

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Irányítástechnika és Informatika Tanszék

Személyazonosítás és eseményfelismerés LiDAR szenzorral

TDK DOLGOZAT

Készítette
Gálai Bence

Konzulens
Dr. Benedek Csaba

2015. október 26.

Tartalomjegyzék

Kivonat	3
Abstract	4
Bevezetés	5
1. Járás- és cselekvésfelismerés korábbi eljárásai	7
1.1. Járásfelismerés	7
1.1.1. Modell alapú módszerek	7
1.1.2. Megjelenés alapú módszerek	8
1.2. Cselekvés felismerés	9
1.2.1. Csontváz alapú módszerek	10
1.2.2. Egyéb módszerek	10
2. LiDAR	12
2.1. A LiDAR szenzor	12
2.1.1. Előnyei és hátrányai	13
2.2. LiDAR keretrendszer	14
3. Gépi tanuló eljárások	17
3.1. Mély tanulás különböző módszerei	17
3.1.1. Convolutional Neural Networks	17
3.1.2. Stacked Denoising Autoencoders	18
3.1.3. Egyéb módszerek	19
3.2. A mély tanulás eddigi eredményei, alkalmazásai	19
3.2.1. Arcfelismerés	19
3.2.2. MNIST adatbázison elért eredmények	19
3.2.3. Egyéb alkalmazások	19
4. Különböző járásjellemzők vizsgálata	20
4.1. Sziluettek kinyerése	20
4.2. A sziluett lenyomat	21
4.2.1. Dynamic Time Warping	21
4.2.2. Tesztek	22
4.3. Gait Energy Image	24

4.4.	Depth Gradient Histogram Energy Image	24
4.5.	Color Gait Curvature Image	25
4.6.	Gait Energy Image neurális hálókcal	26
4.6.1.	Konvolúciós neurális háló alapú megoldás	26
4.6.2.	Többrétegű perceptron alapú megoldás	28
4.6.3.	Paraméter optimalizálás	28
4.6.4.	CNN és MLP együtt	30
5.	Eredménykiértékelés, további módosítások	31
5.1.	Módszerek összehasonlítása	31
5.1.1.	Módszerek időigényessége	32
5.2.	Eredmények javítása szűréssel, forgatással	32
5.3.	LGEI részletes eredményei	34
5.3.1.	Eredménykiértékelés	34
5.3.2.	Egyes sorozatok eredményei	34
5.3.3.	Összesített eredmények	35
6.	Események detektálása	38
6.1.	Kiválasztott jellemzők	38
6.2.	Tesztek, eredmények	39
6.2.1.	Eredmények	39
7.	Értékelés	41
7.1.	Személyazonosítás	41
7.2.	Cselekvés felismerés	41
	Köszönetnyilvánítás	42
	Ábrák jegyzéke	44
	Táblák jegyzéke	45
	Irodalomjegyzék	49
	Függelék	50
F.1.	Tiny CNN könyvtár	50

Kivonat

A videófelügyeleti rendszerek egyik problémája hagyományos optikai kamerákat alkalmazva a 3D információk kinyerése. Több kamera alkalmazása esetén nehézkes az eszközök együttes kalibrációja, amely kisebb problémák esetén újrakalibrálást igényel. Egy többsugaras forgó LiDAR eszköz 3D információt szolgáltat, kitelepítése egyszerű, így ennek használata egy jó ötlet lehet a videó megfigyelésben. Az emberek azonosítására többféle módszert is feltaláltak már, említésre méltóak az írisz, arc, ujjlenyomat olvasó rendszerek, amelyek problémái, hogy a személyek közreműködésére is szükség van, vagy a felvételeket csak közelről lehet jó minőségben létrehozni. A járás, mint biometrikus jellemző évtizedek óta kutatott téma. Ezen jellemző távolról is rögzíthető, így nincs szükség a személyek hozzájárulására, így e nélkül is elvégezhető az identifikáció. A személyek azonosításán felül még egyes cselekvések felismerése is fontos lehet egy 4D videó megfigyelő rendszer esetén. Különböző szakirodalomban fellelhető jellemzők teljesítményének elemzésének következtében mindkét feladat megoldása a szakirodalomban gyakran alkalmazott és továbbfejlesztett Gait Energy Image módszeren alapul, amely egy zajra és mérési pontatlanságokra robusztus jellemző, a felismerést pedig mély tanuláson alapuló neurális hálózatok végzik. Ezen két probléma megoldásával foglalkozik a dolgozat egy Velodyne HDL-64E többsugaras forgó LiDAR eszköz használatával.

Abstract

With traditional optical video cameras one key problem with the video surveillance systems is the 3D data acquisition. Using multiple cameras the calibration of the devices may be difficult with the need of recalibration on minor incidents. The rotating multi-beam LiDAR device supplies 3D information with easy deployment capabilities, thus using it in video surveillance seems to be a good approach. For people identification many kinds of methods have been invented, for example iris, face or fingerprint recognition, where the need of contribution from the given people is a problem. Also good quality measurements can be only accessed from close distance. Gait as a biometric feature have been studied in the recent decades. This feature can be measured from a distance, thus there is no need for people contribution for recognition. Apart from the identification task there is a need for action recognition in 4D video surveillance systems. After analyzing the efficiency of different features mentioned in the literature I used the idea of the often-cited and improved Gait Energy Image for the two tasks, which is a robust feature for inaccurate measurements and noises, the identification is done by deep learning neural networks. In this paper, I present a system that is focused on this two main tasks with the use of a Velodyne HDL-64E rotating multi-beam LiDAR sensor.

Bevezetés

Az automatizált video-felügyeleti alkalmazások fontos szerepet játszanak napjainkban a közbiztonság megteremtésében. Videó-felügyeleti rendszereket használnak terrorizmus elleni védekezésben, bűnözők felismerésére, vagy egyszerű behatolás ellenőrzéshez egy magánterületen. A megfigyelési folyamat során az egyik legfontosabb kérdés a személyazonosság megállapítása. Az emberek felismerése vizuális módszerekkel régóta kutatott téma számos iparági referenciával. Az utóbbi évtizedekben például hatékony eljárásokat dolgoztak ki írisz és ujjlenyomat alapú azonosításra. Ezekkel a biometrikus jegyekkel az a probléma, hogy elemzésük csak a megfigyelt személy hozzájárulásával, a mérőeszközzel való fizikai kontaktus útján történhet. Az arcfelismeréshez már nem kell a felismerni kívánt személy közreműködése, de szintén közelről lehetséges csak jó minőségű felvételeket készíteni róla a tömegben előforduló takarások, elmosódások, esetlegesen rossz szögéből történő felvételek várható jelenléte miatt. Az eddig említett jellemzőkkel szemben a járás vizsgálata hatékony alternatívát nyújt, hiszen a járás alapú felismeréshez elég távolról megfigyelni a személyeket, így valós eseményeket monitorozó kültéri megfigyelő rendszerekben is jól alkalmazható lehet. A járást, mint biometrikus jellemző felhasználhatóságát biztonságtechnikai rendszerekben már évtizedek óta vizsgálják [24]. A szakirodalomban számos hatékony vizuális jellemzőt találhatunk, amellyel nagy mintaadatbázisokon sikerült egyéneket megkülönböztetni [9, 11, 15]. A megfigyelt területet elhagyó, majd visszatérő, esetleg másik megfigyelt területre átsétált egyének újrafelismerése fontos lehet a megfigyelő rendszerekben. Érdekes esetek például a hosszú távú cselekvés elemzés, amelyben az akár kis időre megszakadó trajektória kinyerése fontos feladat, amihez az újrafelismerés nélkülözhetetlen, valamint egy helyszínre gyakran visszatérő, esetleg egy épületbe betérő majd máshol megjelenő személy felismerése. A járásminta vizsgálata ígéretes lehetőséget kínál az ilyen típusú feladatokra.

A mai videó-felügyeleti rendszerek többsége optikai kamerákat, vagy kamerahálózatot használ. A többkamerás rendszerek előnye, hogy nagy területet képesek belátni, és származtatott 3D információkat szolgáltatnak a környezetről, azonban hátrányuk, hogy a kamerákat előzetesen kalibrálni kell, és kis elmozdulások esetén is újbóli kalibrálásokat igényelhetnek. A kitelepítésük tehát bonyolult. Szintén problémát okoznak kameráknál a különböző fényviszonyok, amik a mérések minőségére és a képfelismerő algoritmusokra is közvetlen hatással vannak. Erre egy lehetséges megoldást kínálnak az aktív fényvel működő time-of-flight (ToF) kamerák és egyéb mélységszenzorok (Microsoft Kinect), de ezek fizikai korlátai miatt egyelőre szinte kizárólag viszonylag kis (néhány méter átmérőjű) területet megfigyelő beltéri alkalmazásokban használhatók fel. Nagyobb kültéri helyszínek

megfigyeléséhez lehetőséget nyújtanak a közelmúltban megjelent LiDAR (Light Detection and Ranging) lézerszkennerek. Bár ezek elterjedésének egyelőre gátat szab a magas áruk, a jövőben várhatóan a jobb minőségű eszközök is lényegesen olcsóbbá válnak, így elérhetőek lesznek. A dolgozatban ezért - a Szerző tudomása szerint újszerű módon - javaslatot adunk egy LiDAR alapú járáselemző rendszer kifejlesztésére. A felismerő algoritmus készítéséhez felhasználjuk a korábbi hagyományos kamera alapú járásdetekciós módszerek szakirodalmi eredményeit és azokat a LiDAR adatokban rejlő lehetőségekhez és az adatok főként felbontásbeli korlátaikhoz igazítjuk.

A módszerek kiértékelésénél fontos tényező a tesztalmanoz megválasztása. A korábbi szakirodalmi módszereket általában egyszerű tesztkörnyezetben készített adathalmazokon validálták: például a MoBo [8] adatbázis tesztalanyai jól megvilágított szobában egy futógépen sétáltak, míg a mozgást több nagy felbontású kamera rögzítette. Ilyen ideális mérési körülményekre nem számíthatunk a valós életben, ami kérdéseket vet fel az eredmények alkalmazhatóságával kapcsolatban valódi rendszerekben. Munkánk során ezért igyekeztünk realiztikus tesztkörnyezetet biztosítani: a tesztalanyainkat szabad mozgásuk közben vettük fel, ami közben egy belső udvaron természetes módon sétáltak. Egy jelenetben egyszerre 3-8 személy volt a helyszínen, így figyelembe kellett vennünk gyakori takarási problémákat is. Ezen LiDAR szenzorral történt méréseinkből kinyert személyenkénti pontfelhőkből elkészítettünk egy adatbázist, amely összesen 10 szekvencián 28 fő járásmintáit tartalmazza, tudomásunk szerint ilyen típusú, kültéri realiztikus környezetben készült adathalmazból nincs más nyilvánosan elérhető.

A video-felügyeleti rendszerek fontos feladata lehet még a személyazonosság meghatározásán túl bizonyos cselekvések automatikus jelzése, riasztások küldése esetleges erőszakos tevékenységek felismerésekor. Egy ilyen rendszer kulcsfontosságú szerepet játszhat a közbiztonság megteremtésében, megőrzésében. A szakirodalomban sokféle különböző módszert ajánlottak már cselekvések felismerésére, ám ezek gyakran csak egy adott specifikus feladat különböző eseményeinek azonosítását hivatottak megoldani, például egy Microsoft Kinect segítségével működő táncjáték [28]. Egyre több nem kamera alapú megoldás születik, legtöbb esetben mélységszenzorokat, pontosabban a Kinect szenzort alkalmazták, ezzel a probléma ismét felmerül, miszerint a Kinect kültéren nem használható jól, így ezen módszerek nem feltétlen felelnek meg más környezetben. Több nyilvános adatbázis is elérhető e témában, például a MoBo párja, a MoCap¹, vagy az MSR-Action3D² adatbázis.

A dolgozat 1. fejezete az irodalomban fellelhető korábbi kutatási eredményeket mutatja be a járás- és cselekvésfelismerésről, a 2. fejezet az általunk használt LiDAR szenzort ismerteti. A 3. fejezet a korszerű gépi tanulási módszereket, az erre épülő LiDAR mérési sorozatokat felhasználó különböző módszereket a 4. fejezet részletezi. Az 5. fejezet a járásfelismerés eredményeit összegzi, valamint ezen eredmények javítására szolgáló módszert mutat be sziluettek szűrésével. A 6. fejezet a cselekvésfelismeréssel foglalkozik, míg a 7. összegzi a végzett munkát.

¹<http://mocap.cs.cmu.edu/>, 2015. október 26.

²<http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/>, 2015. október 26.

1. fejezet

Járás- és cselekvésfelismerés korábbi eljárásai

A járás- és cselekvés felismerés terjedelmes szakirodalommal rendelkezik, ennek eredményeképp rengeteg különböző módszert fejlesztettek ki a feladatok elvégzésére. Ebben a fejezetben az eddig használt eljárások összefoglalása olvasható mindkét témában.

1.1. Járásfelismerés

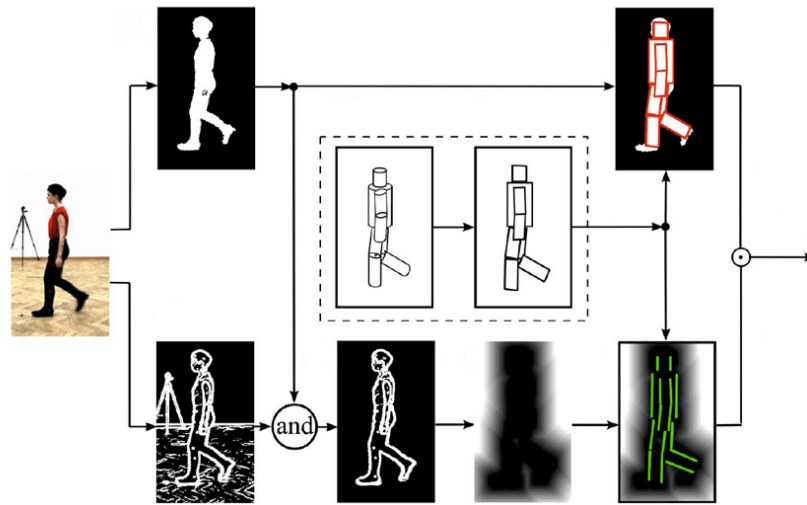
A szakirodalomban két fő megközelítést javasolnak a járás vizsgálatára, ezek a modell alapú, és a megjelenés alapú megközelítések. A főbb különbségeket az alábbi két fejezet részletezi.

1.1.1. Modell alapú módszerek

A modell alapú megoldások alapja a járás karakterisztikájának matematikai modellezése, például dinamikai, kinematikai vagy statikai megközelítésben. Egy ilyen modell paramétere lehet például a lépésköz, lábak által bezárt szög, lépés frekvenciája, vagy bonyolultabb összefüggések, differenciális egyenletek¹ paraméterei, mely egyenletek a járásra és futásra írhatók fel. Ezekben az esetekben egy gyakori ötlet a szilüettekbeli kiindulás [40], amelyet a videóképből a háttér leválasztással állíthatunk elő. Bináris szilüeteket láthatunk az 1.2. ábrán, amelyen a CMU MoBo [8] adatbázisból és a LiDAR szekvenciákból kinyert adatbázisból is 4-4 személy található. Jól láthatóan a LiDAR szilüettek részletessége, felbontása, teljessége, tehát összességében a teljes szilüett minőség rosszabb. Míg egy jó minőségű bináris szilüetből egyszerű algoritmusokkal (éldetektáció, morfológiai műveletek, Hough transzformáció) megállapítható a láb orientációja, a lépésköz hossza, stb., ezek a számítások rosszabb minőségű felvételen már nagy hibákat is előidézhetnek. Egy általános helyszínen, tömegben nehézséget jelent jó minőségű szilüettek kinyerni, így a felsorolt jellemzők ezáltal szintén nem lesznek pontosan mérhetőek. Más modell alapú megközelítésben például különböző görbéket rajzolhatunk fel a fej, a súlypont, lábak mozgására [4]. Ezen

¹<http://sprott.physics.wisc.edu/technote/walkrund.htm>, 2015. október 26.

görbék alapján szintén megkülönböztethetők a járókelők például a Dynamic Time Warping algoritmus alkalmazásával, amelyet részletesen a 4.2.1. fejezet taglal.



