VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Baigiamasis magistro darbas

Nuotraukos animacijos pritaikymas sudėtingoje vaizdo įrašo aplinkoje

(Adapting Photo Animation in Complex Video Environment)

Atliko: 2 informatikos magistro kurso, 1 grupės studentas
Julius Blusevičius (parašas)
Darbo vadovas:
prof. Dr. Aistis Raudys (parašas)

VILNIUS
2021

Turinys

Santrauka

Nuotraukos animacija yra procesas, kuomet nuotraukoje esantis objektas priverčiamas judėti pagal vaizdo įraše esantį tos pačios kategorijos objektą. Šis rezultatas yra pasiekiamas išmokant neuroninį tinklą rekonstruoti vaizdo įrašą, kuomet kaip įvestis yra duodamas pirmas kadras ir vaizdo įrašo judesio reprezentacijos. Moksliniuose darbuose pristatomi neuroninių tinklų modeliai išmokinami naudojantis specialiai surinktomis aibėmis, kuriose objektas yra visą laiką filmuojamas iš to paties kampo ir atlieka paprastus veiksmus. Tai leidžia pasiekti gerus rezultatus, tačiau toks požiūris į nuotraukos animaciją reiškia, kad praktiniai tokio algoritmo pritaikymai yra labai siauri. Šiame darbe bus apžvelgtas procesas, kaip galima panaudoti nuotraukos animacijos algoritmus tam, kad modifikuoti sudėtingą vaizdo medžiagą, kurioje objektai yra skirtingi ir atlieka skirtingus veiksmus, kurie filmuojami iš skirtingų atstumų ir randasi skirtingose kadrų vietose. Sėkmingas algoritmo pritaikymas, galėtų leisti fotorealistiškai modifikuoti vaizdo įrašus, keičiant objektų išvaizdą.

Summary

Image animation is a process in which, image is animated based on motion of another video with an object of the same category. This is achieved by training a neural network to reconstruct video, using the first frame of the video and motion representations. Neural network models presented in research papers are trained using as simple data, as possible, in which an object is always filmed from the same location and is performing simple animations. This allows to achieve photorealistic results, but practical application remains very narrow. In this paper, it will be showed, how image animation can be applied in a complex video environment in which: objects are different in appearance, different motions are performed, object distance between cameras varies and objects are at different locations in frame. Successful image animation application, would allow to photorealistically edit the video by changing an object's appearance.

1. Įvadas

Neuroniniai tinklai jau daugiau nei dešimtmetį yra sėkmingai naudojami klasifikavimo, atpažinimo bei kitom įvairiom problemom spręsti. Galingų kompiuterių bei mokslininkų dėka neuroniniai tinklai sukėlė tikrą revoliuciją dirbtinio intelekto pasaulyje. Vos per dešimtmetį pradėjus juos plačiai taikyti buvo išspręstos problemos, kurios dar prieš 15 metų atrodė visiškai neįveikiamos. Nuotraukų klasifikacija bei objektų aptikimas tai yra užduotys, kuriose įvesties dydis yra milijoniniai pikselių masyvai, ir kurių beveik neįmanoma išspręsti nenaudojant mašininio mokymosi bei neuroninių tinklų. Visoje dirbtinio intelekto istorijoje yra susitelkiama į kažkokios konkrečios problemos sprendimą: ar tai būtų klasifikacija, ar tai būtų išėjimas iš labirinto, ar tai būtų geriausio sprendimo parinkimas kompiuteriniuose žaidimuose.

Labai didelė ir svarbi sąvoka, kuri nusako intelektą yra gebėjimas kurti. Geriausius kūrybinius sugebėjimus demonstruojanti dirbtinio intelekto šaka taipogi yra neuroniniai tinklai. Naudojantis jais galima lengvai generuoti meno kūrinius, iš kelių nuotraukų padaryti vieną jungtinę nuotrauką ir t.t. Tačiau populiariausia bei kontroversiškiausia sritis yra “deepfake” įrašų generacija. “Deepfake” tai yra vaizdinė medžiaga kurioje vaizduojamas žmogus yra pakeičiamas kitu asmeniu naudojantis dirbtiniais neuroniniais tinklais. Šis rezultatas pasiekiamas naudojantis nuotraukos animacijos algoritmais, kurie naudodamiesi nuotraukoje esančia išvaizda, priverčia objektą judėti remiantis vaizdo įrašu su tos pačios kategorijos objektu su skirtinga išvaizda. Šioje srityje kiekvienus metus atsiranda vis naujesni bei geresnius rezultatus sukuriantis algoritmai. Tačiau šie algoritmai yra mokomi bei testuojami su vaizdo įrašais, kurie yra kiek imanoma paprastesni: kamera nejuda ir objektas išlieka toje pačioje pozicijoje.

