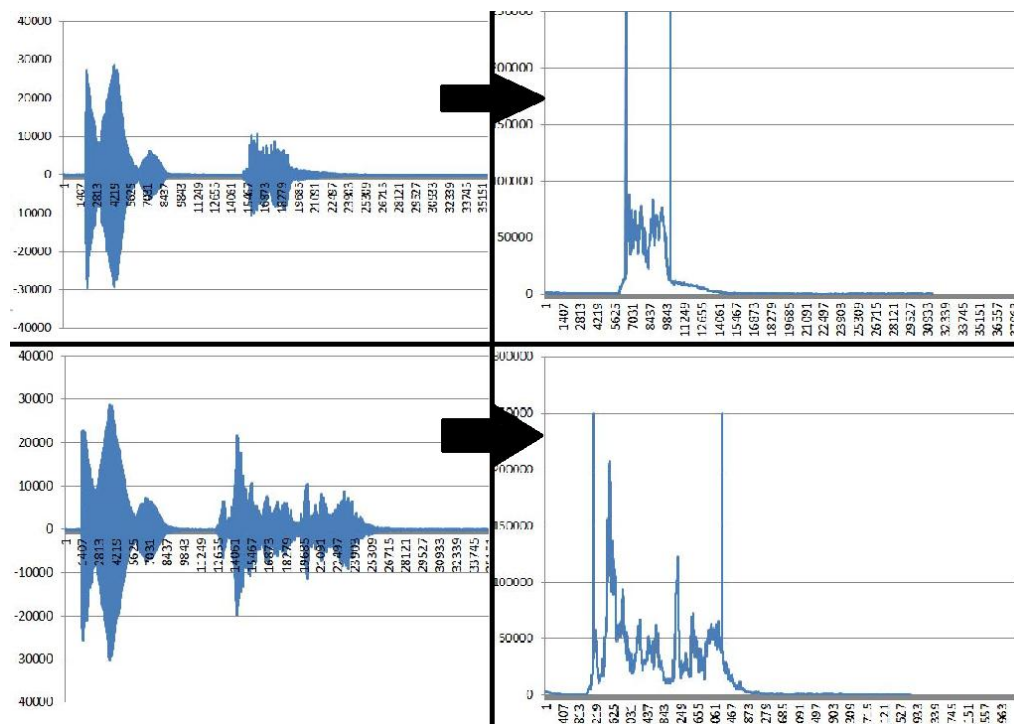
Az ábrán látszik, hogy a szűrő szépen kisimítja a jelet, az ugrásokat viszont megőrzi, így a pontosság nem, de a lényegtelen információ elveszik. Innen már lehetséges a beszéd kiszűrése. Látható, hogy a felvételi idő alatt itt most csak egy régió emelkedik a zajszint fölé. Azonban sokszor előfordul, hogy valamilyen nagy energiájú, túszerű zaj kerül a rendszerbe. Ezeket a következő módon szűröm ki:

1. Végighaladva a jelen, megkeresem a maximumot.
2. A maximum érték alapján egy küszöbszintet számolok: maximális érték/5 .
3. Újra végigmegyek a jelen és egy kétállapotú változó segítségével folyamatosan elmentem, hogy jelenleg a küszöbszint feletti, értelmes beszédet tartalmazó zónában vagyok vagy sem. Mindezt úgy, hogy a küszöb átlépésekor a változót igazgá teszem, a

pozíciót feljegyzem, mint régió kezdet, majd egy számlálót egy adott kezdőértékre állítok. Ha egyszer a jel a küszöbszint alá esik, akkor ennek a számlálónak az értékét csökkentem, és megjegyzem ezt a helyet, mint lehetséges zóna vége. Ha továbbra is küszöbszint alatt haladok, akkor minden lépésben csökkentem a számlálóm értékét. Ha a számláló értéke eléri a 0-t, akkor tudom, hogy az imént feljegyzett utolsó küszöbszint utáni érték volt egy lehetséges beszéd régió vége. Ha a számláló értéke nem jut el 0-ig, és újra küszöbszint fölé kerülök, akkor a számláló értékét visszaállítom a kezdőértékére, és a feljegyzett lehetséges régió véget elfelejtem. Ez azért fontos, mert beszéd közben is lecsökkenhet a jelerősség a küszöb alá, ami ilyesfajta hiszterézis nélkül egyből a régió végét jelentené. Ezzel a módszerrel csak az számít régió végének, ami után sokáig „csönd” van.

4. Miután kinyertem a lehetséges régió kezdő- és végpontokat, megkeresem a leghosszabb régiót, ezt választom a beszéd régiójának. A 11. ábrán a két kiemelkedő, jelen esetben 250 000-es érték jelzi, hogy az algoritmus hová tette a beszéd kezdetét és végét.

További példákat figyelhetünk meg a 12. ábrán.