1.1. ábra. Modell alapú járás elemzés [21].

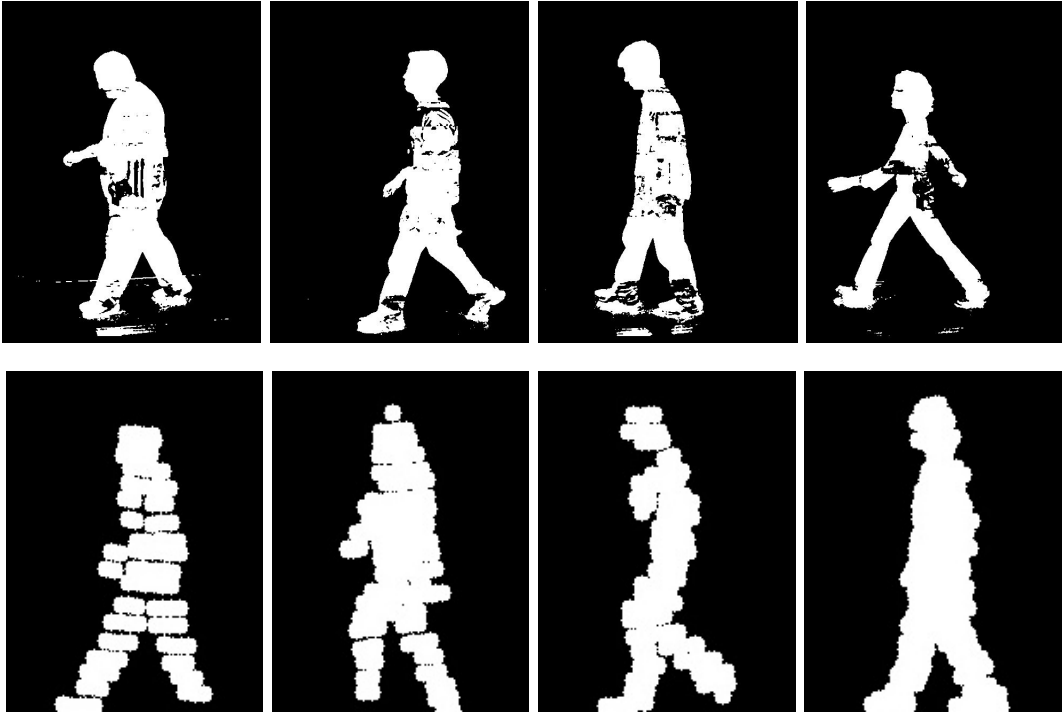
Más esetekben két, esetleg háromdimenziós modellt illesztnek az emberekre [4, 20, 21], például hengerekkel próbálják helyettesíteni az ember törzsét, fejét, lábait, karjait, vagy egyszerűen pálcika emberekkel modellezik a testet 2D-ben. Miután a modellezés megtörtént, ezen alakzatok formája, méretei, pozíciója alapján is lehet különbséget tenni az egyének között.

A modell alapú módszerek hátránya tehát, hogy megfelelő, jó minőségű adathalmaznak kell rendelkezésünkre állnia, a sziluettek megfelelő részletezettségét és hiánytalanságát illetően. Az általunk használt LiDAR szenzor bár pontos térbeli méréseket szolgáltat, viszonylag alacsony felbontású sziluettek nyerhetők ki vele. Ezen felül, a realiztikus tesztkörnyezetünkben fellépő kitakarások miatt a megfigyelhető sziluettek gyakran hiányosak. A fenti tényezők miatt munkánk során a modell alapú eljárások alkalmazását elvetettük, és a következő fejezetben részletezésre kerülő nem modell alapú technikákra fókuszáltunk. Megjegyezzük, hogy LiDAR helyett optikai kamerarendszert alkalmazva is komoly kihívásokat jelentene a megfelelő sziluettek kinyerése egy kültéri környezetben. A kívánt leírók kiszámítása után az egyedek osztályozása már többféle kiforrt osztályozó eljárásokkal történhet. Például rejtett Markov modellekkel, vagy szupport vektor gépekkel (Support Vector Machines - SVM) [20, 40].

1.1.2. Megjelenés alapú módszerek

Ez a megközelítés nem próbál meg hengereket, csontvázat, stb. illeszteni a sziluettre, se kinyerni belőle olyan jellemzőket, mint a lábak hossza, lépésköz, stb. Az ember általában ezekből próbálja megítélni a személyazonosságot egy járásminta alapján, de ez, gépi tanulásnál, nem feltétlen az optimális választás. Ezekkel a jellemzőkkel ellentétben gyakran a bináris sziluettekbelől kinyert alacsony szintű képi jellemzőket használja [9, 15] a megjelenés

alapú megközelítés, amelyeken különböző transzformációkat, dimenzió csökkentést hajtanak végre. Esetenként a sziluettet különböző részekre vágva történik a felismerés, például külön feldolgozása sziluett alsó és felső régióinak [40], majd ezek együttes eredményéből történik a becslés. Az ezekből kinyert jellemzőket végül rejtett Markov modellekkel, neurális hálókkal, vagy hasonló osztályozó rendszerekkel dolgozza fel a felismerő rendszer.



1.2. ábra. *Bináris sziluett minták (a) CMU MoBo [8] adatbázisból és a (b) LiDAR adatbázisból.*

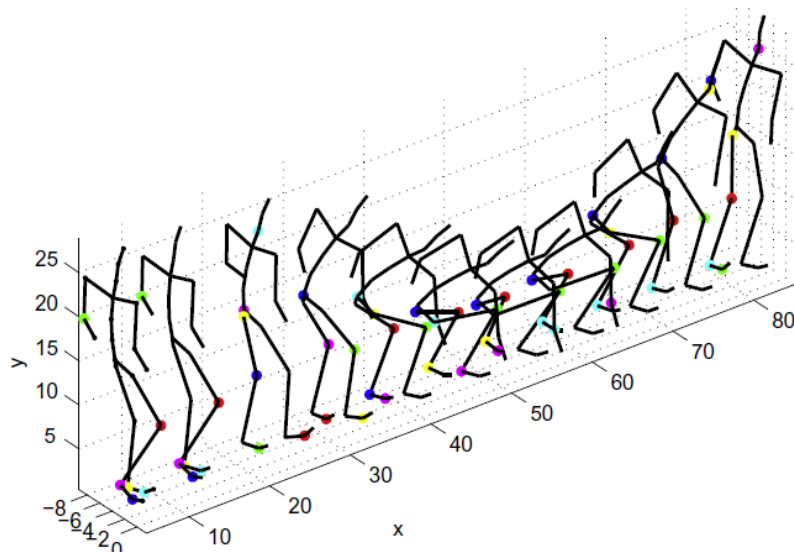
A járás felismeréséhez megfelelően hosszú mintasorozatra van szükségünk. Minél több minta áll rendelkezésünkre egy-egy járásciklusból, annál jobban tudjuk ezekből az egyes személyek járását megkülönböztetni. Az időben egymást követő sziluett képek együttes vizsgálata a robusztusságot, hatékonyságot is növelheti. Ezen bináris képek átlagolásából a zaj is valamelyest kiszűrődik, így kevésbé érzékeny, mint magában egy-egy kép. Han és Bhanu bevezetett egy képátlagolás alapú reprezentációt [9], amit "járás energiaképnek" (Gait Energy Image, GEI) nevezett el. A GEI-t első publikálása óta számos módszerben felhasználták, alapjául vették új módszereknek, például a Depth Gradient Histogram Energy Image-nek [11], amely nem csak a bináris sziluettekből, de mélységi adatokból, élekből és gradiensékből is kiindul, így jobb eredményt is ért el a felismerésben, mint magában a GEI. A dolgozat 4.6 fejezetében bemutatunk egy saját módosított GEI alapú megoldást is.

1.2. Cselekvés felismerés

A járáshoz hasonlóan a cselekvések felismerését is lehetőségünk van modell alapú és megjelenés alapú megközelítésben vizsgálni, ahol a modell alapú módszerek többsége mind csontvázakból indul ki, míg a megjelenés alapúak nyers pontfelhőkből, sziluettekből.

1.2.1. Csontváz alapú módszerek

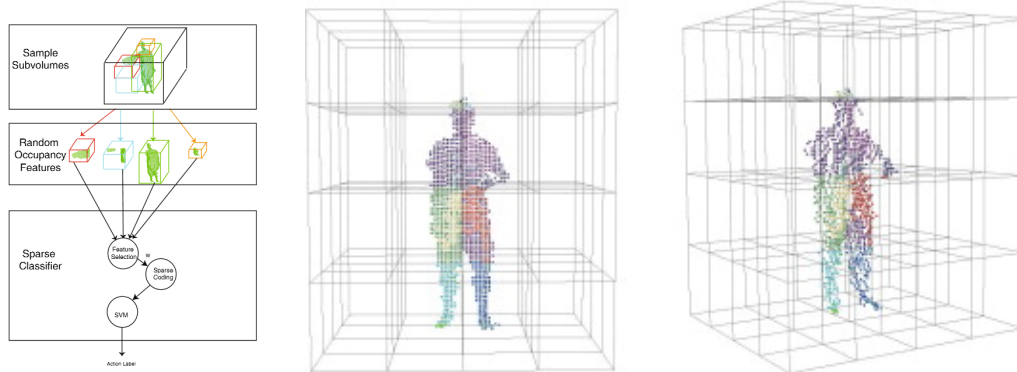
A szakirodalomban előforduló módszerek nagy része egy csontvázat illeszt a testre, vagy legalább a hajlatokat és végtagok végét egy-egy pontot reprezentálja. Nagy előnyt jelent ilyen esetben a Kinect szenzor használata, amelyben egy beépített algoritmus azonnal kiszámítja valós időben a csontváz pontjainak koordinátáit. Talán legegyszerűbb leírója ezek után egy cselekvésnek a csontváz pontok páronkénti távolságának időbeli változása, bonyolultabb módszerek esetében pedig a csontvázból kinyert pontokból különböző jellemzőket készítenek, majd arra főkomponens analízist alkalmaznak [39] vagy hisztogramokkal számolnak tovább [38] és így ezek alapján történik az osztályozás.



1.3. ábra. Cselekvések csontvázzal történő reprezentációja [25].

1.2.2. Egyéb módszerek

A csontváz illesztéstől eltérő információt tud nyújtani a személyekről készített pontfelhő. E pontfelhő kellő pontossággal leírhatja a mozgást, amennyiben annak sűrűsége eléri egy adott szintet. Több eltérő megközelítés is született a pontfelhő alapú feldolgozásnak. Ezek egyike a foglaltsági minták [1, 35, 37] használata, amely a test körüli teret felosztja valamilyen módon különböző területekre és az ott található pontok számából következtet a cselekvés formájára. Egy másik megközelítés a pontfelhőn, vagy az abból készített mélységképen történő hisztogramozás. Erre láthatunk példát [26] és [27] esetében, ahol 4D normákat és a pontfelhő adott részein elvégzett főkomponens analízis eredményeit gyűjtik hisztogramokba. Munaro és társai [23] szintén felosztották a személyek körüli teret egy adott felbontáson, majd az egymást követő két időpillanathoz tartozó pontfelhőkben a pontokat megfeleltették egymásnak, ez által nyerték ki 3D motion flow információkat, és végeztek osztályozást az adott cellákon ilyen módon kinyert információkból készített különböző jellemzőkkel. Az alábbi ábrán látható két minta a megjelenés alapú cselekvés felismerésre.



1.4. ábra. (a) Egy foglaltsági minta módszer, a *random occupancy patterns* [1], valamint (b) a 3D motion flow cella felosztása [23].

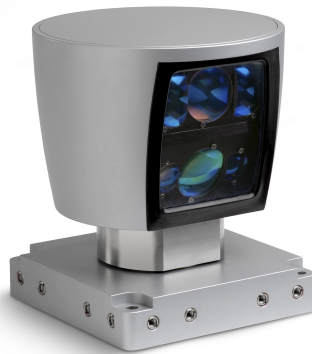
Egy szintén érdekes megközelítése a 3D konvolúcióval történő osztályozása a cselekvéseknek [14]. Itt a szerzők optikai videó kamerás felvételeken egy konvolúciós neurális hálót tanítottak be úgy, hogy az egymást követő képkockákból egy 3D szerkezetet készítettek, melynek a harmadik dimenziója az idő, így a 3D konvolúcióban használt képsorozat téglatestek (i, j, t) -edik eleme a t -edik időpillanathoz tartozó kép (i, j) -edik eleme. E megoldás azért kiemelendő, mivel a dolgozat későbbi fejezetei foglalkoznak és használják a konvolúciós neurális hálózatokat, ám annak csak 2D-s változatát.

2. fejezet

LiDAR

2.1. A LiDAR szenzor

A LiDAR egy mozaik szó, Light Detection and Ranging megfelelője. Ez egy time of flight típusú lézerszkenner. A szakdolgozatban említett mérések egy, a 2.1. ábrán látható Velodyne HDL-64E típusú LiDARral lettek rögzítve. Ez az eszköz egy forgó, többsugaras (rotating multi-beam, RMB) szenzor, amely 5-15Hz FPS-sel (frame per second) képes rögzíteni összesen 64 sugárral 360°-os horizontális és 26,8°-os vertikális látószöggel másodpercenként akár 1.3 millió pontot. E pontfelhő méret lehetőséget ad a környezet igen részletes elemzésére. Mivel nagy látószöge van jól alkalmazható a videó megfigyelésben.



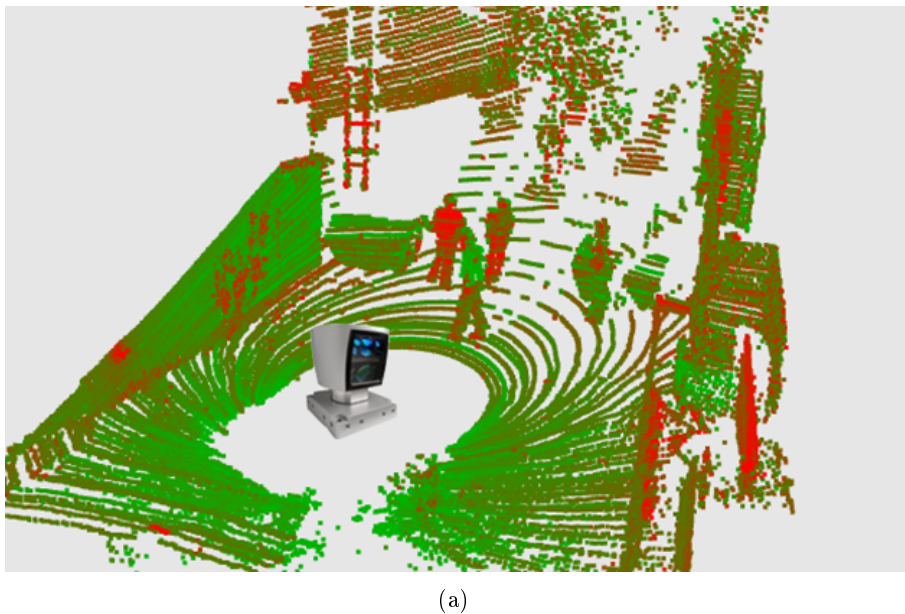
2.1. ábra. *Az általunk is használt Velodyne HDL-64E.*

A Velodyne forgó többsugaras eszköztől eltérő LiDARok is jelen vannak a piacon, valamint a több információ kinyerése érdekében sokszor egyéb szenzorokkal is megsegítik őket. Mobil térképező rendszerekben például a LiDAR mellé kamerát, IMU-t, szinkronizációs egységeket szerelnek. Különböző szakterületekre többféle LiDAR is készült, vannak olyan változatai, amelyek a statikus környezetről készítenek részletes modellt, a mozgó objektumok pedig csak elmosódva jelennek meg rajta, másikkak légi felvételek készítésére lettek kialakítva. Használják a szenzort többek közt geológiában, erdőszetben, városi környezet felmérésében, hadseregben. Két jelentős típusa a LiDARnak a térszkennelésre és a jármű

navigációra alkalmasak. Utóbbi funkciót használja ki példa a Google önjáró autója¹. Itt a járműre felszerelt LiDARt a környezet feltérképezésére, objektumot felismerésére, azok kikerülésére alkalmazzák. Méréseinket egy jármű navigációra alkalmas LiDARral végeztük a földre helyezve, döntötten felfelé néző video-felügyeleti konfigurációban volt használva. Szokásos használata közben, és az általunk használt döntött elhelyezés látható a 2.2. ábrán.