Šiame darbe nagrinėjamas procesas, kurio metu nuotraukos animavimo algoritmai bus bandomi pritaikyti prie vaizdo medžiagos, kurioje kamera nuolat juda. Objekto pozicija kadre ir atstumas tarp objekto ir kameros pastoviai keičiasi. Sėkmingas nuotraukos animacijos pritaikymas prie sudėtingos vaizdo įrašo aplinkos suteiktų galimybę vaizdo įrašuose realistiškai pakeisti norimą personažą į asmenį, kuris yra atvaizduojamas nuotraukoje.

2. Aktualumas

Nuotraukų animacijos sukurti rezultatai viešoje erdvėje dažniausiai žinomi kaip “Deepfake”.Sugeneruota vaizdo medžiaga gali būti panaudota šmeižimo, klaidingų naujienų ar kitiem realybę iškraipančiam tikslam. Dar palyginti neseniai atsiradusi technologija buvo pritaikyta tam, kad paniekinti politinį personažą. Pavyzdžiui 2019 metais socialiniuose [tinkluose buvo paskleisti vaizdo įrašai, kuriuose Jungtinių valstijų politikės Nancy Pelosi kalbėjimas buvo pakeistas į neryškią kalbą, kuri būdinga apsvaigusiems žmonėms [DO19]. Greitai buvo patvirtinta, kad šie vaizdo įrašai buvo sukurti naudojantis “Deepfake” technologijomis, tačiau didžioji dalis žiniasklaidos savo eteryje tai pristatė kaip realią naujieną.

Kita pritaikymo sritis yra filmų kūrimas. Nors idėja šias technologijas naudoti filmuose yra dar menkai pribrendus, tačiau jau yra filmų kuriuose ši technologija buvo pritaikyta. “Deepfake” technologija buvo panaudota filme “Žvaigždžių karai” kuriame buvo panaudotas jau mirusios aktorės veidas. Kitame “Žvaigdžių karų” filme buvo panaudota aktoriaus kuriam dabar yra 77 metai, jaunystės nuotrauka[RP19].

Tačiau plačiausiai “Deepfake” technologijos yra naudojamos socialiniuose tinkluose. Pvz. programėleje Zao galima aktorių veidą pakeisti į savąjį įvairiuose serialuose bei filmuose[DJ19]. Tokios programos kaip „Snapchat“ leidžia uždėti įvairius filtrus ant realiu laiku filmuojamo veido.

Kadangi ši technologija yra labai nauja, jos panaudojimo galimybės dar nėra pilnai išnagrinėtos. Sėkmingas nuotraukos animacijos panaudojimas sudėtingoje vaizdo įrašo aplinkoje atvertų galimybę filmuose dirbtinai pakeisti vaidinančius aktorius.

3. Darbo tikslas ir uždaviniai

Šio darbo pagrindinis tikslas yra ištirti, kaip nuotraukos animacijos algoritmai gali būti pritaikyti prie sudėtingos vaizdo įrašo aplinkos. Tam, kad pasiekti šį tikslą, buvo iškelti šie uždaviniai:

1. Literatūros apžvalga, kurios metu bus apžvelgti darbai, kuriuose yra pristatomi nuotraukų animacijos algoritmai.
2. Išnagrinėti vaizdo modifikavimo procedūras, kurios leistų supaprastinti vaizdo medžiagą.
3. Ištestuoti autorių pristatytus tinklus su tinklams nematytais duomenimis.

4. Išmokinti ir ištestuoti tinklą su duomenų aibe, kuri automatiškai yra sugeneruota iš kasdieninėje aplinkoje dažniau matomų duomenų, tokių kaip filmas.

4. Galutiniai Rezultatai

Darbo metu tikimasi sėkmingai pritaikyti egzistuojančius algoritmus prie duomenų, kurie skiriasi nuo standartiškai naudojamų aibių mokymams. Visos duomenų aibės, kurios yra naudojamos tinklams mokyti yra sudarytos taip, kad algoritmui nereikėtų spręsti tokios problemos, kaip slenkantis fonas bei objekto atstumas tarp kameros. Šiame darbe bus išbandoma kaip algoritmas gali prisitaikyti prie duomenų, kurie yra labiau randami kasdienybėje, pavyzdžiui kaip filmas. Pavykus eksperimentams, aprašytas procedūras, galima būtų pritaikyti, asmenų sukeitime vaizdinėje medžiagoje.