12. ábra: Példák hang feldolgozására és a beszéd régió megtalálására

Ha már rendelkezésre állnak a beszéd régiók, akkor ezeket kinyerve a referencia és az aktuálisan felvett hangból, kezdődhet az összehasonlítás. Ennek során az összehasonlítandó beszédre részt ugyanolyan hosszúra alakítom, mint a referencia jelszó hossza (nyújtás, majd

interpoláció). Ez azért fontos, mert ugyanazt a szót többféle ritmusban is ki lehet mondani. Ha már egyforma hosszúak, keresztkorrelációval kiszámítom az egyezést a két hangalapú jelszó közt. Ha egy előre megadott küszöböt meghalad a hasonlóság mértéke, akkor a bediktált jelszót elfogadom, egyébként pedig elvetem.

Ez a módszer tehát lehetővé teszi, hogy arc nélkül is lehetséges legyen egy személy beazonosítása. Meg kell említeni, hogy ez pontosan visszavezetett ahhoz a problémához, mint amivel a közönséges riasztóberendezések rendelkeznek, révén egy adott jelszótól függ, amelyről bárki tudomást szerezhet.

3.4.3. Dinamikus arckövetés

Már az előzőekben említés esett arról, hogy fontos és elengedhetetlen az arcról nagyon gyorsan, a lehető legnagyobb felbontással képeket készíteni. Ehhez szükségünk van egy olyan eljárásra, amely biztosítja, hogy ne a teljes képet kelljen átkutatni arc után. Ehhez két feltevéssel élhetünk:

1. Emberi alak fellelése után elegendő csak az emberi alak helyének felső részén keresni arc után egy adott méretű régióban.
2. Ha már találtunk arcot, valószínű, hogy a következő képen is ehhez a helyhez nagyon közel lesz az arc.

Ezeket kihasználva egy olyan módszert fejlesztettem ki, amely „nyomon tudja követni” az arc helyzetét. Az eljárás egy növekvő méretű keresési ablakot használ. Az ablak mérete azzal arányosan nő, hogy hány képkocka óta nem volt arc az adott ablakban. Ha ismét talál egy arcot, akkor az ablak középpontját áthelyezi az arc középpontjába és innen folytatja újra a növekedést, ha ismét eltűnne az arc. A 13. ábrán egy olyan sorozatot láthatunk, amelynek első képén az ablak az emberi alak detektálás után állítódott be. Majd a további képeken a mozgás hatására változó helyzetű és méretű ablak figyelhető meg.



13. ábra: Dinamikus arckövető algoritmus működése közben készült képek.

3.4.4. Saját kamera kezelő osztály

Az alkalmazás szempontjából kritikussá vált, hogy elérhetőek legyenek a következő, kamerát érintő funkciók:

1. Változtatni lehessen a kamerából érkező kép felbontását.
2. Folyamatosan bekapcsolva lehessen tartani a felvétel során a kamerához tartozó LED vakut (éjjeli üzemmód).

Ezen funkciókat csak saját kamera kezelő osztály megírásával lehet elérni. Az OpenCV4Android szerencsére segítséget nyújt ehhez. Több példaprogramot is biztosít, amelyek a kamera kezelését mutatják be. Ezek segítségével sikerült nekem is megírni egy saját osztályt, amely így alkalmassá vált arra, hogy a hardver szintről változtassa az érkező kép felbontását és ki- ill. bekapcsolva tartsa a kamerához tartozó vaku LED égőt.

A felbontás váltása hardver szinten gyorsan, kevesebb, mint egy másodperc alatt végbemegy, ami lehetővé tette, hogy az emberi alakot még kis-, de az arcot már nagy felbontású képen kereshessem. Az emberi alak detektálása után kevés idővel már a dinamikus arckeresés indulhat el, az elérhető maximális felbontású képen.