2.2. ábra. *LiDAR használati módjai. a) jármű navigációra használt, nem döntött. b) döntött, video-megfigyelési használat.*



2.3. ábra. *A LiDAR használata közbeni pozíciója egy az általa készített pontfelhőben feltüntetve.*

2.1.1. Előnyei és hátrányai

Előnyei:

- Éjjel-nappal alkalmazható: mivel lézeres technológia, így az általános fényviszonyok

¹<https://www.youtube.com/watch?v=YaGJ6nH36uI>, 2015. október 26.

nem befolyásolják, tehát ebben jobb a hagyományos kameráknál, amelyek éjjel nem alkalmazhatóak

- Több, mint 100 méteres hatótávolság
- Kihelyezése egyszerű: 3D adatszolgáltatáshoz egy LiDARt nem kell olyan kalibrációnak alávetni, mint ha több kamerás rendszert szeretnénk felépíteni

Hátrányai:

- Jelenleg nagyon drága technológia: ez várhatóan a jövőben már nem így lesz
- Ködben, nagy esőben nem működik jól

Fontos megjegyezni, hogy az egykamerás (mono) rendszernél a háttér leválasztás után nincs lehetőségünk az objektummal a 3D térben geometriai műveleteket végezni, LiDAR által készített pontfelhőben könnyedén skálázhatjuk és forgathatjuk az objektumokat. Továbbá az egykamerás rendszernél a járókelők takarása esetén csupán mozgás alapú háttérkiemelés a részleges takarásban lévő alakzatokat egy objektumba olvaszthatja, míg a LiDAR esetén ezek a mélységinformáció segítségével elkülöníthetők. A járásfelismerésről szóló irodalomban több eljárás érzékeny a kinyert sziluett kamerához viszonyított orientációjára, többször ezért oldalnézeti sziluettekben indulnak ki a felismerési tesztek [9, 15]. Ennek a feltételnek a valós életben való teljesülését egy egyszerű kamerával nem tudjuk biztosítani, mivel egy utcai környezetben nem garantált, hogy mindenki párhuzamosan haladjon a kamera képsíkjával. Több kamerás rendszerekben már lehetőség nyílik 3D geometriai információ kinyerésére, de ilyen esetekben a kamera kalibrációs és sztereo rekonstrukciós hibák már nagyban befolyásolhatják a működést. A Velodyne-LiDAR kihelyezése ezzel szemben, mint már említettem az előnyöknél, egyszerű, csak a kijelölt helyre kell telepíteni és már működik is, nincs szükség a terepen újabb beállításokat végezni, valamint a pontfelhő transzformációval mindenkor kinyerhető egy oldalnézeti kép, amennyiben elegendő mennyiségű pontból áll az objektum.

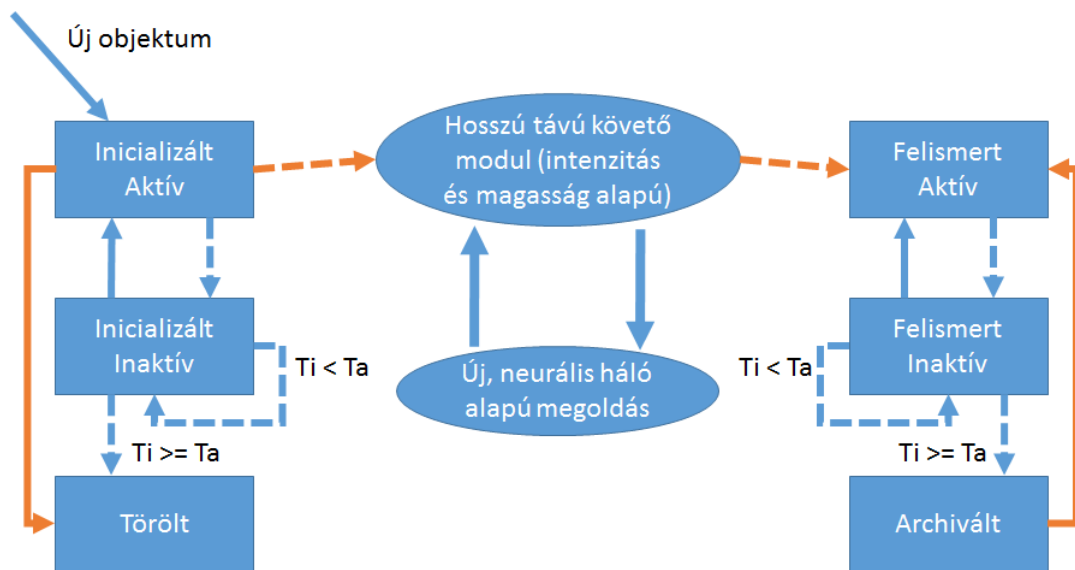
2.2. LiDAR keretrendszer

[2] cikk alapján az MTA SZTAKIban elkészült keretrendszer implementációban a LiDAR felvételeken lehetőség van az emberek pontfelhőit leválasztására a háttérből, a járókelők követésére, valamint újra felismerésre, amennyiben elhagyták a LiDAR látóterületét és újra megjelentek.

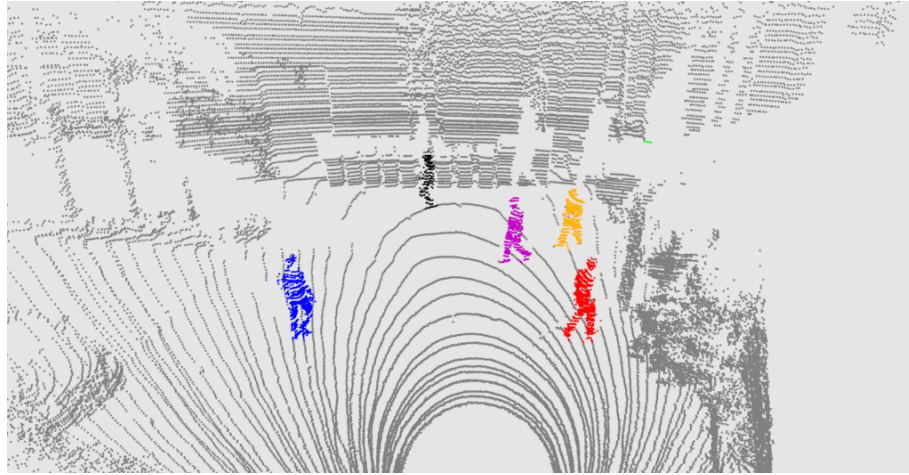
A keretrendszer különböző funkciókkal bír. A dolgozat témájához az objektumfelismerő része kapcsolódik, amelyben végig tudjuk követni a mérést vizuálisan, a pontfelhőt megjelenítő ablakban. Ezen felül látható még a tesztalanyok trajektóriája, valamint az adott pillanathoz tartozó sziluettjük, sziluett lenyomatuk, azonosító számuk, és még néhány, a korábbi felismerő modulhoz tartozó adat. Minden LiDAR mérés mellett készült egy felvétel egy videokamerával is, amely csak az eredmények verifikációjára szolgál, a felismerésben nem vesz részt.

A keretrendszerben egy nagyon fontos része az objektumok pontfelhőinek háttérből való leválasztását végző modul, amely működését [2] részletezi. A már 1. fejezetben megismert módszerek többsége a sziluettekre épít. Ezek kinyerése kulcsfontosságú kérdés.

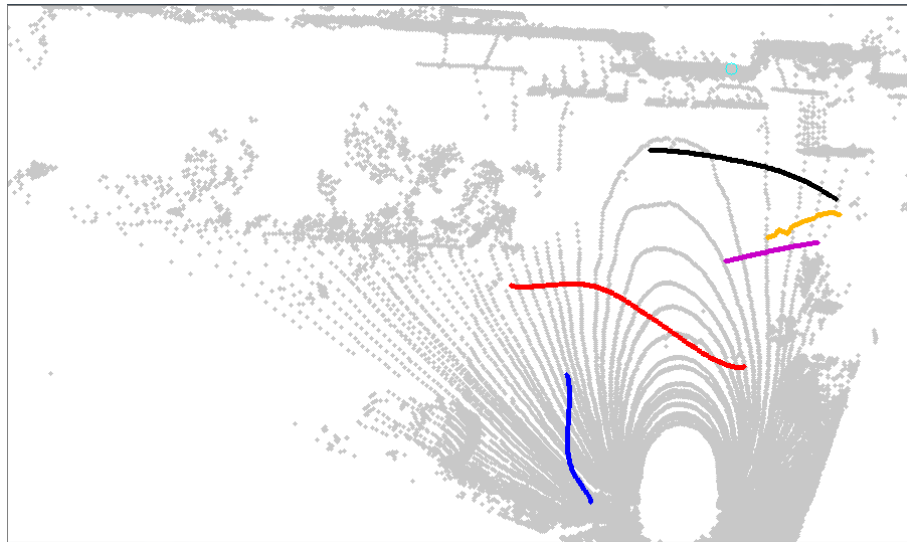
Az objektumkövetést és újrafelismerést két modul végzi: egy rövid távú és egy hosszú távú követő modul. A rövid távú modul alapja egy hagyományos Kálmán szűrő. Az objektumok eddigi pozícióinak ismerete alapján meg tudjuk becsülni a következő pontot, ahol valószínűleg meg fog jelenni. Minden egyes emberre kiszámítva ezt, képezhetünk egy négyzetes mátrixot a becsült és valós pozíciókról, majd a magyar módszerrel teljes párosítással azonosíthatjuk a járókelőket. Ezzel a módszerrel nagy pontossággal meghatározhatjuk a már felismert egyének identitását a mozgása során. A hosszú távú modul az újra felismerésért felelős. Egy objektum lehet inicializált és felismert szakaszban, valamint lehet aktív és inaktív. Ez alapján négy állapotot vehetnek fel. A négy állapotot és folyamatábrát a 2.4. ábra mutatja. Kezdetben minden objektum az inicializált aktív állapotba kerül. Amennyiben sikerül a felismerés, úgy a korábbi archivált állapot aktiválódik és az új objektum törlődik, ha nem sikerült a felismerés, úgy egy új objektumként létezik tovább aktív felismert állapotban. Aktív állapotból inaktívba úgy kerülhetnek az objektumok, ha a LiDAR sorozaton adott ideig nem láthatóak. Egy adott időn belül, ha továbbra is inaktív marad egy objektum, akkor archiválttá válik. Az archivált objektumokat a rövid távú modul nem követi. A 4. és 5. fejezetben ennek a hosszú távú felismerő modulnak az azonosításhoz szükséges biometrikus jellemző kiválasztása történik.



2.4. ábra. A keretrendszer folyamatábrája. Folytonos kék vonal a rövid távú modul sikeres, szaggatott a sikertelen követése. Folytonos narancs vonal a hosszú távú modul sikeres, szaggatott a sikertelen követése. T_i : inaktívan töltött idő, T_a : archiválási időközszög. Az általam fejlesztett új, neurális háló alapú hosszú távú követő modul cseréli le a régi intenzitás és magasság alapú módszert.



(a)



(b)

2.5. ábra. A nyár5 szekvencia pontfelhő-sorozatának (a) egy pillanatnyi képe és a hozzá tartozó (b) trajektóriák. Minden egyes szereplőhöz más-más színkód tartozik, ezzel a színkóddal a trajektóriákat be tudjuk azonosítani. A trajektóriák egy felülnézeti pontból láthatók.

3. fejezet

Gépi tanuló eljárások

3.1. Mély tanulás különböző módszerei

A mély tanulás (deep learning) napjainkban egy sokak által kutatott területe a gépi tanuló rendszereknek. Több eltérő felépítésű neurális hálót fejlesztettek ki, amelyek tanítási eljárásai is különbözhetnek. A legjobb teljesítményt napjainkban a konvolúciós neurális hálók (Convolutional Neural Network, CNN [3]) adják. Ettől teljesen eltérő felépítésű az autoencoderek egymásra építéséből kapható háló [3], amely teljesítménye elmarad a CNN eredményeitől, de meghaladja a korábbi hagyományos neurális hálókét.

3.1.1. Convolutional Neural Networks

A hagyományos többrétegű neurális hálónál (pl. MLP) általánosságban az egymást követő rétegek minden neuronja össze van kötve. A lokális kapcsolatok ötlete Fukushima Neocognitronjában [7] 1980 körül megjelent. Később LeCun és társai [22] ezt az ötletet követve bevezették a konvolúciós neurális hálót. Ezen háló neuronjai úgy helyezkednek el, mint egy kép pixelei, a neuronok közötti kapcsolatok súlyai pedig közös súlyok, együtt változnak a tanítás alatt. Ezek a közös súlyok úgy képzelendők el, mint egy konvolúciós ablak hagyományos képfeldolgozó algoritmusban. A neurális háló tanítása során pedig ezen konvolúciós ablakok súlyait tanítjuk. Ezek az ismétlődő egységek közös paraméterezése (súlyok, eltolások) által pedig egy jellemző teret (feature map) alkotnak. Egy konvolúciós háló tanításánál a neuronok száma, a konvolúciós "ablak" mérete mellett a jellemző terek számát is meg kell adnunk. Ezzel különbözőképp szűrt képeket kapunk az eredeti képből. A kutatások szerint ez a módszer - ahogyan a CNN-ek működnek - hasonlítanak valamelyest az emberi látás mechanizmusához.

A konvolúciós hálók nem csak konvolúciós rétegekből (convolutional layer, CL) állnak. Ezen kívül még alul-mintavételező (sub-sampling, SS) rétegek és MLP-kenél is megtalálható teljesen összekötött rétegek (fully-connected layer, FCL) is találhatóak. A SS rétegek a jellemző tér dimenziójának csökkentésére használható. Itt a kívánt méretű ablakkal a képen átlagolás, maximum számítás történik, ez által robusztusság is növelhető (például eltolásokra lesz kevésbé érzékeny). A szokásos CNN felépítésben általában konvolúciós és alul-mintavételező rétegek váltakoznak, majd a kívánt jellemző tér és dimenzió elérésekor

teljesen összekötött rétegeket alkalmaznak.

Dropout alkalmazása

A dropout [31] módszer lényege, hogy egyes neuronok nem mindig vesznek részt a tanítás során, adott valószínűsége van annak, hogy a kimenetén a megfelelő értéke, vagy nulla jelenik meg. Ennek a trükknek köszönhetően a túltanulás valamilyen szinten elkerülhető, és javít az eredményeken is. Ehhez hasonló, erre épülő módszer a maxout, amelyet Bengio és társai mutattak be [13], szintén javít az eredményeken.

3.1.2. Stacked Denoising Autoencoders

Autoencoder

Az autoencoderek egyszerű többrétegű perceptronok, amelyek kimenete ugyan annyi neuronból áll, mint a bemenete, és egy adott bemenetre várt eredmény maga a bemenet. Amennyiben egy ilyen felépítésben egy rejtett réteget alkalmazunk, amelynek kevesebb a neuron száma, mint a bemeneté, akkor ezen réteg egy csökkentett dimenziójú reprezentációját nyeri a tanított adatnak, amelyből még fel lehet építeni az eredeti értékeket. Ezzel nemlineáris főkomponens analízist végezhetünk, amely sok esetben jobban tudja reprezentálni az adathalmazt, mint egy általános PCA.