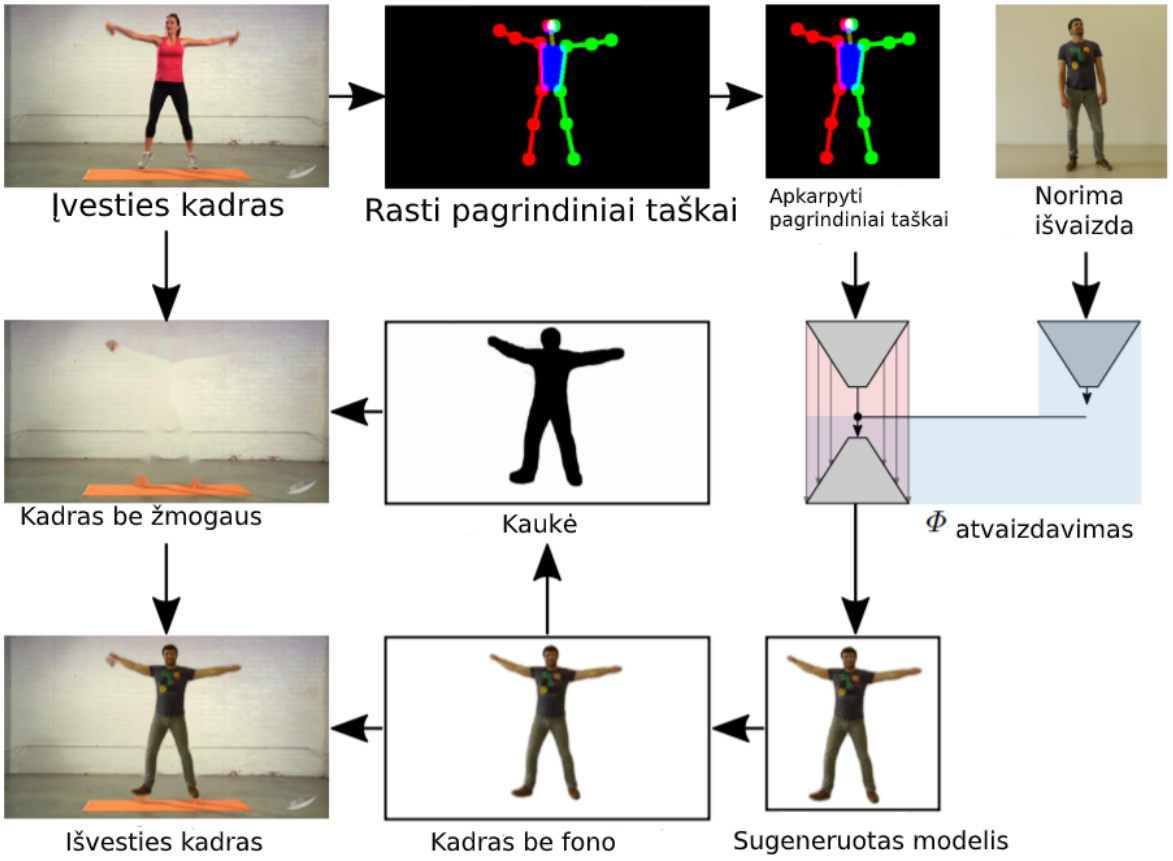
5. Literatūros apžvalga

Šiame skyriuje bus apžvelgti pagrindiniai darbai, kuriuose yra aprašomi nuotraukos animavimo algoritmai.

5.1 Žmogaus judėjimo simuliacija

Realus žmogaus elgesio generavimas yra labai svarbi sritis, kurios pritaikymas yra labai dažnas tokiose srityse kaip animacijoje, virtualioje realybėje bei kompiuterinių žaidimų kūrime. Paprastai veiksmus atliekančių asmenų animacijos yra perteikiamos iš 3D modelių. Šie modeliai dažniausiai yra suprojektuoti rankiniu būdu, naudojantis specialiaisiais jutiklių duomenimis. Šiame skyriuje bus apžvelgiamas darbas[EHM18], kuriame nagrinėjama, kaip leisti generuoti žmogaus animaciją, naudojantis duomenimis, kurie yra lengvai pasiekiami internete. Tam, kad pasiekti rezultatus, buvo panaudoti specialių jutiklių duomenys, kuriuos išmoksta neuroninis tinklas. Taip pat buvo išnagrinėta, kaip sugeneruoti animaciją iš vaizdinės informacijos. Rašoma tyrimo tematika, yra labiau susijusi su žmogaus animavimu, remiantis vaizdine įvestim, todėl nebus nagrinėjama metodika, kaip generuoti žmogaus animacijas, naudojantis jutiklių duomenimis.