4. A rendszer működési fázisainak vizsgálata

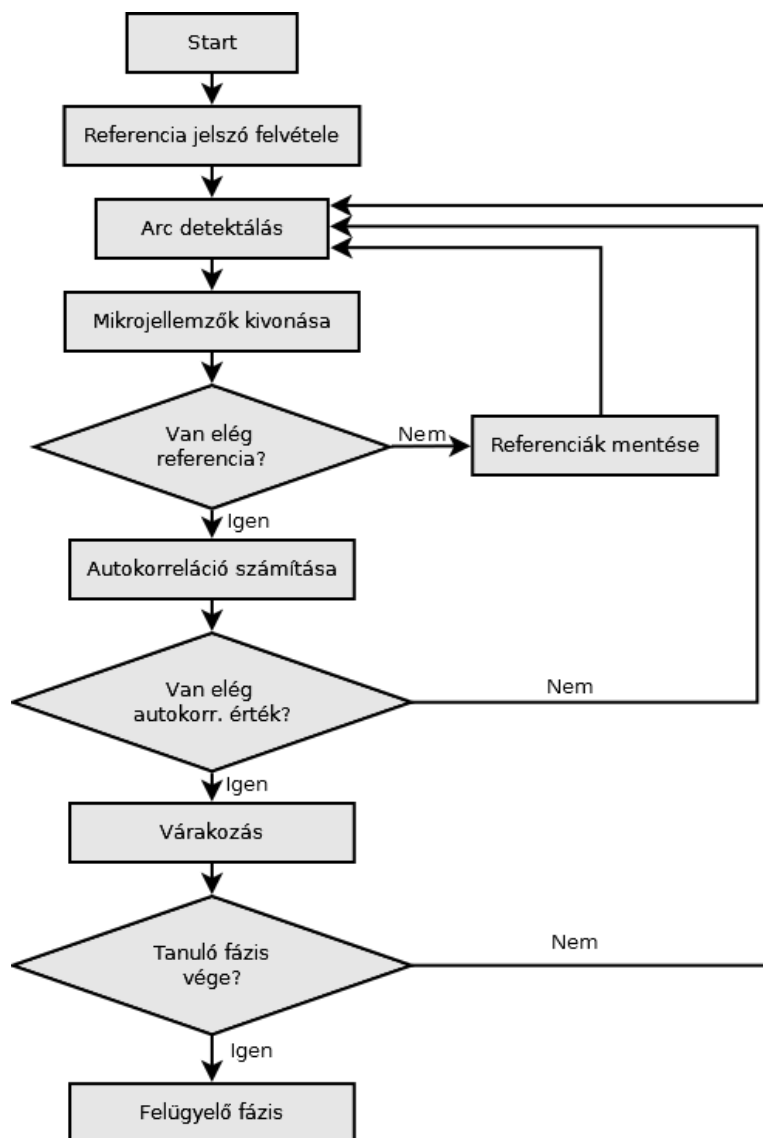
4.1. Tanuló fázis

A rendszer feladata, hogy megjelenő arcokat „ismert” vagy „ismeretlen” címkékkel lásson el. Ehhez azonban először egy felügyelt tanulási fázison kell átesnie, amely során elsajátíthatja a később felismerendő arcot. Ezt az alkalmazás első indításakor teszi meg.

Maga a tanulás a később felismerni kívánt jellemzők kivonását, elmentését jelenti. Jelen megvalósításban ezek a leírók az arcról kivont mikrojellemzők: száj, orr, bal szem, jobb szem, szempár. Ez az öt kép kerül elmentésre egy referencia arcról, minden arcrész egy előre megadott egységes méretre skálázva. A rendszer ezen mikroképeket egy napig gyűjti, fontos, hogy ez az idő alatt csak a később felismerni kívánt felhasználó jelenjen meg a felügyelt területen.

A képkészítések során homályos, elmosódott képek sokszor készülnek. Ez természetesen nagyon rossz hatással van az egész felismerő rendszerre. Ennek kiszűrése érdekében egy referencia elmentése helyett több kerül elmentésre minden arcrészről, amely szám paraméterezhető, jelenleg 5-re állított. Továbbá a korrelációt zavarja a fény intenzitásának és irányának változása, ezért ezt kiküszöbölendő, a rendszer egy nap során,

több alkalommal is készít referenciákat, időben elszeparálva, minden kis lokális időben pedig többet is, hogy a rossz képeket később szűrni lehessen. Tehát az első nap végére, feltételezve, hogy a felhasználó 10-szer került a kamera elé, lesz $10 \cdot 5$ darab referenciánk minden arcrészről, ahol az ötös csoportok időben jól elválasztottak, mivel a rendszer óránként készít referenciákat. Ha egy lokális referencia gyűjtési időben már megvan a kellő számú minta minden mikrojellemzőről, akkor a rendszer tovább folytatja a képek gyűjtését, és az itt kivont jellemzőket azonnal össze is hasonlítja az éppen rögzített referencia jellemzőkkel. Így bizonyos értelemben autokorrelációról van szó, hiszen egy személy adott arcrészét sajátjával hasonlítjuk össze. Ezen autokorrelációs értékekből sok készül el, a jelenlegi beállítás alapján ez 50 darab. Ezekből átlagot számolva egy dinamikusan meghatározott küszöbszintet állítunk elő a későbbi felismerés számára, amely időben hozzá van rendelve a referenciákhoz.



14. ábra: A tanuló fázis folyamatábrája.

Ha elkészültek a dinamikus küszöbszintek minden arcrészhez, akkor a rendszer alvó állapotba tér, várakozik, majd egy óra múlva ismét referencia gyűjtési módba lép. Ez azért fontos, hogy a referenciák jól elkülönüljenek időben. A tanulási fázis végére tehát a következőkkel rendelkezünk:

- adott számú referenciacsoporttal, ahol az egyes csoportok rögzítési ideje között jelentős idő telik el
- referencia csoportonként dinamikusan meghatározott küszöbszintekkel minden arcrészre
- referencia csoportonként adott számú mintával minden arcrészről (pl. 5)

A tanulás menetét egy folyamatábrán összefoglalva a 14. ábrán, egy referenciacsoport képeit és a hozzájuk tartozó küszöbszinteket pedig a 15. ábrán figyelhetjük meg.