Denoising

Az autoencoderek teljesítményének növeléséhez egy egyszerű módszert vezettek be. Amennyiben a bemeneti adat valamennyi értékét véletlenszerűen választva maximális vagy minimális értékre állítjuk, de az elvárt kimenetet továbbra is az eredeti adatra irányul, úgy kis hibákkal terhelt adatból is vissza tudja állítani a valós értékeket, ezáltal robusztusabb működést biztosít, és jobb eredményt produkál a tanítás. [36] adatai alapján jobb eredményt ért el egy olyan Stacked Autoencoder, amely e módszert alkalmazza, mint amelyik nem, valamint Deep Belief Network-öknél és SVM-eknél is jobb eredményt produkált.

Autoencoderek egymásra építése

Az autoencoderek által tehát dimenziót tudunk csökkenteni, mindezt úgy, hogy abból még visszaállítható az eredeti minta. Egy ilyen réteg tanítása után a súlyainak fixen tartása mellett lehetőségünk van annak kimenetén egy újabb autoencoder tanítására, így még kevesebb dimenzióba kerül az adat. Mindebből következik, hogy az autoencodereket egymásra építve taníthatjuk azokat rétegről-rétegre, majd egy kívánt szinten a nem ellenőrzött tanítást felválthatja az ellenőrzött tanítás és osztályozás. Mivel túl nagy hálóknál nehéz a tanítás véletlenszerűen beállított súlyokkal, így a betanított nem ellenőrzött tanítással nyert súlyok lehetnek az inicializáló értékek, amelyből további tanítással pontosíthatunk a hálón. Az így kapott architektúrát hívják Stacked Autoencodernek. Ezt lehet tovább fokozni a denoising alkalmazásával.

3.1.3. Egyéb módszerek

A két említett deep learning módszeren túl vannak egyéb más lehetőségek is, például a Restricted Boltzman Machine-eken (RBM) alapuló Deep Belief Network-ök [3], amelyek hasonlóan felépíthetők RBM-ekből, mint a Stacked Autoencoderek, de egy réteg tanítása eltér egy autoencoder tanításától, mivel az RBM-ek energia alapú modellek [3] és így nem hibavisszaterjesztés alapú a tanításuk.

3.2. A mély tanulás eddigi eredményei, alkalmazásai

3.2.1. Arcfelismerés

1997-ben Lawreace és társai [32] konvolúciós hálót alkalmaztak arcfelismerésre, ez által az addigiaknál magasabb felismerési arányt sikerült elérniük.

3.2.2. MNIST adatbázison elért eredmények

Az MNIST adatbázis kézzel írt számjegyek sokasága. Ez egy fontos banchmark a tanuló rendszerek eredményeinek mérésére. Sok éven keresztül az adatbázison elért eredmények stagnáltak, míg 2010-ben, majd 2011-ben sikerült csökkenteni. 7 konvolúciós neurális háló közös eredményével [6] sikerült levinni a hiba értékét 0.27 százalékra.

3.2.3. Egyéb alkalmazások

Egy másik banchmark adatbázis az ImageNet¹. Több ezer különböző osztályba besorolható több millió képet tartalmaz, ezen felül pedig éves felismerési versenyt is hirdetnek (Large Scale Visual Recognition Challenge). Ezen a versenyen is alkalmaznak konvolúciós neurális hálókat [18, 19], amely ezen versenyen is a legjobb eredményeket produkálja.

Valós környezetben is elképzelhető alkalmazásai is akadnak a CNN-nek. Erre egy példa az utcai jelzőtáblák felismerése [5, 30]. Ciresan-ék és LeCun-ék eredményéből látható, hogy a táblákat 99 százalékos eredménnyel felismerő architektúráknak nincs szüksége előzetes jellemzők kinyerésére, valamint az eredményt növelni lehet tovább CNN és MLP együttes becslésével.

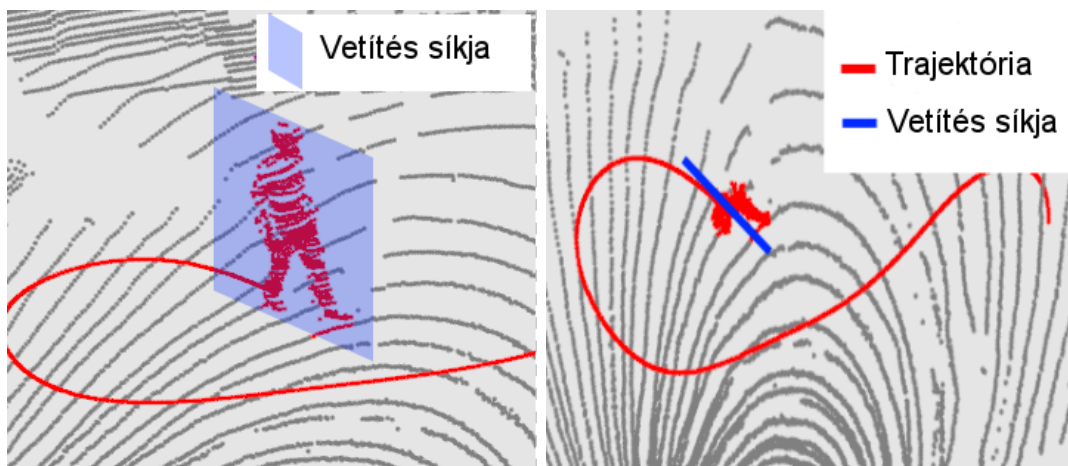
¹<http://www.image-net.org>

4. fejezet

Különböző járásjellemzők vizsgálata

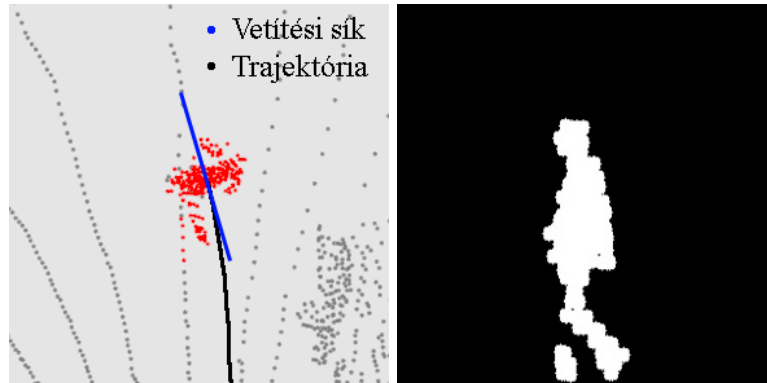
4.1. Sziluettek kinyerése

A 2.2. fejezetben bemutatott keretrendszer alapján lehetőség nyílik az egyes emberek pontfelhőinek elérésére, ezen felhőket pedig könnyedén tudjuk transzformálni, ami fontos az oldalirányú sziluettek kinyeréséhez. A járásról oldalsó nézőpontból tudjuk a legtöbb információt megkapni, ez látható abból is, hogy a következőkben bemutatott módszerek mindegyike oldalnézetből vizsgálják a járást. Az egyes emberek pontjait el tudjuk forgatni bármilyen irányba, például a trajektória mentén, így képesek vagyunk a pontok járás irányára merőleges vetítésére. A 4.1., 4.2. és 4.3. ábrákon látható a forgatás iránya, valamint egy forgatás utáni és előtti sziluett azonos időpontból a *nyár5* szekvencián.

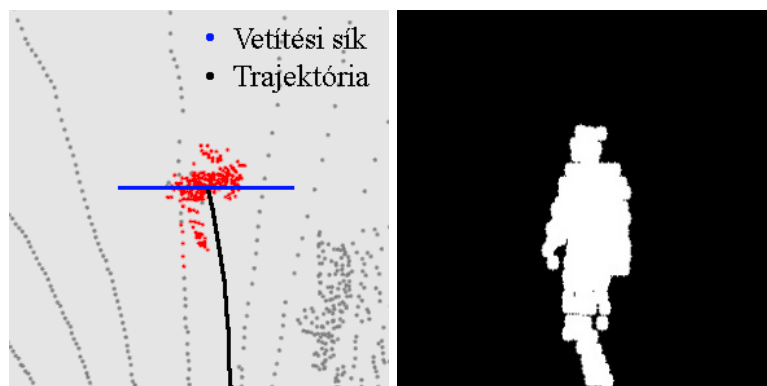


4.1. ábra. A trajektóriára történő forgatás síkja oldal- és felülnézetből.

A 3D-s pontfelhőből ez a trajektóriára merőleges vetítés képezi számunkra a 2D-s sziluetteket. A pontokból viszont képesek vagyunk mélységi információkat is kinyerni, így mélységképeket is tudunk gyártani belőle, valamint a nyers pontfelhőt is fel tudjuk használni. A következő módszerek ezeket a sziluetteket, mélységképeket, pontfelhőket használják.



4.2. ábra. Sziluett vetítés a trajektória érintőjével párhuzamos síkra .



4.3. ábra. Sziluett vetítés egy a személy "középpontjára" eső LiDAR sugárra merőleges síkra.

4.2. A sziluett lenyomat

A sziluett lenyomat módszer esetében a bináris sziluett körvonala az alapvető leíró. A kontúrból elkészítünk egy szélesség vektort, amely értékei a sziluett egyes soraiban a bal és jobb szélső pixel távolsága. Az egymás utáni képeken így képzett vektorokkal készítünk egy új képet, ez lesz a sziluett lenyomat, amelyen minél világosabb egy pixel, annál nagyobb a szélesség. Az így kapott sziluett lenyomatok periodikus mintázatot vesznek fel, ahogy a járás is periodikus mozgás.

4.2.1. Dynamic Time Warping

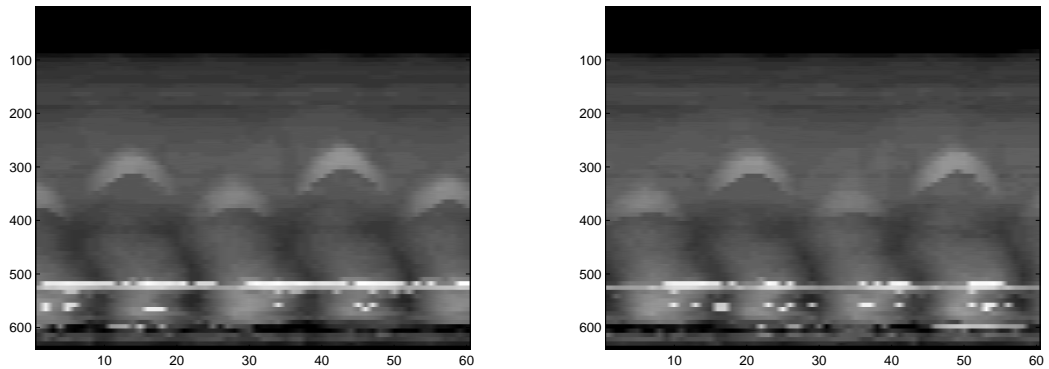
A DTW algoritmus idősor analízisnél alkalmazott módszer, mely két - időben és sebességben is akár eltérő - sorozat közötti hasonlóságot méri. A DTW-t például beszédfelismerésben alkalmazzák, de használatos más egyéb videó- és hanganalízisben is. Esetünkben egy többdimenziós DTW algoritmussal kell dolgoznunk. Erre általános módszer az egyes dimenziók külön-külön ellenőrzése egy-egy 1D-s DTW-vel, majd a különböző dimenziók távolságát összegzi az nD-s változat. A Dynamic Time Warping algoritmussal történő tesztek MATLAB környezetben végeztem.

4.2.2. Tesztek

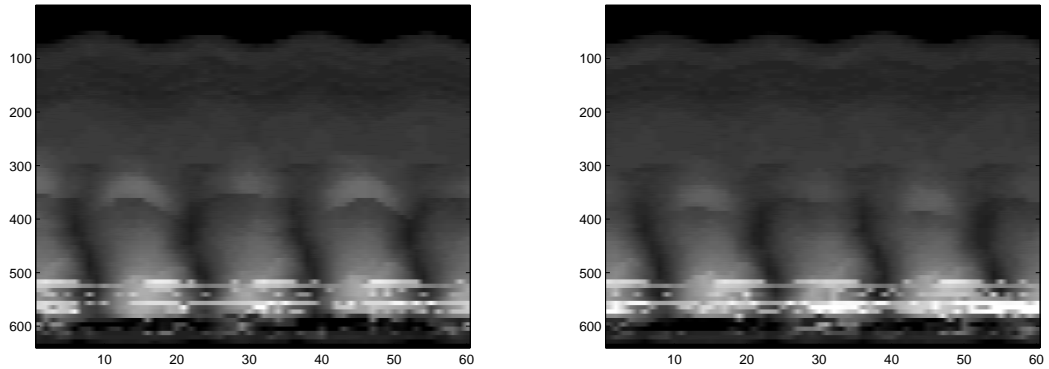
MoBo tesztek

A MoBo adatbázis a Carnegie Mellon University által készített mintahalmaz. Ebben jól ellenőrzött körülmények között rögzített felvételeket láthatunk futópadon sétáló emberekről különböző nézetekből, mint például oldal, hátul, ferdén hátulról, előlről stb..

A MoBo-n végzett tesztek ígéretes eredményt adtak, melyeken a magyar módszer minden esetben megtalálta a helyes párosítást. Az 4.4 és 4.5 ábrán látható két tesztalany sziluett lenyomata a tanító és tesztelő mintákból.



4.4. ábra. *MoBo* mintasorozat 1-es alanyának tanító és teszt sziluett lenyomata.



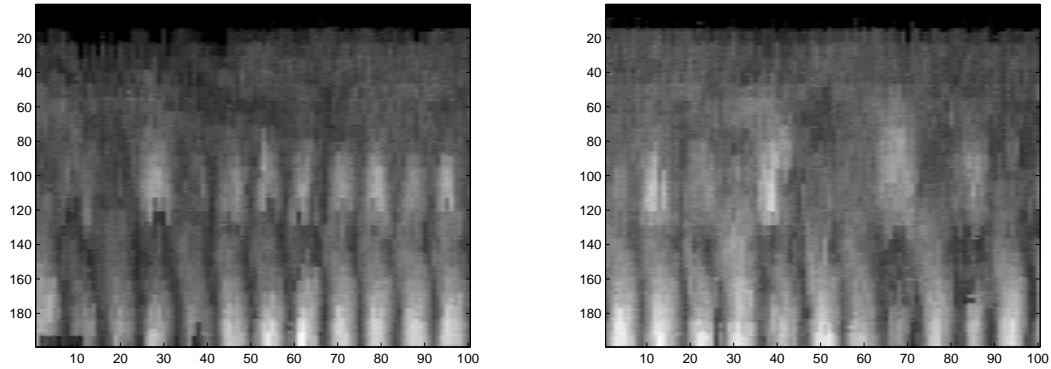
4.5. ábra. *MoBo* mintasorozat 2-es alanyának tanító és teszt sziluett lenyomata.

LiDAR tesztek

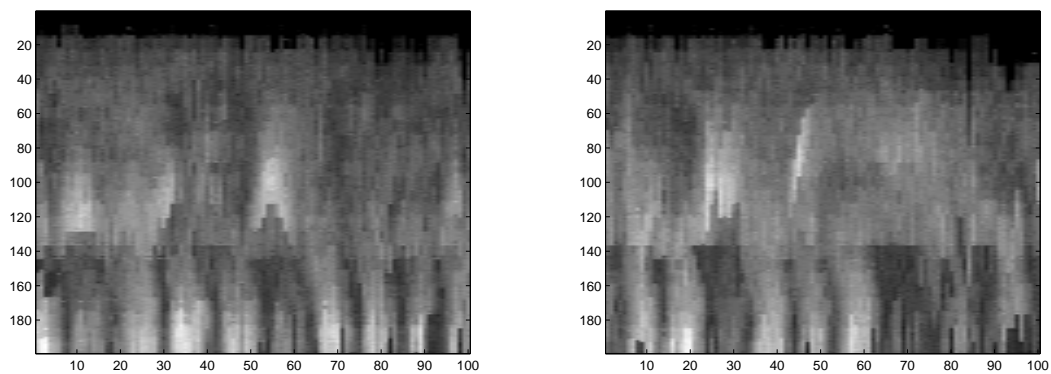
LiDAR szekvenciákon várhatóan rosszabb eredményt ad a módszer, mint a MoBo adatbázison. Mivel a LiDAR pontfelhőből készített képeken láthatóak lyukak, más torzulások, így a szélesség vektorban ezek hibákat generálhatnak. A 4.6. és 4.7. ábrákon látható LiDAR

mérésekből kinyert sziluett lenyomatok két különböző emberre. Jól láthatóan a sziluett lenyomatok minősége, és a teszt-tanító minták közötti hasonlóság jelentősen eltér a MoBo-ban tapasztaltakhoz képest.

A sziluett lenyomat tesztelése során a tanító mintahalmazból készült öt lenyomat minden személyről, majd ehhez lett hozzá mérve a DTW-vel a tesztkép. A CMU MoBo adatbázison ezen tesztek közül mindegyik sikeres volt, tehát ez a módszer a jó minőségű mintákon megfelelő. Ugyan ilyen alapon a LiDAR sorozatokon végzett tesztek eredménye lényegesen gyengébb, értékeik a 5.1. tábla első oszlopában láthatók. Vegyük észre, hogy a *tél4* sorozaton a MoBo adathalmazon elért eredményekhez hasonló értéket produkált a teszt. Ez a mérés volt az első 6 szekvenciák közül a legjobb minőségű képeket generáló. Ezen eredmény alapján nem lehet kijelenteni, hogy a tesztelési módszer lenne az oka a gyengébb eredményeknek LiDAR környezetben.



4.6. ábra. *Nyár5 mintasorozat 0-ás alanyának tanító és teszt sziluett lenyomata.*



4.7. ábra. *Nyár5 mintasorozat 2-es alanyának tanító és teszt sziluett lenyomata.*

4.3. Gait Energy Image

A választott jellemző, amely alapján a járásfelismerést végzi a rendszer a járás energiakép, (Gait Energy Image, GEI) [9]. Egy GEI az egymást követő bináris sziluettek átlaga az alábbi egyenlet alapján, ahol G az így nyert jellemzőt, S pedig a bináris sziluetteket jelöli.

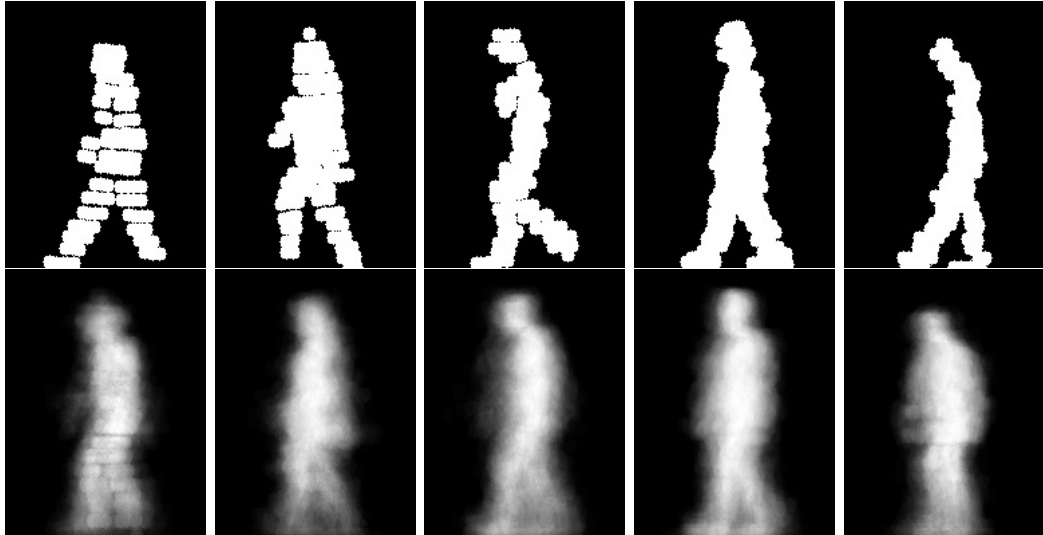
$$G(x, y) = \frac{1}{N} \sum_{t=1}^N S_t(x, y) \quad (4.1)$$

A Gait Energy Image csökkenti a zajok hatását, ezáltal robusztusabbá teszi a jellemzőt. Han és Bhanu által bevezetett módszerben az így kinyert szürkeárnyaltos képeken először egy főkomponens analízissel (PCA), majd pedig többszörös diszkrimináns analízissel (multiple discriminant analysis, MDA) csökkentették a dimenziót. Az így kapott eredményről pedig egyszerűen a tesztek és tanító minták távolságát mérték. A járás energiaképeket egy-egy járásciklusból állították elő, ehhez meg kellett határozni a ciklus elejét és végét, és a köztes sziluetteket átlagolták. A valós mintákon kívül bevezettek szintetikus mintákat, amelyekkel növelték a mintahalmazt, valamint a felismerést külön-külön végezték a valós és szintetikus képeken, majd ezek eredményeit vették közös döntéshozatalhoz. A szintetikus képek előállításánál szerepet játszott, hogy a rendelkezésükre álló képek alján torzulások voltak láthatók, a szintetikus képek előállításánál cél volt ezek kiküszöbölése is.

Mivel a kültéri LiDAR nem képes olyan jó minőségű képeket készíteni, mint egy beltéri, jó fényviszonyok melletti környezetben felszerelt kamera, így több változtatást is kellett tennünk az eredeti GEI módszerben. Először is nem választjuk le a járási ciklusokat, mint ahogy azt Han és Bhanu tette, mivel a LiDAR felvételeken erre nem mindig van lehetőség az átfedések és más jellemző problémák miatt, helyette hosszabb tartományt átlagolunk, másodszer pedig nem készítünk szintetikus sablonokat a mintahalmaz növelésére, javítására. Nincs szükség a képek alját levágni, mivel nem látható a LiDAR felvételeiből készített sziluettek alján olyan minőségű romlás, mint ahogy [9] cikkben látható. Ezzel szemben választunk véletlenszerűen 100 kezdő pillanatot, és innen indítva átlagoljuk a bináris képeket. Az így kialakult jellemzőt nevezzük LiDAR-alapú járás energiaképnek (LiDAR-based Gait Energy Image, LGEI). A 4.8. ábrán láthatunk 5 személyről egy-egy LGEI-t, és a hozzájuk tartozó átlagolási intervallumból választott sziluetteket. Az LGEI képekből PCA és MDA általi dimenziócsökkenés után vektor összehasonlításból kapott eredményt felhasználó módszert a későbbiekben LGEI+VC (vector comparison) néven említem.

4.4. Depth Gradient Histogram Energy Image

A Depth Gradient Histogram Energy Image (DGHEI) [12] a GEI átlagoló ötletét követi, bár nem a nyers képeken, hanem azok feldolgozása utáni jellemzőkön végzi azt. A módszert a Kinect szenzor által készített mélységképekre fejlesztették, ahol az egyes képeken mélységi gradienseket nyerünk ki és tároljuk el hisztogramokban, majd ezeket a hisztogramokat átlagoljuk a teljes járásciklusokon. Dimenziócsökkentés céljából a GEI-nél látható PCA és MDA párost használták, osztályozáshoz pedig a Nearest Neighbor osztályozót. A 4.3.



4.8. ábra. Nyár5 5 tesztalanyának sziluettjei és hozzájuk tartozó LGEI-k.

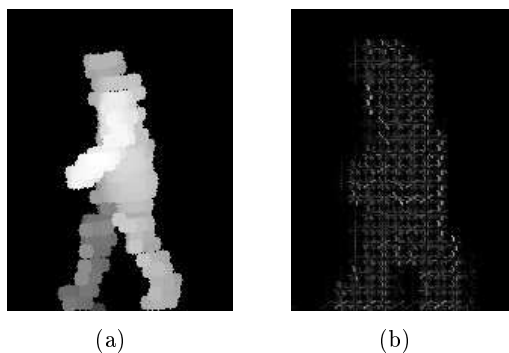
fejezetben már említett megfontolásokból nem a teljes cikluson, hanem egymást követő 80-100 képen történt az átlagolás.

A személyenkénti 200-400 pontból álló felhőkből képesek voltunk elfogadható minőségű mélységképeket generálni. Ilyen képek láthatók a 4.9. ábrán. A DGHEI a GEI-től több információt tartalmaz, mivel itt már mélységi adatok is be lettek vonva a számításokba, valamint előállításuk az elődjéhez hasonlóan egyszerű, habár a gradiensek és belőlük képzett hisztogramok előállítása időigényes. A 5.1. táblázatban látható, hogy a DGHEI módszer két esetben is felülmúlja a GEI alapú, akár neurális hálós, akár az eredeti vektor távolságos módszereket, ez által kijelenthetjük, hogy akár egy jó alternatívája is lehet a kiválasztott eljárásnak. Sűrűbb pontfelhők esetén nagy valószínűséggel jobb teljesítményt is adna a GEI-től.

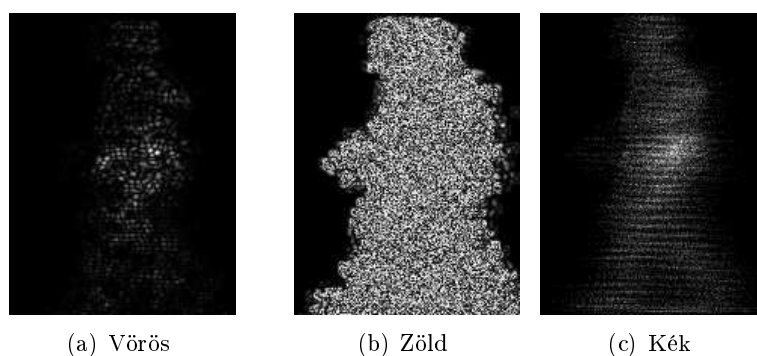
4.5. Color Gait Curvature Image

Ahogy a DGHEI, úgy a Color Gait Curvature Image (CGCI) [33] is a Kinect felhasználásával lett fejlesztve. A Kinect szenzor nagyságrendekkel nagyobb pontfelhőket készít az általunk használt LiDAR-tól, mégis érdemes kipróbálni ezen új, pontfelhő alapú módszert. Gauss és átlag görbületi jellemzők [34] kinyerésével, valamint a pontfelhő sűrűségével történik a CGCI RGB képének előállítása, nevezetesen a két görbületi jellemzőből R és G komponenseket állít elő, a pontfelhő sűrűségéből készített kép pedig maga a B komponens lesz, majd 2D koszinusz transzformáció és 2D-PCA alkalmazásával csökkentjük a dimenzióját az R, G és B csatornáknak külön-külön. Az osztályozást a teszt és tanító minta csatornáiból nyert jellemző vektorok abszolút különbségének súlyozott összegével történik.

A pontfelhő méretének jelentős csökkenése súlyos teljesítményromlást eredményezett, ez jól látható a 5.1. tábla értékeiből. A CGCI kép vörös, zöld és kék csatornája látható a 4.10. ábrán.



4.9. ábra. A nyár5 szekvencia egy tesztalányának (a) mélységképe és (b) DG-HEI képe [12].



4.10. ábra. A nyár5 szekvencia egy tesztalánya CGCI [33] képének komponense. (a) és (b) a görbületi jellemzőkből [34] készített kép, (c) a pontfelhő sűrűsége.

4.6. Gait Energy Image neurális hálókkal

Annak ellenére, hogy a GEI egy 2006-ban bemutatott jellemző, még mindig egy versenyképes megközelítés, sokan továbbfejlesztették és 2015-ös cikkek [29] is használják alapötletüként. Érdekes lehet tehát vele foglalkozni, mivel magában a GEI egy jól működő, robusztus konstrukció. A mély tanulás, mint említettem a 3. fejezetben, napjainkban a legjobb eredményeket hozza sorra különböző területeken, így érdemes kipróbálni a járásanalízisnél is. Az általam bevezetett módszer (LGEI+NN) két osztályozó rendszert használ. Az egyik az említett PCA-MDA alapján, de az így kapott csökkentett dimenziójú vektorokra egy hagyományos többrétegű perceptront építve ad eredményt, míg a másik egy mély tanulási módszerrel, a konvolúciós neurális hálóval végzi a felismerést. Ezen megoldások részletezése olvasható az alábbi alfejezetekben.

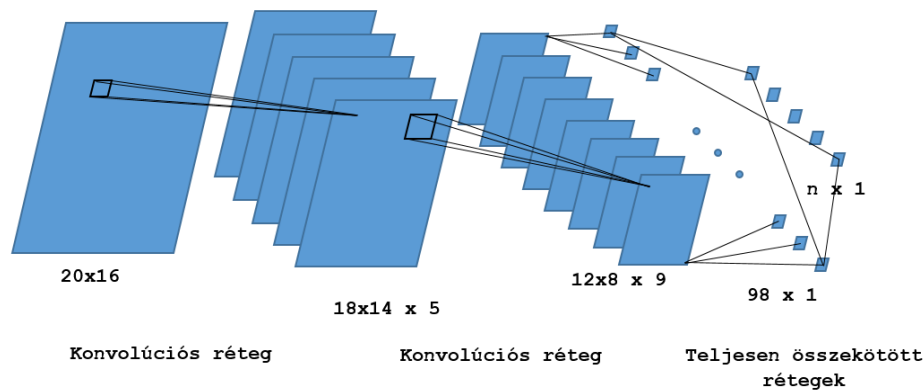
4.6.1. Konvolúciós neurális háló alapú megoldás

A korszerű mély tanulási rendszerek egyike, a már 3.1.1. fejezetben említett konvolúciós neurális háló. Ezzel state-of-the-art teljesítményt értek el az MNIST adatbázison, valamint arcfelismerésben és sok más területen is. Ezek alapján egy jó döntésnek tűnik ezt a tanuló

rendszert alkalmazni a járás felismerésére is.

A háló architektúrája

A konvolúciós háló jó paraméterválasztása nehézkes feladat. Erre az irodalomban sincs egységes jó megoldást, hogy hogyan érdemes felvenni a jellemző terek számát, a háló mélységét, konvolúciós "ablakok" méretét. Ennek következtében a változók hangolására egy rajintelligencia algoritmust alkalmaztam, amely a 4.6.3. fejezetben lesz részletesen bemutatva. Az így alakult végső megoldás látható a 4.11. ábrán.



4.11. ábra. A megvalósított CNN architektúra.

A megvalósított architektúrában a bementi képeink 20 pixel magas és 16 pixel szélesek. Ez által az első, konvolúciós réteg a 20×16 -os méretű képekből, 5 jellemző teret készít 18×14 -es méretű képeket 3 -as konvolúciós ablak mérettel. Ezek után egy újabb konvolúciós réteg következik, 7 -es ablak mérettel, 9 kimeneti jellemző térrel. Az ez által keletkezett 9 12×8 -as méretű kép értékeiből végül egy teljesen összekötött háló az $9 \times 12 \times 8$ bementből először 98 , majd pedig n darab kimenet keletkezik, ahol n a tesztalanyok száma. A tanítás ezek után az elforgatott és szűrt, méretében lezsugorított 20×16 pixeles képeken, 72 epochon keresztül történt.