15. ábra: Egy referenciacsoport képei (bal) és a hozzájuk tartozó küszöbszintek (jobb)

4.2. Felügyelő fázis

Amikor a rendszer befejezte a tanulást, az éles, riasztó üzembe lép át. Ez a fázis az alkalmazás leállításáig tart és folyamatos működésű. Ekkor történik a felügyelő rendszeri feladatok ellátása.

Ebben a módban, a rendszer folyamatosan, bizonyos időnként referencia betöltést hajt végre. Ez a dinamikus referencia betöltés. Ennek során az elmentett referenciákat végignézve dönt arról, hogy a betöltés idején (valószínűsíthetőleg) melyek azok, amelyek a legjobban illenek az akkori fényviszonyokhoz (intenzitás, irány). Ezt oly módon teszi, hogy a betöltés órában kifejezett idejéhez képest kiválasztja azokat a referencia képeket, amelyek elmentési ideje órában kifejezve a legközelebb áll az adott értékhez. Tehát pl.: mentettünk el referenciákat 9,10,12,13,15,18 órakor a tanulás napján, a dinamikus betöltést a rendszer 11:32-kor kezdi el, akkor végignézve a lehetséges opciókat, a 10 és 12 órakor rögzítettek a legmegfelelőbbek. Ebből a későbbit, tehát a 12 órakor rögzítetteket fogja betölteni. A

következő betöltésig így biztosan ez az 5*5 referencia kép és a hozzájuk tartozó 5 küszöbszint lesz a felügyelő rendszer referencia bázisa.

A felügyelés során folyamatosan mozgásérzékelés történik, itt nincs alvás vagy várakozás. Ha mozgás jelenik meg a képen, akkor a rendszer egyből emberi alak detektálásra vált át. Ha nem talál embert, akkor bizonyos idő után visszaáll mozgásérzékelésre. Ha talál, akkor a dinamikus arckövető eljárással arc után kezd kutatni. Ha nincs arc, akkor jelszót kér bemondásra, hiszen arc nélkül a felismerő rendszer nem működhet.

Ha talál arcot, akkor rögzíti, és újra keres, hogy ne teljen el sok idő az arcok egyenkénti feldolgozásával, hanem minél több arcot lehessen rövid idő alatt felvenni. Ha sikerült egy előre megadott maximális számú arcról képet elmenteni, akkor befejezi a keresést, illetve ha nem sikerült, de valahány arc azért rendelkezésre áll, akkor elkezdődik az arcok felismerésének folyamata. Először a mikrojellemzők kinyerése kezdődik minden arcról készült képről, majd ezek korrelációja a referencia jellemzőkkel számítható ki. Ebből lesz sok korrelációs értékünk, arcokra és arcrészekre bontva. Innen elindul a döntéshozatal, a korrelációs értékek felhasználásával, amelynek alapja, hogy egy arc egy mikrojellemzőjére egy lokális „igen” a döntés, ha a korreláció meghaladja az éppen a rendszerben szereplő, az ezen arcrészre vonatkoztatott küszöbszintet, ha kisebb tőle, akkor „nem”, ha ki sem volt nyerhető a jellemző, akkor „ismeretlen”. A döntéshozás a következő feltevések alapján hoz egy végső ítéletet:

- ha sok jellemző nem volt kinyerhető egy képről, akkor arra az arcra vonatkozóan a döntés „ismeretlen”
- ha sok olyan képünk van, amikre a döntés „ismeretlen”, akkor a végső döntés is „ismeretlen”
- ha azonban van olyan kép, amelyről az összes jellemző kinyerhető volt, akkor:
 - ha 80%-ot meghaladja a lokális „igenek” száma, akkor erre az arcra a döntés „extra igen”
 - ha 80%-ot meghaladja a lokális „nemek” száma, akkor erre az arcra a döntés „extra nem”
- ha van minimum 2 „extra igen” vagy „extra nem” szavazat, akkor a végső döntés „igen” illetve „nem”
- ha van egy „extra igen”, akkor, ha a maradék képekre a döntések több, mint fele „igen”, akkor a végső ítélet „igen”, ha „nem”, akkor pedig a végső „ismeretlen”, ugyanez, csak fordítva az „extra nemre” is

- ha nincs extra jó vagy rossz, akkor, ha az arconkénti döntések (arconként, ha a lokális döntések több, mint fele „igen”, akkor „igen”, egyébként „nem”) 80%-a „igen”, akkor a végső döntés „igen”, ugyanez „nemekre” is
- egyébként a végső döntés „ismeretlen”

Ezt formalizálva a következők érvényesek:

Az OpenCV a következő képlet alapján számolja a normalizált keresztkorrelációt [8]:

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') * I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 + \sum_{x', y'} I(x + x', y + y')^2}}$$

ahol T a referencia kép, I pedig az aktuális, összehasonlítandó kép. Esetünkben egy korrelációs értéket kapunk az összehasonlítás eredményeként, mivel a képeink megegyező méretűek. Innen egy lokális döntés:

$$\text{lokális döntés} \begin{cases} \text{igen} & \text{ha } R(T, I) \geq \text{küszöbszint} \\ \text{nem} & \text{ha } R(T, I) < \text{küszöbszint} \\ \text{ismeretlen} & \text{ha nem sikerült kinyerni a jellemzőt} \end{cases}$$

ahol $R(T, I)$ a referencia és az összehasonlítandó kép keresztkorrelációjának értéke, a *küszöbszint* pedig a dinamikusan, autokorrelációkból származtatott érték. Innen egy arcra vonatkoztatott döntés:

$$\text{arc szintű döntés} \begin{cases} \text{extra igen} & \text{ha } i \geq e * n \\ \text{igen} & \text{ha } i > t * n \\ \text{nem} & \text{ha } i < t * n \\ \text{extra nem} & \text{ha } i < (1 - e) * n \\ \text{ismeretlen} & \text{ha } u \geq (1 - e) * n \end{cases}$$

ahol: i azon lokális döntések száma, amelyek eredménye „igen”, e egy magas érték a $[0,1]$ -ből, t alacsony érték $[0,1]$ -ből, $n = r_f * n_f$, n_f a jellemzők száma, r_f a jellemzőnként rögzített képek száma, u az „ismeretlen” lokális döntések száma. Jelenlegi megvalósításban: $n_f=5$, $r_f=5$, $e=0,8$, $t=0,5$. Innen a végleges döntés:

$$\text{végleges döntés} \left\{ \begin{array}{ll} \text{igen} & \text{ha } e_i \geq w * f \\ \text{nem} & \text{ha } e_n \geq w * f \\ \text{ismeretlen} & \text{ha } e_i = 1 \text{ és } n_i < t * (f - 1) \\ \text{igen} & \text{ha } e_i = 1 \text{ és } n_i \geq t * (f - 1) \\ \text{ismeretlen} & \text{ha } e_n = 1 \text{ és } n_n < t * (f - 1) \\ \text{nem} & \text{ha } e_n = 1 \text{ és } n_n \geq t * (f - 1) \\ \text{igen} & \text{ha } n_i \geq h * f \\ \text{nem} & \text{ha } n_n > (1 - h) * f \\ \text{ismeretlen} & \text{egyébként} \end{array} \right.$$

ahol: f az arci képek (döntések) száma, e_i az „extra igen”, e_n az „extra nem”, n_i az „igen”, n_n a „nem” arconkénti döntések száma, w egy $[0,1]$ -ből származó kis értékű szám, t ugyanaz, mint korábban, h egy $[0,1]$ -ből származó szám. Jelenlegi megvalósításban $h=t$.

Azért van szükség ilyen bonyolult döntéshozóra, mert sok értékünk van és ki is akarjuk szűrni a hibás képekből származó rossz értékeket, de annyira se szabad szűrni, hogy abba a ténylegesen adatot hordozó mérések is bele tartozzanak. Így preferálva van az, ha egy arcról jó képet sikerült készíteni és a lokális döntések egyoldalúak (az „extra igen” és „extra nem”). Ugyanakkor, ha ez csak egyszer sikerül, akkor lehet véletlen is, és a többi szavazat is számít. Ha semmi kiemelkedő sincs, és ha az egyes szavazatok a többségi döntés alapján mégis gyenge „igenekeket” és „nemeket” generálnak arconként, akkor a statisztikai alapokat felhasználva mondható, hogy az arc „igen” vagy „nem”, de csak ha elég sok gyenge „igen” vagy „nem” van. Ha ezek egyike sem teljesül, akkor a döntés bizonytalan és a végső kimenet „ismeretlen”.