Ez az architektúra kis mértékben eltér a szakirodalomban találhatóaktól. Ott általánosságban vehetjük, hogy egy konvolúciós réteget egy alulmintavételező réteg követ, és így tovább ezeket alternálva jut el végül a teljesen összekötött rétegegig. A háló megtervezésénél először egy ilyen felépítést próbáltam ki, amely egy konvolúciós, valamint egy alul-mintavételező réteget tartalmaz, de a tesztelések során a már említett két konvolúciós architektúra jobbnak bizonyult.

Fontos kérdés a hálók tervezésénél, hogy egy darab n kimenetű hálónk legyen, vagy n darab 1 kimenetű háló? Az irodalomkutatás alatt talált háló architektúrák mindegyike az 1 - n mintára épül, ám ezek nem valós időben hivatottak működni. A kérdés inkább az, mire is szeretnénk pontosan alkalmazni a rendszert. Amennyiben csupán a felismerés a fontos, és a módszer használhatóságát hivatott bemutatni, úgy az 1 - n minta megfelelő számunkra. A valós környezetben, például egy térre való kitelepítés során, ahol egy adott ember felbukkanására vagyunk kíváncsiak, ott az n - 1 minta lenne a megfelelő. Offline tanulás

során, ahol a tanítás ideje nem számít, nem probléma n darab különböző háló tanítása. Mindezek ellenére a bemutatott architektúra a LiDAR méréseken történő járásfelismerés működőképességét hivatott bemutatni, az irodalomban fellelhető konvolúciós hálókhoz hasonló mintára lett tervezve.

Megjegyzendő, hogy a mély tanuláshoz a szakirodalomban található alkalmazásoknál nagy méretű adathalmaz szükségességét említik. A területek, ahol a mély tanulás, pontosabban a konvolúciós neurális hálót alkalmazzák és versenyeket nyernek vele [19] a hálók architektúrája lényegesen bonyolultabb, nagyobb, nagyságrendekkel magasabb számú paramétert tartalmaznak. Az itt bemutatott felépítés ennél egyszerűbb, jelentősen kevesebb paraméterrel, így a nagy adathalmaz méretére is engedményt tehetünk.

4.6.2. Többrétegű perceptron alapú megoldás

Adat előfeldolgozás

A konvolúciós hálóval ellentétben a többrétegű perceptront (multilayer perceptron, MLP) nem a nyers képeken szokás alkalmazni. A 20×16 pixel méretű kép túl nagy dimenziójú, így előkerül a "curse of dimensionality" probléma. A dimenzió csökkentésére a főkomponens analízis (PCA) és a Fisher-féle lineáris diszkrimináns analízis¹ (FDA) alkalmazásával kerül sor. A PCA úgy csökkenti a dimenziót, hogy közben a lehető legjobban reprezentálja a kevesebb dimenzió az adatot, míg az FDA úgy, hogy a lehető legjobban szeparálja egymástól a különböző osztályokat. Ez által a 320 dimenzióból először PCA-val 35 dimenzióra csökken az adathalmaz, majd az FDA-val 5 dimenziósra. Az így kapott 5 változója egy-egy adatnak a lehető legjobban reprezentálja a valós LGEI-eket, valamint a lehető legjobban el is különíti az egyes személyeket egymástól. Az MLP tehát ezt az 5 dimenziós vektort kapja bemenetnek, majd 6 rejtett neuronnal számítja ki az n dimenziós eredményt.

4.6.3. Paraméter optimalizálás

A paraméterek hangolása egy rajintelligencia (Swarm Intelligence², SI) algoritmussal, az Artificial Bee Colony (ABC) Algorithm-mel [16, 17] történt. Az ABC algoritmus az élelmet begyűjtő és felderítő méheket utánözva próbál optimális megoldásokat találni. 2005-ban mutatta be Karaboga. A lépései az alábbiak:

1. Populáció inicializálása
2. Ismétlés
3. A dolgozó méhek elhelyezése az élelem forrásuknál
4. A nézelődő méhek kihelyezése az élelem forrásokhoz azok értékei szerint
5. Felderítő méhek küldése a keresési területre új élelem források felfedezésére
6. Az eddigi legjobb források helyének megjegyzése

¹http://en.wikipedia.org/wiki/Linear_discriminant_analysis

²http://en.wikipedia.org/wiki/Swarm_Intelligence

7. mindaddig amíg a feltétel teljesül

Ez alapján implementáltam egy saját megoldást az algoritmusra. Ebben összesen 10 dolgozó méh és 8 + 6 nézelődő méh próbálja felfedezni a paraméterteret. A tíz dolgozó méh közül a legjobb kettőt nevezzük elit méhnek, ezen elit méhekhez csatlakozik négy-négy nézelődő, akik az elit méhek paramétereinek környezetében próbálnak új jobb megoldást találni. A két legjobb találat után következő három találatot elért méhet nevezzük jó méheknek. A jó méhekkel további két-két méh csatlakozik az nézelődők közül, akik ismételtén a jók paramétereinek közelében keresik az optimális megoldást. A maradék öt dolgozó méh pedig véletlenszerűen újra kap paramétereket (5-ös pont), ez által próbálja az algoritmusom biztosítani, hogy ne ragadjon be lokális minimumokba a keresés. Az így kapott új 8 + 6 + 5 és már korábbi 2 + 3 dolgozó méhek által megjegyzett találat közül pedig megtartjuk a tíz lejobbát, és iterálunk tovább így, amíg a leállási feltételt el nem érjük.

Az így kapott optimalizáló algoritmust az MLP-re és a CNN-re is lefuttattam. Az élelem forrás értéke a 6 valós környezetet bemutató szekvenciák eredményeinek átlaga, erre próbál meg az algoritmus minél jobb eredményt keresni. MLP esetében a paraméterteret a PCA és FDA komponensek száma, a rejtett réteg neuronjainak száma, valamint az epochok száma adja. Ezen paraméterek felvehető értékei sorban 1-100, 1-20, 1-100, 50-150 közötti értékek voltak. A CNN esetén a paraméterek az első és második konvolúciós réteg kerneljének mérete, szintén e két réteg jellemző tereinek száma, valamint az utolsó előtti réteg neuronjainak száma, valamint az epochok száma. Ezek felvehető értékei sorban: 1-9, 1-7, 1-10, 1-20, 1-100, 50-150.

Szekvencia	PCA	FDA	REJTETT NEURON	EPOCH	EREDMÉNY
tél4	50	11	3	82	100
tél6	14	7	22	84	88
tavaszb	13	18	5	67	96
tavaszb8	36	6	8	95	66
nyár5	15	10	20	80	100
nyár6	30	10	22	100	99
átlagolt	35	5	6	97	85

4.1. táblázat. Egyéni és átlagolt optimalizálás eredményei paraméterekkel többrétegű perceptronhoz.

Az algoritmust külön-külön lefuttatva egyes tesztesetekre mind más paraméterezést adnak eredményül, de az adott szekvenciára jobb eredményt ad, mint manuális paraméterválasztás esetén el tudtam érni. Az átlagolt keresés eredményei viszont összesen néhány százalékkal térnek csak el az egyéni optimalizált értékektől, ahogyan ezt a 4.2 tábla mutatja. A végleges eredmény MLP esetén: 35 PCA, 5 FDA komponens, 6 rejtett rétegi neuron valamint 97 epoch. CNN esetén: konvolúciós ablakok mérete 3 és 7, jellemző terek száma 5 és 9, 98 neuron a rejtett rétegben és 72 epoch a tanításra.

Szekvencia	manuális	átlagolt	egyéni
tél4	99	98	100
tél6	77	90	88
tavaszb6	81	95	96
tavaszb8	80	66	66
nyár5	94	95	100
nyár6	72	95	99

4.2. táblázat. *ABC-vel optimalizált egyéni és átlagolt, valamint manuális paraméterválasztásos eredmények MLP esetén.*

4.6.4. CNN és MLP együtt

Mint ahogy [5] is említi, az eredmények növelhetők a konvolúciós hálók és többrétegű perceptronok együttesével. Az MLP és a CNN közös eredménye a két háló pozitív találatainak uniója, azaz ha bármely háló pozitív eredményt ad, azt a mintát elfogadjuk. Ennek előnye, hogy amennyiben tévesen negatív értékeket adna az egyik háló, úgy a másik azt kompenzálhatja, de ugyan így igaz a tévesen pozitív találatokra is, amely értékek szintén növekedhetnek. Ez a felismerési arány javulás jól látható a LiDAR sorozatok tesztelése közben is a 5.4. táblázatban, a közös eredmény több százalékkal is meghaladhatja a külön-külön elérteteket.

5. fejezet

Eredménykiértékelés, további módosítások

5.1. Módszerek összehasonlítása

Összesen 10 különböző LiDAR mérési sorozatunk van, amelyből hat készült az összes módszer kiértékelése előtt, 4 pedig utána az adatbázisban szereplő személyek számának növelése céljából. Az összehasonlításhoz használt hat szekvencia közül 2 téli (tél4, tél6), 2 tavaszi (tavasz6, tavasz8), 2 nyári (nyár6, nyár5). Minden tesztalany természetes járása közben került felvételre, a méréseken 4, 5, 6, és 8 ember együttes mozgása látható, akik egymás pályáját szabadon keresztezhetik. A tesztek nevében található szám jelöli a tesztalanyok számát.

Szekvencia	SP-DTW	LGEI+VC	DGHEI	CGCI	LGEI+NN
<i>tél4</i>	0.96	0.98	0.97	0.36	1.00
<i>tél8</i>	0.33	0.83	0.89	0.27	0.85
<i>tavasz6</i>	0.64	0.94	0.81	0.32	0.95
<i>tavasz8</i>	0.33	0.57	0.59	0.20	0.70
<i>nyár5</i>	0.39	1.00	0.59	0.40	0.99
<i>nyár6</i>	0.33	0.67	0.83	0.29	0.79
<i>Átlag</i>	0.50	0.83	0.78	0.31	0.88

5.1. táblázat. Az öt módszer kiértékelésének eredményei: helyes újrafelismerések arányai.

A táblázatban látható, hogy a sziluett lenyomat módszer a *tél4* szekvencián a MoBo adatbázis 100 százalékos eredményét közelíti, ám a többi, nagyobb létszámú, így gyakoribb kitarakásokat bemutató szekvenciákon igen rossz eredményeket ért el. A *tél4* kimagasló eredménye annak köszönhető, hogy ezen sorozatban található sziluettek között nem fordulnak elő lyukas, fél embert ábrázoló sziluettek mind teljesek és a sziluettek kontúrja sem zajos. A SP-DTW eredményeknél is rosszabb a pontfelhő alapú CGCI, amely eredményeiből megállapíthatjuk, hogy a Kinecttől jóval ritkább pontfelhőket generáló LiDAR méréseken nem használható jól ez a megközelítés. Második helyen a leírók közül a DGHEI áll, egyedül a két GEI alapú módszer előzte meg azt. Ehhez a módszerhez a jó minősé-

Módszer	Tanító halmaz generálása	NN tanító fázisa	100 teszt minta felismerése
SP+DTW	8.43	–	43.7
LGEI+VC	142.7	–	0.39
DGHEI	110.3	–	0.98
CGCI	108.9	–	0.26
LGEI+NN	142.7	46.9	0.98

5.2. táblázat. *Kiszámítási ideje (másodpercekben) az egyes fázisoknak a különböző módszerekre a LiDAR alapú adatbázison. NN - neurális háló, VC - vektor összehasonlítás.*

gű mélységkép nélkülözhetetlen, a mi személyenként 200-400 pontból álló felhőinkből már egy tiszta képet tudtunk előállítani, így igen jó eredményt érhattünk el. További javulást eredményezhetnek a jobb minőségű mélységképek, nagyobb sűrűségű pontfelhők. A robusztusságának is köszönhetően a GEI került ki győztesként a módszerek közül, annak is a neurális hálóval módosított változata, amely két szekvencián (*tavaszd*, *nyár6* is lényegesen jobb eredményt ért el a vektor összehasonlítás alapjánál. Ezen kettő a feltüntetett 6 közül a legzajosabb, legtöbb kitakarást tartalmazó szekvencia, ez pedig arra a következtetésre ad okot, hogy a neurális hálók egy robusztusabb osztályozót képeznek az ilyen nehezebb, rosszabb körülmények között felvett esetekben.

Mivel a GEI alapú módszerek voltak a legeredményesebbek, így azokkal teszteltem egy további javulást eredményező szűrő eljárást, amely a 5.2 fejezetben.

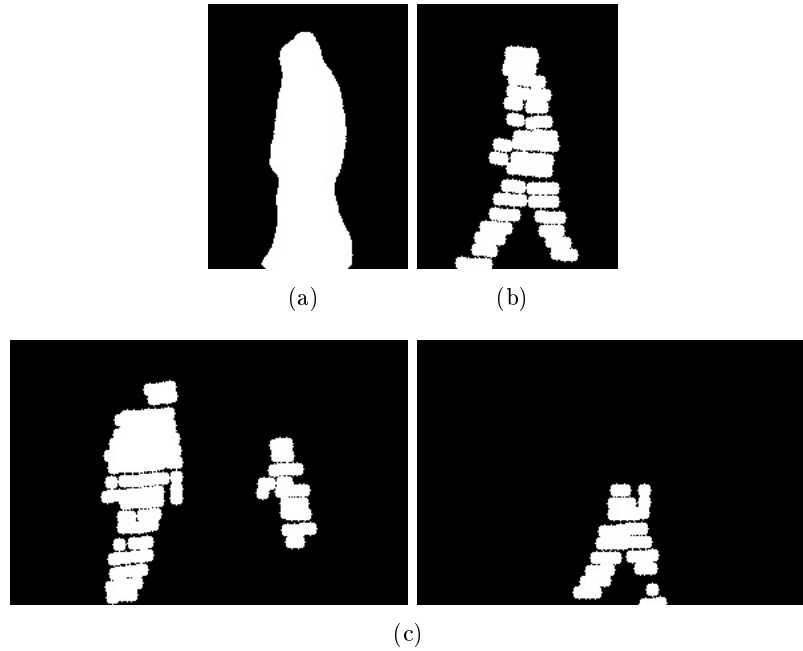
5.1.1. Módszerek időigényessége

Az alábbi táblázat mutatja az egyes módszerek kiszámításának idejét. Habár az SP+DTW tanítási ideje gyorsabb a többiétől a tesztelésnél szintén jelentős különbségek láthatók, mivel az legalább 50-szer lassabb a többiétől a DTW algoritmusnak köszönhetően. Az LGEI+NN módszer jelentős időt igényel a neurális hálók tanításához, ám a felismerési szakasz továbbra is igen gyors, kevesebb mint 0.01 másodperc tesztenként.

5.2. Eredmények javítása szűréssel, forgatással

A LiDAR által felvett mérési sorozatban, mint már említettem a 2. fejezetben, a járókelők kitakarhatják egymást, ez által nem egy teljes alak pontfelhőjével történik a további számítás. Az ilyen esetekben kerül elő több olyan sziluett kép is, amelyen félbevágott embereket, lemaradt lábakat láthatunk. Ezek a képek a felismerés eredményét torzítják, így kiszűrésük fontos feladat.

Egy nagyon egyszerű, ám hatásos módszert vezettem be erre. A tanítóhalmaz összes bináris sziluett képét összegezve egy maszkot kaptam. Ezt a maszkot alkalmaztam az adatok szűrésénél. Minden egyes tanító- és tesztképre a maszk alatti és maszkon kívüli 1-es értékű pixelt szummázva, majd ezen értékek szerint szűrve került egy kép ki az adathalmazból. Amennyiben a maszk méretének 30 százalékánál nagyobb terület esett kívül a maszkon, vagy a maszk alatti terület volt kisebb, mint a maszk területének 40 százaléka,



5.1. ábra. *Nyár5 szekvencia (a) szűrés maskja, és a 0-ás alanyának (b) egy meghagyott és (b) két kiszűrt sziluettje.*

úgy kiszűrésre került a minta. Ezzel több százalékos javulást is eredményezett egyes mintasorozatokon. Természetesen azon adathalmazokon, amelyeken kevés rossz minőségű sziluett volt, nem nagy eredmény javulást eredményezett. Átlagosan egy tesztalany sziluettjeinek 13 százaléka került kiszűrésre.

Fontos még megjegyeznünk, hogy a 4.1. fejezetben bemutatott forgatás az összes vizsgált módszerhez szükséges, mivel azok mind oldalnézeti sziluettekből indulnak ki. A forgatás hiánya, mint látható a 5.3. táblázatban, jelentős teljesítmény romlást eredményez, de látható, hogy annak hiányában is jó eredményeket tudunk elérni. A fentebb bemutatott eredmények kiértékelése során a trajektóriára merőleges vetítés minden módszernél fel lett használva. A szűrés és forgatás által okozott eredmény javulást a 5.3. táblázat összegzi.

Szekvencia		T	SZ	SZ T
tél6	78	85	81	95
tavaszb6	80	95	81	98
nyár5	99	99	100	100
nyár6	75	79	83	95

5.3. táblázat. *Adathalmaz szűrése, valamint a trajektóriára történő forgatás eredményei négy szekvencián. SZ: szűrt, T: trajektóriára forgatott*

A *tél6*, *tavaszb6* és *nyár6* mérésnél jól láthatóan sok százalékos eredmény javulás látható a szűrés és forgatás eredményeképp. A *nyár5* sorozaton viszont ezzel ellentétben jelentős javulást nem eredményez az eljárás, ennek oka az lehet, hogy a *nyár5* minőségében talán legjobb szekvencia az itt vizsgáltak közül, így az adathalmaz szűrése szinte csak a minták

elvesztéséhez vezetett.

5.3. LGEI részletes eredményei

Az alábbi táblázatokban láthatóak a CNN, MLP és a kettő együttesének eredményei különböző mintasorozatokon szűrt és szűretlen képeken. Az ábrákon $n \times 200$ tesztminta eredményeinek átlaga látható. Az egyes sorok a tesztalanyokat reprezentálják, míg az oszlopok a háló egyes kimeneteit. Piros szín jelöli a negatív, zöld a pozitív értékeket. Minél nagyobb egy négyzet, annál nagyobb az abszolút értéke. A várt érték a főátlóban 1, míg minden más helyen -1.

A tesztelés során, mint már említettem minden alanyról 200 minta kerül vizsgálatra. Ezek egymás után egyenletesen elosztott kezdőértékekről indított 100 sziluett átlagolásából nyert LGEI tesztek. A felismerés "esélyegyenlősége" miatt mindenkiről ugyan annyi tesztminta készül, még akkor is, ha 700 vagy 200 mintából kell a 200 LGEI-t elkészíteni. Minden felhasznált mérőszorozatban van egy pont, ahol a tesztalanyok egy időre eltűnnek, majd újra megjelennek. Ezen pont előtti adatokból lettek tanítva a háló, az utána lévő mintákból pedig tesztelve.

5.3.1. Eredménykiértékelés

Mind a CNN, mind az MLP n kimenettel rendelkezik, ahol n a megkülönböztetni kívánt személyek száma. Az egyes tesztminták alapján tehát egy n elemű vektor a kimenet, ahol az értékek $[-1,1]$ intervallumról várhatóak. Helyes felismerésnek tekinthető, ha a vektorban a tesztmintához tartozó indexű elem a legnagyobb pozitív érték. Ezek alapján tehát engedélyezett a több pozitív elem, ezek közül mindig a maximális az, amely meghatározza a felismerés sikerességét. Az így számított helyes felismerések aránya az összes tesztek számához látható a 5.3, 5.4 és 5.5 táblákban.

5.3.2. Egyes sorozatok eredményei

Egyes mintákon (pl. *tél6*, *tavas8*) látható tévedések és gyengébb teljesítmény a kevés mintának tudható be. A *tavas8* mintahalmazban például az 5-ös számú tesztalanyról kevesebb, mint 90 sziluett maradt az adatszűrés után. Ennek következtében a *tavas8* szekvencia tesztelése során nem 100 egymást követő adatból lett előállítva egy LGEI.

A magyar módszerrel történő párosítások a *tavas8* és *tél6* kivételével minden szekvencián megfelelő eredményt adtak. A *tél6* sorozaton az CNN-nek nem sikerült felismernie a 6-os tesztalanyt, helyette tévesen a 2-esnek vélte, így ezzel hibázott, a *tavas8* sorozaton pedig az MLP-nek nem sikerült helyes felismerést végezni az 5-ös és 8-as tesztalanyoknál. A két sorozaton a másik háló viszont helyes eredményeket adott, bebizonyítva tehát, hogy a két háló együttese jobb felismerési arányt képes elérni.

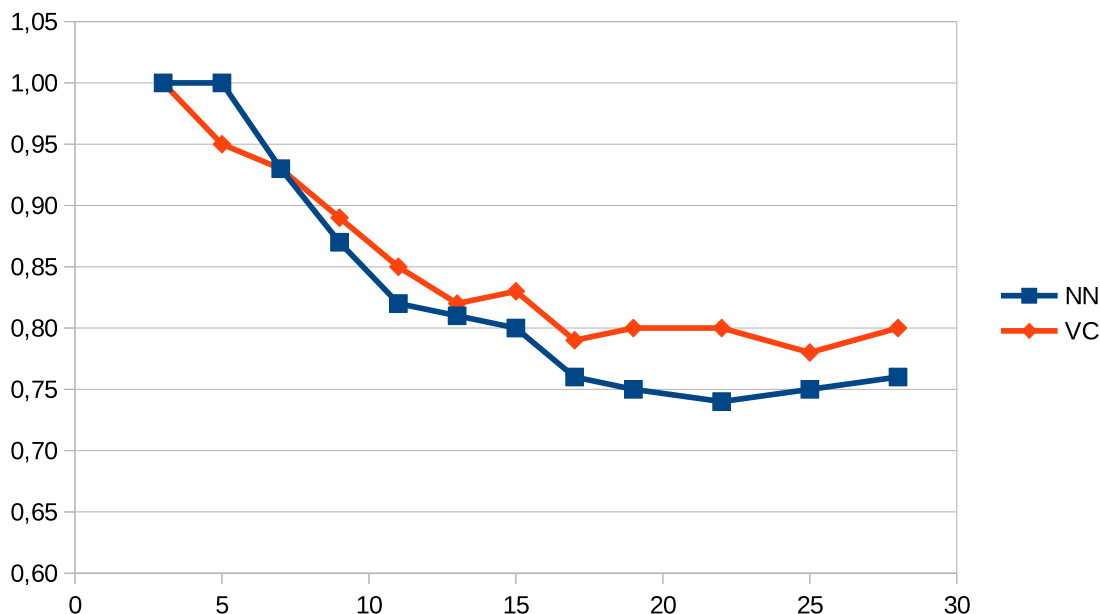
Szekvencia	MLP	CNN	CNN-MPL	MDA
tél4 T	98	94	100	98
tél4 SZ T	98	94	99	100
tél6	68	72	78	74
tél6 T	81	83	85	83
tél6 SZ	77	76	81	80
tél6 SZ T	90	85	95	88
tavaszi6	75	70	80	84
tavaszi6 T	90	90	95	94
tavaszi6 SZ	70	69	81	87
tavaszi6 SZ T	95	91	98	97
tavaszi8 T	48	66	70	57
tavaszi8 SZ T	66	63	70	70
nyár5	96	86	99	96
nyár5 T	97	98	99	97
nyár5 SZ	100	84	100	100
nyár5 SZ T	95	99	100	93
nyár6	74	57	75	80
nyár6 T	72	78	79	67
nyár6 SZ	83	70	83	86
nyár6 SZ T	95	77	95	82
járás2 T	94	89	95	95
járás2 SZ T	99	91	100	100
átlag	84	81	89	87

5.4. táblázat. *Konvolúciós háló és többrétegű perceptron eredményei egyes szekvenciákon. SZ: szűrt, T: trajektóriára forgatott. MLP: a többrétegű perceptron eredményei, CNN: a konvolúciós háló eredményei, CNN-MPL: az MLP és a CNN együttesének eredményei, MDA: a [9] által bemutatott módszer eredményei.*

5.3.3. Összesített eredmények

Az összesen 10 felvételen 28 személy jelent meg. Egyesek akár mindegyik mérésen, mások kevesebben, többen csak egy alkalommal voltak tesztalanyok. Az így nyert 28 emberről lehetnek téli, nyári, tavaszi felvételek, ezeken a kabát például befolyásolhatja egy LGEI alakját, így minden emberről a róla legtöbb mintát tartalmazó sorozatot választottam, nem pedig összefésültem az összes képét, és ez által hoztam létre egy 28 emberből álló egységes összesített adatbázist. Az így képzett tesztekben az alább látható eredmények születtek. A táblázat és a diagram által bemutatott eredményeken mindig a legtöbb tanító mintával rendelkező személlyel növeltem a tesztalanyok számát. Az itt bemutatott eredmények mind szűrt és forgatott adathalmazokon lettek kiértékelve.

Habár az LGEI+NN módszer igen jó eredményeket adott az egyes szekvenciákon, teljesítménye felülmúlta az LGEI+VC módszerét, az itt bemutatott eredmények alapján nem jelenthető ki egyértelműen annak előnye. A nehézkes paraméter választása a neurális hálóknak annak teljesítményének rovására megy, így a vektor összehasonlítás lehet mégis jobb választás a hálók helyett. A teljes adatbázison történő tesztek során szintén magas felismerési arányok láthatók. A tesztalanyok számának növekedésével eleinte folyamatos



5.2. ábra. A teljes 28 fős adatbázison végzett kiértékelések eredményei. **NN** - neurális háló, **VC** - vektor összehasonlítás.

n	NN	VC	n	NN	VC
3	1.00	1.00	15	0.80	0.83
5	1.00	0.95	17	0.76	0.79
7	0.93	0.93	19	0.75	0.80
9	0.87	0.89	22	0.74	0.80
11	0.82	0.85	25	0.75	0.78
13	0.81	0.82	28	0.76	0.80

5.5. táblázat. Kiértékelés eredményei a korábbi (VC, [10]) és az általam javasolt (NN) módszeren: helyes újrafelismerési arányok. *n* - személyek száma, **NN** - neurális háló, **VC** - vektor összehasonlítás.

eredményromlás tapasztalható, azonban a 17. személy bevonásával és az az utáni tesztek során az eredmény 80 százalék körüli szinten stagnál.

A vektor összehasonlítást használó osztályozás mellett szól még a számítási komplexitása is. A GEI képek generálásán felül a neurális hálók tanítása jelentős időt vesz igénybe, amely egy online tanuló rendszer esetén problémás lehet, így ott a vektor összehasonlítás jobb, ám felmerül a kérdés, hogy egy új személy megjelenésekor hova sorolódik? Egy távolság küszöb felállításával ez a probléma kiküszöbölhető, ám ezen paraméter beállítása is egy nehéz feladat és az eddig bemutatott eredményeken akár ronthat is a küszöb használata. Valós környezetben történő alkalmazás esetében ha lehetőség van egy már elkészített, az itt bemutatott 28 fős adatbázisnál nagyobb, több száz, akár több ezer fős adatbázis elérésére, úgy a neurális hálózatok is jobb eredményt érhetnek el, tekintve hogy a mély tanulás egy fontos alapja a nagy adathalmaz.

Az eredmények alapján kijelenthetjük, hogy a LiDAR alapú személykövetés és -azonosítás

egy valós időben kivitelezhető feladat olyan környezetben, ahol 8-10 személy együttesen jelen van. Magasabb számú járókelő esetén a kitakarások olyan méretekig nőhetnek, hogy a felismerés eredményeket túlságosan lerontják. Ennek ellenére offline felismerés továbbra is lehetséges különböző szekvenciákról összegzett minták alapján.

6. fejezet

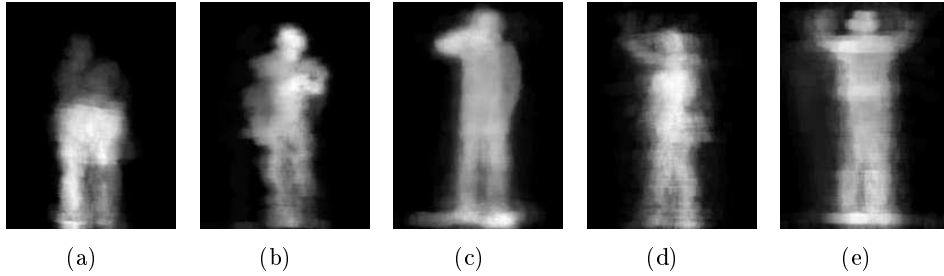
Események detektálása

6.1. Kiválasztott jellemzők

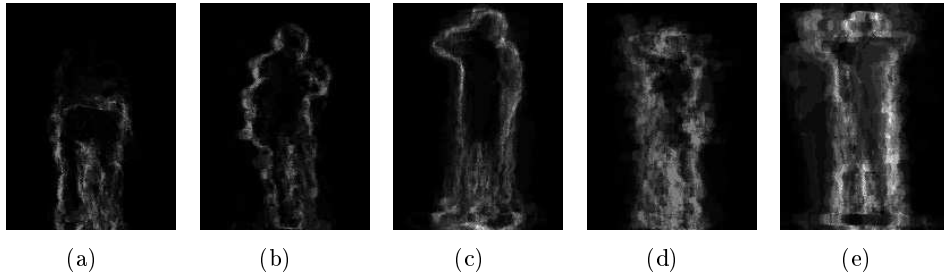
Az események felismerésére fejlesztett módszerek nagy része pontfelhőket vagy mélységképeket alkalmaz, amelyeken foglaltsági mintákat vagy egyéb előre megtervezett jellemzőket nyernek ki [23, 26, 27, 35]. A Histogram of Oriented Principal Components (HOPC [27]) és a 3D Motion Flow [23] ötleten alapuló eljárások tesztelése közben a járásanalízisnél kipróbált CGCI-hez (4.5. fejezet [33]) hasonlóan rossz eredményeket kaptam, ezzel is bizonyítva a feltételezést, hogy ilyen ritka, személyenkénti 200-400 pontból álló felhőkön ezek az eljárások nem tudnak olyan teljesítményt nyújtani mint a nagyságrendekkel sűrűbb Kinectes méréseken. Ezen kísérletek során kiderült, hogy egy robusztusabb jellemzőre van szükség, és a tesztek kimutatták, hogy a GEI-szerű sziluett átlagoló módszerek biztosítják a legmagasabb felismerési arányokat.

Két képi jellemző elkészítése történt. Előnézeti mélységi és bináris képeket generáltam a pontfelhőkből minden egyes képkockára, majd átlagoltam a mélységképeket egymást követő 30-40 képkockán, amely jellemzően az egyes cselekvések hosszával megegyező időtartam volt, ez által jött létre az egyik jellemző, az előnézeti átlagolt mélységi kép (frontal averaged depth map, FADM). A mélységi képek átlagolásával a cselekvés közbeni 3D információkat is ki lehet nyerni, amelyek segítik a komplex cselekvések felismerését, ezt a plusz információt (pl. előre történő hajolás, nem pedig guggolás) bináris képek átlagolásával nem lehet kihasználni. A második jellemző kép a bináris sziluettekéből készült. Két egymást követő kép XOR műveletéből készült képeken történt átlagolás, így jön létre az átlagolt XOR kép (averaged XOR, AXOR), amelyen minél világosabb egy pixel annál többször történt változás az adott pozícióban. Az AXOR kép dinamikai információkat tartalmaz az egyes cselekvésekről. A járásanalízis során a végtagok mozgása periodikus és szimmetrikus, így nem veszítünk információt az átlagolás során úgy, mint tennénk azt egy aperiodikus, aszimmetrikus mozgás, bármilyen cselekvés (pl. integetés) során. Ehhez szükséges a dinamikai információk kinyerésére, amelyre az AXOR képünk képes is. E két jellemző, az FADM és az AXOR kép együttesen képes leírni a mozgási és alaki jellemzőit az egyes cselekvéseknek, így ezek kombinációja, együttes használata megfelelően képes leírni a mozgásokat. Az FADM képek és az AXOR-ok négyzetei láthatóak a 6.1. és 6.2. ábrákon.