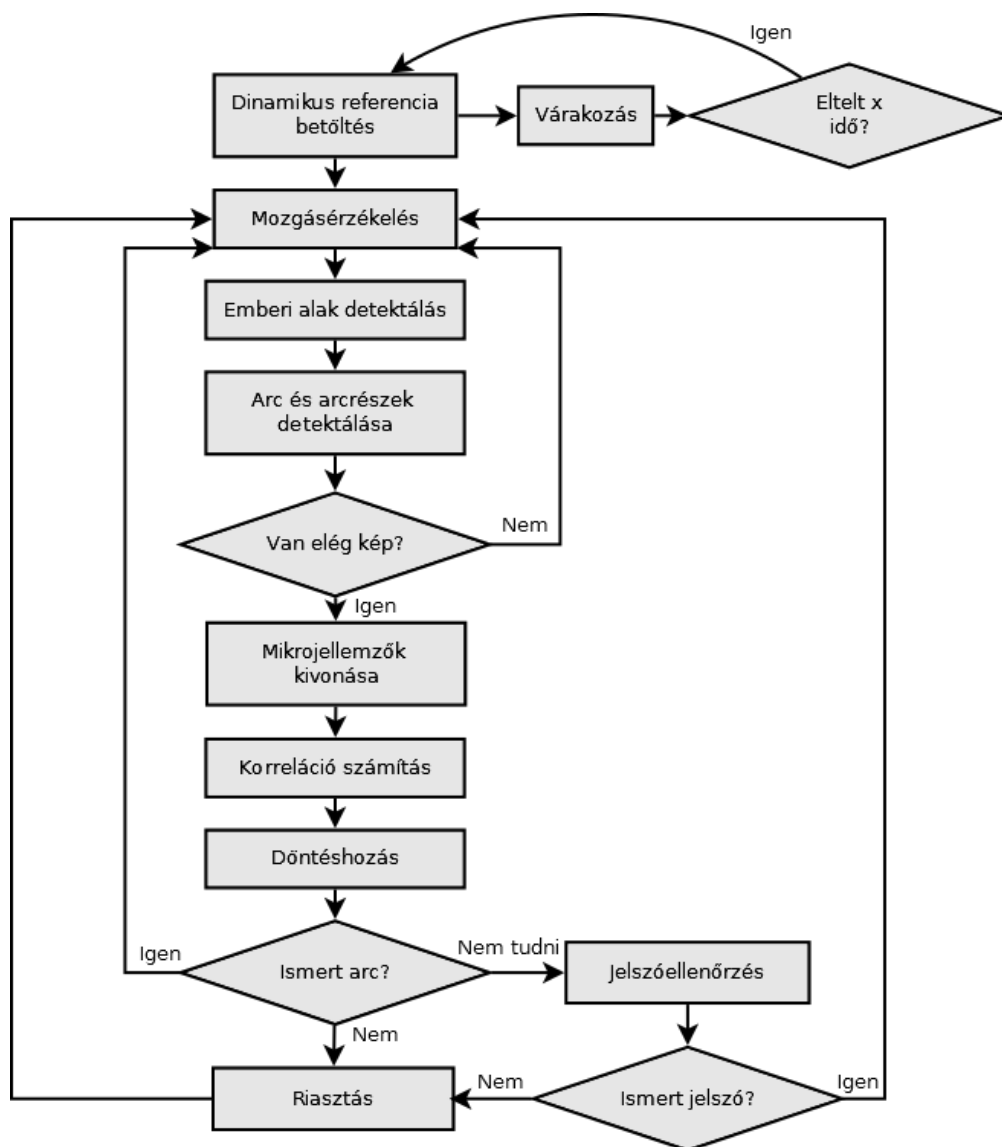
A döntéshozó „nem” kimenete esetén a rendszer egy e-mail ill. MMS üzenet küldését vonja maga után, amely tartalmaz egy képet az ismeretlen arcról. Ezt a funkciót csak teszteltem, ténylegesen nem használtam, ehelyett elmentettem egy képet, amelyre ráírtam adatokat a döntéshozásról és a végső döntésről. A riasztás után ismét mozgásérzékeléssel folytatódik a felügyelet.

A döntéshozó „ismeretlen” kimenete esetén, a rendszer hangjelzéssel kéri a megfigyelt területen tartózkodó személyt, hogy mondja be az azonosításhoz szükséges jelszót. Ennek ellenőrzésével teszi „igenné” vagy „nemmé” a bizonytalan kimenetet. Ha a jelszó nem megfelelő, a fentebb leírt riasztás indul el.

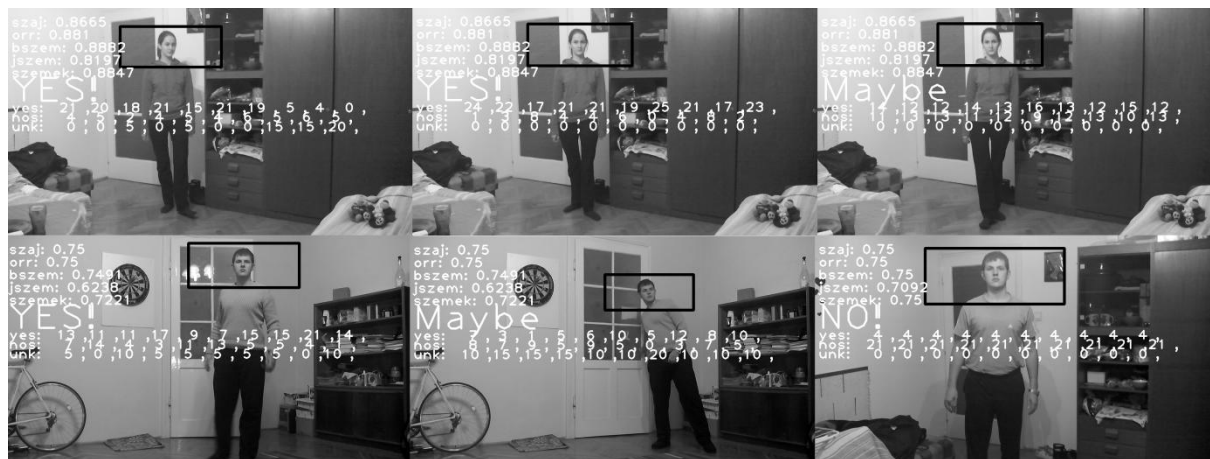
A döntéshozó „igen” kimenetére a rendszer nyugtázza a döntést és mozgásérzékeléssel folytatja munkáját.

Az eddig tárgyalt működés folyamatábrája látható a 16. ábrán. Egy pár elkészült döntést tartalmazó kép pedig a 17. ábrán tekinthető meg (egy képen: bal felső sarokban az

éppen aktuális küszöbszintek arcrészek szerint feltüntetve, alatta a végleges döntés, alatta a lokális döntések, arconként oszlopba gyűjtve).



16. ábra: A felügyelő fázis működésének folyamatábrája



17. ábra: Felügyelő rendszer kimenetéből származó képek

5. A rendszer tesztelése

Az elkészült felügyelő rendszer tesztelése több módon és fázisban zajlott. Külön-külön teszteltem az egyes komponensek megbízhatóságát, majd az együttműködésük eredményét. Némely fázist több készüléken is elvégeztem. Ezek közt szerepelt: Pulid F11, 2x1GHz elérhető maximális felbontás 1280x720, ár: 20 000Ft (kínai); LG G3, 4x2,5GHz, használt felbontás 2048x1080, ár: 150 000Ft; Samsung S Advance, 2x1GHz, elérhető maximális felbontás 800x600, ár: 40 000Ft,

Először a mozgásérzékelés, emberi alak és arc detektálást vizsgáltam meg. Ezt mindhárom eszközön sikerült tesztelni. Sajnos az alkalmazás ezen tesztek alatt már nem támogatta a másodpercenkénti képszám, fps (*frame per second*), kiírását, így csak a Pulid F11-en mért eredményekhez tudtam viszonyítani. Mindhárom eszközön a mozgásérzékelés és az emberi detektálás 176x144-es felbontású képen futott, amelyet mindhárom akadálytalanul fel is tudott dolgozni, tehát biztosan tudták a >10fps sebességet. Érdeemes megjegyezni, hogy a Pulid F11 volt a leglassabb minden tekintetben, így ezek az eredmények az alsó határát képezik a többi telefon teljesítményének. Az arc detektálás teljesítményét már egy kicsit bonyolultabb mérni. Mindhárom eszköz más felbontású képen végezte az arc keresését, és a keresési ablak mérete is folyamatosan változik. Pulid F11-en 10 egymás utáni arc képének rögzítése nagyságrendileg 1 másodpercig tart. Ehhez képest a másik két eszköz jobban teljesít. Kiemelkedő az LG G3 teljesítménye, amely egy szempillantás alatt képes a 2048x1080-as felbontású képről 10 arcot lementeni. Talán ez már hátrány is, mivel a cél az lenne, hogy az arcról kicsit mindig más szemszögből készüljön a kép, de ez az eszköz általában teljesen egyforma képeket készít a sebessége miatt. Képélességben is messze jobban teljesít a többiektől, lévén lézeres autofókusszal rendelkezik. A megbízhatóság tekintetében nem észlelhető különbség, minden eszközön minden észlelés megtörtént, a további eredmények az alkalmazott módszerektől függenek.

Magának az arcfelismerő algoritmusnak a tesztelése volt a másik fő feladat. Az ütemezett referencia készítésnek és betöltésnek a tesztelése sikeres volt. A személyek felismerését több módon is teszteltem. Az első tesztelési forma statikus állapotban, a referencia készítésének helyén történő mérés volt. A statikus állapot a személyre igaz, vagyis nem mozgott a képek készítése során, és a referencia készítés helyén és időpontjában történt a tesztelés. Ez a legjobb eshetőség, hiszen pontosan azokat az anomáliákat szünteti meg, amelyek a korrelációt megzavarják. A tesztek során a felhasználó szerepelt a képeken, tehát a

válasz mindig „igen” kellett volna, hogy legyen. Az eredmények az 1. táblázatban figyelhetők meg.

Eszköz	Mérések száma	Igen	Nem	Talán (megerősítve hang alapon)
Pulid F11	30	18	4	8 (6)
Samsung S Advance	20	10	3	7 (6)
LG G3	15	8	4	3 (0)

1. táblázat: Statikus, megtanul személy felismerése

A másik eset, amikor nem az ismert személy, hanem idegen szerepel a képeken, a jelszót sem ismeri. Az eredmények a 2. táblázatban szerepelnek.

Eszköz	Mérések száma	Igen	Nem	Talán (megerősítve hang alapon)
Pulid F11	20	3	10	7 (1)
Samsung S Advance	20	5	11	4 (0)
LG G3	0	0	0	0

2. táblázat: Statikus, ismeretlen személy felismerése.