Minden cselekvésre (pl. integetés, órára pillantás) három konvolúciós neurális háló lett betanítva. A hálók bemenetei a lecsökkentett méretű (20×16 pixel) normalizált FADM-ek, AXOR-ok, valamint az AXOR képek négyzetei, amelyek még jobban kiemelik az AXOR képen látható változásokat, illetve a négyzetre emelés zajszűrési funkciókat is ellát. A kimenetek címkézése egyszerű: a betanított cselekvésre 1, minden másra -1 (pl. egyszerű természetes járásra is). Az egyes jellemző képek egyenkénti teszteléséből kiderült, hogy ezek kiegészítik egymást, egyik se volt minden cselekvésnél jobb a másiknál. A CNN-ek kimenetei -1.0 és 1.0 közötti értékek, egy bemeneti X képre a felismeréshez történő elfogadási teszt a következő: $FADM(X) > 0.6$ és $AXOR(X) > 0.6$ és $AXOR^2(X) > 0.6$. Ahogyan a járás felismerésnél használt konvolúciós hálók, úgy ezek is nagyságrendekkel kevesebb paramétert használnak, mint a (pl. ImageNet LSVRC) versenyeken használt társaik [19], így kisebb számú adaton is jól tudnak dolgozni.



6.1. ábra. *FADM képek a (a) lehajlás, (b) órára pillantás, (c) telefonálás, (d) integetés és (e) kétkezes integetés cselekvésekhez.*



6.2. ábra. *AXOR² képek a (a) lehajlás, (b) órára pillantás, (c) telefonálás, (d) integetés és (e) kétkezes integetés cselekvésekhez.*

6.2. Tesztek, eredmények

A cselekvések felismeréséhez 4 mérési sorozatot készítettünk, kültériket és beltériket egyaránt, amelyeken 1, 3, 4 személy egy időben sétálva cselekvéseket végez.

6.2.1. Eredmények

Az eredmények a 6.1. táblában találhatóak. Láthatjuk, hogy a lehajlás és kétkezes integetés eseményeket minden alkalommal sikerült felismerni, habár a kétkezes integetés több

alkalommal is detektálva lett az egykezes integetés során is és a mutató többször lett más cselekvés esetén jelezve mint a való mutató alkalmával. Ez utóbbi két okból lehetséges: i) a valamire mutató összekeverhető akár egy olyan integetéssel ahol a kart kinyújtva csak tenyér forgatásával történik az integetés - ilyenre van példa az adatbázisban ii) a mutatóra van a legkevesebb minta az adatok között. Annak ellenére, hogy az adathalmazunk kis méretű, valamint az események felismerésében voltak tévedések, a járólók körkörös sétálása során nem történt jelzés csak tényleges tevékenységek közben, így a cselekvés jelzés sikeresnek tekinthető. Több LiDAR felvétel javulást okozhat az eredményekben, valamint pontosítható az események összekeverésében.

	<i>Lehajolás</i>	<i>Karóra</i>	<i>Telefon</i>	<i>Integetés</i>	<i>Integetés2</i>	<i>Mutató</i>
<i>Lehajolás</i>	13					
<i>Karóra</i>		8		1	2	1
<i>Telefon</i>			4		1	
<i>Integetés</i>				11	1	1
<i>Integetés2</i>				4	5	1
<i>Mutató</i>				2	2	1
<i>Eredeti</i>	13	9	6	14	5	2

6.1. táblázat. Az összetévesztési mátrix, az oszlopok a való cselekvéseket, a sorok pedig a megfigyeléseket jelzik.

7. fejezet

Értékelés

7.1. Személyazonosítás

A személyazonosítási funkciót a járásfelismerés az eredmények alapján jól teljesíti, 28 fős adatbázison elért 80 százalék egy jó eredménynek számít, míg az 5-6 fős szekvenciákon elért felismerési arányok még ennél is magasabbak. Mindez alapján kijelenthető, hogy a járás, mint személyazonosításhoz használt jellemző LiDAR felvételeken kültéri realisztikus környezetben egy működőképes koncepció.

7.2. Cselekvés felismerés

A cselekvés felismerés LiDAR környezetben lényegesen nehezebb a szakirodalomban található körülményekhez képest, aholis több ezer pontból álló felhőkön történik az osztályozás, felismerés, míg az általunk előidézett realisztikus környezetben ennél lényegesen ritkább pontfelhőkkel dolgozunk. Ilyen megfontolások mellett a tevékenységek jelzése sikeresnek mondható, mivel nem történt cselekvés jelzés, amikor nem történt semmilyen esemény, bár a rendszer még további javításokra és módosításokra szorul. Az esetek többségében mégis sikeresen felismerésre került az adott cselekvés, igaz néha hozzá hasonlókkal együtt.

Köszönetnyilvánítás

Dolgozatom készítése során kapott támogatásért hálás köszönet és elismerés illeti az MTA SZTAKI Elosztott Események Elemzése kutatólaboratóriumát, külön név szerint is Dr. Benedek Csaba konzulensemét a kollegiális és szakmai segítségért, valamint Dr. Szirányi Tamást, a kutatócsoport vezetőjét, hogy hozzáférést biztosított az intézmény informatikai és tudományos infrastruktúrájához az elmúlt év során.

Ábrák jegyzéke

1.1.	Modell alapú járás elemzés [21].	8
1.2.	Bináris sziluett minták (a) CMU MoBo [8] adatbázisból és a (b) LiDAR adatbázisból.	9
1.3.	Cselekvések csontvázalattal történő reprezentációja [25].	10
1.4.	(a) Egy foglaltsági minta módszer, a random occupancy patterns [1], valamint (b) a 3D motion flow cella felosztása [23].	11
2.1.	Az általunk is használt Velodyne HDL-64E.	12
2.2.	LiDAR használati módjai. a) jármű navigációra használt, nem döntött. b) döntött, video-megfigyelési használat.	13
2.3.	A LiDAR használata közbeni pozíciója egy az általa készített pontfelhőben feltüntetve.	13
2.4.	A keretrendszer folyamatábrája. Folytonos kék vonal a rövid távú modul sikeres, szaggatott a sikertelen követése. Folytonos narancs vonal a hosszú távú modul sikeres, szaggatott a sikertelen követése. Ti: inaktívan töltött idő, Ta: archiválási időkülönb. Az általam fejlesztett új, neurális háló alapú hosszú távú követő modul cseréli le a régi intenzitás és magasság alapú módszert.	15
2.5.	A nyár5 szekvencia pontfelhő-sorozatának (a) egy pillanatnyi képe és a hozzá tartozó (b) trajektóriák. Minden egyes szereplőhöz más-más színkód tartozik, ezzel a színkóddal a trajektóriákat be tudjuk azonosítani. A trajektóriák egy felülnézeti pontból láthatók.	16
4.1.	A trajektóriára történő forgatás síkja oldal- és felülnézetből.	20
4.2.	Sziluett vetítés a trajektória érintőjével párhuzamos síkra	21
4.3.	Sziluett vetítés egy a személy "középpontjára" eső LiDAR sugárra merőleges síkra.	21
4.4.	MoBo mintasorozat 1-es alanyának tanító és teszt sziluett lenyomata.	22
4.5.	MoBo mintasorozat 2-es alanyának tanító és teszt sziluett lenyomata.	22
4.6.	Nyár5 mintasorozat 0-ás alanyának tanító és teszt sziluett lenyomata.	23
4.7.	Nyár5 mintasorozat 2-es alanyának tanító és teszt sziluett lenyomata.	23
4.8.	Nyár5 5 tesztalanyának sziluettjei és hozzájuk tartozó LGEI-k.	25
4.9.	A <i>nyár5</i> szekvencia egy tesztalanyának (a) mélységképe és (b) DGHEI képe [12].	26
4.10.	A <i>nyár5</i> szekvencia egy tesztalanya CGCI [33] képének komponense. (a) és (b) a görbületi jellemzőkből [34] készített kép, (c) a pontfelhő sűrűsége.	26
4.11.	A megvalósított CNN architektúra.	27

5.1.	<i>Nyár5</i> szekvencia (a) szűrés maszkja, és a 0-ás alanyának (b) egy meghagyott és (b) két kiszűrt sziluettje.	33
5.2.	A teljes 28 fős adatbázison végzett kiértékelések eredményei. NN - neurális háló, VC - vektor összehasonlítás.	36
6.1.	<i>FADM</i> képek a (a) lehajlás, (b) órára pillantás, (c) telefonálás, (d) integetés és (e) kétkezes integetés cselekvésekhez.	39
6.2.	<i>AXOR²</i> képek a (a) lehajlás, (b) órára pillantás, (c) telefonálás, (d) integetés és (e) kétkezes integetés cselekvésekhez.	39

Táblázatok jegyzéke

4.1. Egyéni és átlagolt optimalizálás eredményei paraméterekkel többrétegű perceptronhoz.	29
4.2. ABC-vel optimalizált egyéni és átlagolt, valamint manuális paraméterválasztásos eredmények MLP esetén.	30
5.1. Az öt módszer kiértékelésének eredményei: helyes újrafelismerések arányai. .	31
5.2. Kiszámítási ideje (<i>másodpercekben</i>) az egyes fázisoknak a különböző módszerekre a LiDAR alapú adatbázison. NN - neurális háló, VC - vektor összehasonlítás.	32
5.3. Adathalmaz szűrése, valamint a trajektóriára történő forgatás eredményei négy szekvencián. SZ : szűrt, T : trajektóriára forgatott	33
5.4. Konvolúciós háló és többrétegű perceptron eredményei egyes szekvenciákon. SZ : szűrt, T : trajektóriára forgatott. MLP : a többrétegű perceptron eredményei, CNN : a konvolúciós háló eredményei, CNN-MLP : az MLP és a CNN együttesének eredményei, MDA : a [9] által bemutatott módszer eredményei.	35
5.5. Kiértékelés eredményei a korábbi (VC , [10]) és az általam javasolt (NN) módszeren: helyes újrafelismerési arányok. n - személyek száma, NN - neurális háló, VC - vektor összehasonlítás.	36
6.1. Az összetévesztési mátrix, az oszlopok a valós cselekvéseket, a sorok pedig a megfigyeléseket jelzik.	40

Irodalomjegyzék

- [1] *Robust 3D Action Recognition with Random Occupancy Patterns*. Springer, October 2012.
- [2] C. Benedek. 3D people surveillance on range data sequences of a rotating Lidar. *Pattern Recognition Letters*, 50:149–158, 2014. Special Issue on Depth Image Analysis.
- [3] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.
- [4] Mark S. Nixon Chew-Yean Yam. *Model-based Gait Recognition*.
- [5] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1918–1921, July 2011.
- [6] D.C. Cireşan, U. Meier, L.M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1135–1139, Sept 2011.
- [7] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119 – 130, 1988.
- [8] Ralph Gross and Jianbo Shi. The CMU Motion of Body (MoBo) Database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Pittsburgh, PA, June 2001.
- [9] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(2):316–322, Feb 2006.
- [10] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2):316–322, Feb 2006.
- [11] M. Hofmann, S. Bachmann, and G. Rigoll. 2.5D gait biometrics using the depth gradient histogram energy image. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 399–403, Sept 2012.
- [12] M. Hofmann, S. Bachmann, and G. Rigoll. 2.5D gait biometrics using the depth gradient histogram energy image. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 399–403, Sept 2012.

- [13] Goodfellow I. J., D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *ICML*, 2013.
- [14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, Jan 2013.
- [15] A. Kale, N. Cuntoor, B. Yegnanarayana, A.N. Rajagopalan, and R. Chellappa. Gait analysis for human identification. In Josef Kittler and MarkS. Nixon, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*, pages 706–714. Springer Berlin Heidelberg, 2003.
- [16] Dervis Karaboga and Bahriye Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108 – 132, 2009.
- [17] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007.
- [18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [20] B. Kwolek, T. Krzeszowski, A. Michalczuk, and H. Josinski. 3D gait recognition using spatio-temporal motion descriptors. In *Intelligent Information and Database Systems*, volume 8398 of *Lecture Notes in Computer Science*, pages 595–604. Springer International Publishing, 2014.
- [21] Bogdan Kwolek, Tomasz Krzeszowski, Agnieszka Michalczuk, and Henryk Josinski. 3d gait recognition using spatio-temporal motion descriptors. In *Intelligent Information and Database Systems*, volume 8398 of *Lecture Notes in Computer Science*, pages 595–604. Springer International Publishing, 2014.
- [22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.
- [23] Matteo Munaro, Gioia Ballin, Stefano Michieletto, and Emanuele Menegatti. 3d flow estimation for human action recognition from colored point clouds. *Biologically Inspired Cognitive Architectures*, 5:42 – 51, 2013. Extended versions of selected papers from the Third Annual Meeting of the {BICA} Society (BICA 2012).

- [24] M. PAT MURRAY, A. BERNARD DROUGHT, and ROSS C. KORY. Walking patterns of normal men. *The Journal of Bone & Joint Surgery*, 46(2):335–360, 1964.
- [25] F. Ofi, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 8–13, June 2012.
- [26] O. Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 716–723, June 2013.
- [27] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision, ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 742–757. Springer International Publishing, 2014.
- [28] Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '11*, pages 147–156, New York, NY, USA, 2011. ACM.
- [29] Imad Rida, Somaya Al Maadeed, and Ahmed Bouridane. Unsupervised feature selection method for improved human gait recognition. In *European Signal Processing Conference*, Nice, France, 2015.
- [30] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2809–2813, July 2011.
- [31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.
- [32] Ah Chung Tsoi Steve Lawrence, C. Lee Giles and Andrew D. Back. Face recognition: A convolutional neural-network approach. *IEEE Trans. on neural networks*, 8(1):98–113, January 1997.
- [33] J. Tang, J. L., T. Tjahjadi, and Y. Gao. 2.5D multi-view gait recognition based on point cloud registration. *Sensors*, 14(4):6124, 2014.
- [34] P. Tosranon, A. Sanpanich, C. Bunluechokchai, and C. Pintavirooj. Gaussian curvature-based geometric invariance. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*, volume 02, pages 1124–1127, May 2009.

- [35] AntonioW. Vieira, EricksonR. Nascimento, GabrielL. Oliveira, Zicheng Liu, and MarioF.M. Campos. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In Luis Alvarez, Marta Mejail, Luis Gomez, and Julio Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441 of *Lecture Notes in Computer Science*, pages 252–259. Springer Berlin Heidelberg, 2012.
- [36] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.
- [37] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297, June 2012.
- [38] Lu Xia, Chia-Chih Chen, and J.K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27, June 2012.
- [39] Xiaodong Yang and YingLi Tian. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1):2 – 11, 2014. Visual Understanding and Applications with RGB-D Cameras.
- [40] Z. Zhang, M. Hu, and Y. Wang. A survey of advances in biometric gait recognition. In Zhenan Sun, Jianhuang Lai, Xilin Chen, and Tieniu Tan, editors, *Biometric Recognition*, volume 7098 of *Lecture Notes in Computer Science*, pages 150–158. Springer Berlin Heidelberg, 2011.

Függelék

F.1. Tiny CNN könyvtár

A Tiny CNN könyvtár¹ egy header-only C++11 szabványra épülő könyvtár. Benne konvolúciós hálók létrehozására, tanítására használható osztályok vannak implementálva. Lehetőség van konvolúciós, alul-mintavételező, teljesen összekötött rétegek alkalmazására, valamint a dropout is implementálva van benne és többféle aktivációs függvény is választható a neuronokban. Mivel egyedül headerekből álló könyvtár, integrálása egy projektbe meglehetősen egyszerű, ennek ára viszont a sebesség, mivel nem egy GPU-n működő könyvtár.

¹<https://github.com/nyanp/tiny-cnn>