A mérésekből levont tanulságok: A fejlesztést a Pulid eszközön végeztem. Ennek az eredményei a legjobbak. Statikus esetben ezek az eredmények talán kielégítőnek nevezhetőek. A második táblázatban több hibás mérést találunk. Ez sajnos a rosszabb eset, mert itt ismeretlen arcot ismertnek tételezett fel a rendszer. Összességében a számok 80% körüli pontosságnál mozognak statikus, ismert személy esetén, és alacsonyabbak nem ismert személy esetén. Meglepően jól működik a hang alapú jelszó felismerés, amely magasabb, mint 85% pontossággal működött ezeknél a méréseknél, és csak a modul tesztelésekor is.

A nehezebb felismerési forma, amikor a személy nem vesz tudomást a készülékről és nem is néz bele, illetve mozgást végez, amely miatt a képek sem élesek. Nyilvánvaló, hogy itt nagyobb szerepet kap a hang alapú ellenőrzés. Maga a tesztelés egy megtanult mozgási sor végrehajtásával történt, hogy az eredmények összehasonlíthatóak legyenek. A 3. táblázatban szereplő mérések esetén a rendszer felhasználója, a megtanul személy felismerése volt a cél. A 4. táblázatban a mozgó, ismeretlen személy alapú mérések összegzése található.

Eszköz	Mérések száma	Igen	Nem	Talán (megerősítve hang alapon)
Pulid F11	20	5	6	9 (7)
Samsung S Advance	20	7	7	6 (6)
LG G3	10	2	4	4 (3)

3. táblázat: Dinamikus, ismert személy felismerése

6. Elért eredmények, összefoglalás

A kutatásom célja egy olyan mobil alapú, intelligens, alacsony költségű felügyelő rendszer megalkothatóságának a vizsgálata és létrehozása volt, amely képes felismerni a felhasználója arcát. Legfőbb eredménynek azt tekintem, hogy sikerült egy olyan rendszert megalkotni, amely lehetővé teszi arcokról készült képek gyors gyűjtését, ezzel teret adva további, mélyebb kutatásoknak. Jelen megvalósításban szereplő felügyelő rendszer az arcról kivont mikrojellemzők (száj, orr, bal szem, jobb szem, szempár) képtérbeli összehasonlításával próbálja megítélni, hogy ismert arc szerepel-e egy adott képen. Habár sikerült így is megfelelő eredményeket elérni, nyilvánvaló, hogy ezen jellemzők nagyon érzékenyek a varianciákra. További fejlesztésként elsősorban ennek javítását tűzöm ki célul.

A megvalósított algoritmusok közül különösen jól működött a hang alapú jelszóellenőrzés, amely érzékenységet a külső zajokra még javítani kell, de így is 85% feletti pontosságot ért el a tesztelés során.

Összefoglalva a rendszer erősségeit, elmondhatom, hogy egyszerűen bővíthető, újabb jellemzők vizsgálata, bevétele a rendszerbe könnyedén megoldható. A legtöbb eljárás paraméterezhető, tehát további kutatás folyamán is segítséget nyújt a szabad kísérletezésre. Az alkalmazott detektáló eljárások nagyon hatékonyan működnek, ezzel lehetővé téve az arcok gyors rögzítését. A saját kamera kezelő osztály további funkciók kifejlesztését teszi lehetővé, mint például az éjjeli üzemmód, a LED vaku ugyanis szabadon kapcsolható programból. Az alkalmazás Android alapokon íródott, így szinte tetszőleges eszközre változtatás nélkül feltelepíthető és azonnal használható. Az arcfelismerő funkció jelenleg statikus helyzetben lévő személyek esetén megfelelően biztonságosan, még mozgó egyének esetén, érthető korlátok miatt gyengébben szerepel. Azonban mégis úgy gondolom, hogy lehetséges egy olyan rendszer megalkotása, amely közel minden esetben képes egy adott arcra megfelelő döntést hozni. További bővítési és fejlesztési lehetőségek közé sorolnám a pontosított arcfelismerést, a több megtanult arcra működő azonosítást, a gépi tanulással támogatott döntéshozást és a fejlettebb interakciós formákat a felhasználóval. A rendszer tömegesebb alkalmazása esetén segítene felhasználni az egyébként kihasználatlan okostelefonokat és egyben segítene megóvni otthonainkat, értékeinket és támogatná a bűnüldözést.

Irodalomjegyzék

- [1] www.opencv.org
- [2] N. Dalal and B. Triggs "Histogram of oriented gradients for human detection", Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog., pp.886 -893 Jun. 2005
- [3] http://docs.opencv.org/modules/gpu/doc/object_detection.html
- [4] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2001.
- [5] http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html
- [6] Z. Zivkovic "Improved adaptive Gaussian mixture model for background subtraction", Proc. Int. Conf. Pattern Recognit., pp.28 -31 2004
- [7] <http://docs.opencv.org/java/org/opencv/video/BackgroundSubtractorMOG2.html>
- [8] http://docs.opencv.org/modules/contrib/doc/facerec/facerec_api.html