5.1.1 Algoritmas



pav 1. Žmogaus modelio generavimas [EHM18]

Pirmame algoritmo etape yra atliekama koordinačių normalizacija. Norint išmokti žmogaus stovėseną bei pagrindinius taškus, būtų labai ne efektyvu mokyti tinklą su nuotraukomis, kuriose žmogus yra pavaizduotas skirtingose vietose. Pats modelio generavimas neturėtų priklausyti nuo asmens pozicijos nuotraukoje. Tam, kad paduoti tinklui tinkamo formato nuotrauką, yra randami žmogaus sąnarių pagrindiniai taškai ir aplink juos yra apibrėžiamas kvadratas su 10% padidintu plotu. Tuomet iškirpus šį kvadratą, apdorojimą galima tęsti toliau.

Mokyme yra naudojama didelė aibė nuotraukų, kuriose žmonės yra atvaizduojami skirtingose stovėsenose. Kiekvienoje mokymo aibėje esančioje nuotraukoje yra gaunamos segmentacijos kaukės (plačiau žiūrėti skyriuje 5.3.3 Mask R-CNN), bei žmogaus sąnarių pagrindiniai taškai. Mokant tinklą yra naudojama “COCO” duomenų aibė [LMB14]. Išmokti sąnarių pozicijas galima tiksliai, nes duomenų aibėje jos yra pažymėtos. Taip pat yra pažymėtos ir kaukės. Iš esmės tinklo mokymo metu, tinklas turi išmokti sugeneruoti modelį iš dviejų objektų: žmogaus išvaizdos ir supaprastinto pagrindinių taškų atvaizdo. Kad neapkrauti tinklo

mokymo metu, pagrindiniai taškai yra paverčiami į “stickman” formato nuotrauką. Šioje nuotraukoje visiškai nesimato nieko, išskyrus abstrakčiai pavaizduotos žmogaus stovėsenos. Taigi tinklo mokymo procedūrą galima įsivaizduoti šitaip:

1. Iš įvesties kadro, gaunami sąnarių pagrindiniai taškai.
2. Sukuriama supaprastinta figūra, kurioje yra tiesiog pagrindiniai taškai.
3. Ši figūra yra iškerpama ir perduodama tinklui kartu su nuotrauka, kurioje yra žmogus, pagal kurio išvaizdą yra generuojamas modelis.
4. Tinklui išmokus rekonstruoti modelį, yra apskaičiuojamos koordinatės originalioje nuotraukoje ir taip yra suformuojamas galutinis vaizdas tik be fono.
5. Iš šio modelio yra gaunama kaukė, kuria naudojantis yra sugeneruojamas fonas originalioje nuotraukoje.
6. Galutinis modelis yra įklijuojamas į finalinę nuotrauką.

Šį algoritmą galite pamatyti pav. 1.

5.1.2 Apibendrinimas

Šiame darbe daugiau dėmesio buvo skirta žmogaus modelio generavimui, naudojantis jutiklių duomenimis, negu iš vaizdinės medžiagos. Kadangi mano tyrime jutiklio duomenys bus neprieinami, buvo nagrinėta dalis, kuri žmogaus modelio judėjimą perteikia iš vaizdo įrašo. Šiame darbe galima išskirti šias problemas:

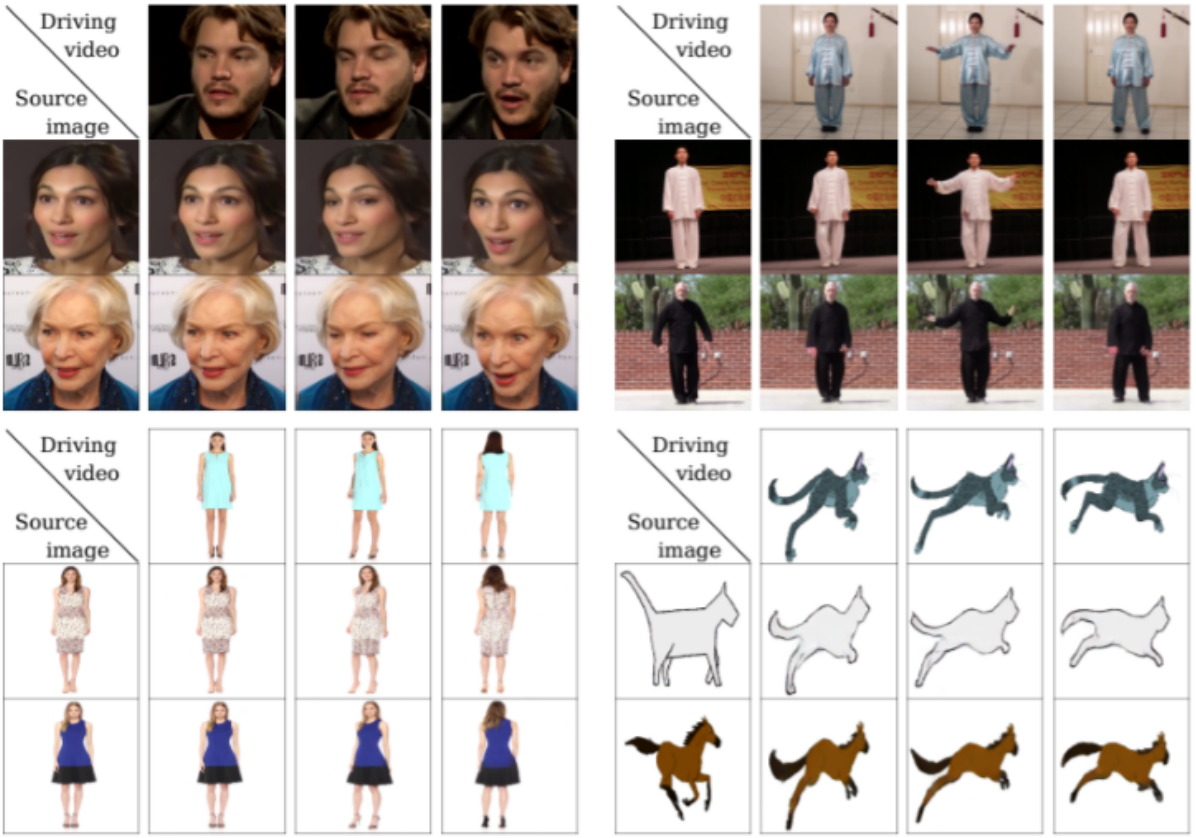
1. Netgi tobuliausiose sąlygose (kamera nejuda, žmogus neišeina iš kadro) yra aiškiai matomos ydos modelio generavime, todėl, kad matosi kadrai, kuriuose pradingsta žmogaus dalys (galva, kojos). Šie kadrai sudaro labai mažą procentą, tačiau jų pakanka nusakyti, kad įrašas yra sugeneruotas kompiuterio.
2. Kadangi fonas su kiekvienu kadru yra piešiamas iš naujo, aiškiai matosi, kad algoritmas ne tolygiai jį generuoja. Tai galima spręsti iš susiliejusios aplinkos aplinkui įdėta modelį.
3. Šio darbo suprogramuotas modelis nėra viešai prieinamas, todėl jo negalima pabandyti įvairesnėse situacijose, nei yra duoti pavyzdžiai.

5.2 Pirmos eilės judėjimo modelis

Nuotraukos animacija susideda iš vaizdo įrašo sukūrimo, kuriame objektas esantis nuotraukoje suanimuojamas, remiantis vaizdo įraše išreikštu judėjimu. Vienas iš tiksliausių dabar egzistuojančių algoritmų yra “Pirmos eilės judėjimo modelis” (angl. First order motion

model)[SLT19a]. Šio darbo pagrindinė idėja buvo sukurti algoritmą, kuris visiškai nenaudoja anotacijų arba kitos su vaizdo įrašu/nuotrauka susijusios informacijos. Tam, kad tai pasiekti buvo naudojama neuroniniais tinklais paremta formulė, kuri atskiria išvaizdos ir judėjimo informaciją.

Nuotraukos animacijoje yra dvi įvestis: nuotrauka (angl. source image), kuri nusako animuojamo objekto išvaizdą ir vaizdo įrašas (angl. driving video), kuriame nusakoma, kaip objektas turi judėti. Pavyzdžiui, nuotrauka su žmogaus veidu gali būti suanimuojama remiantis vaizdo įrašu, kuriame kitas asmuo kalba(pav 2.).

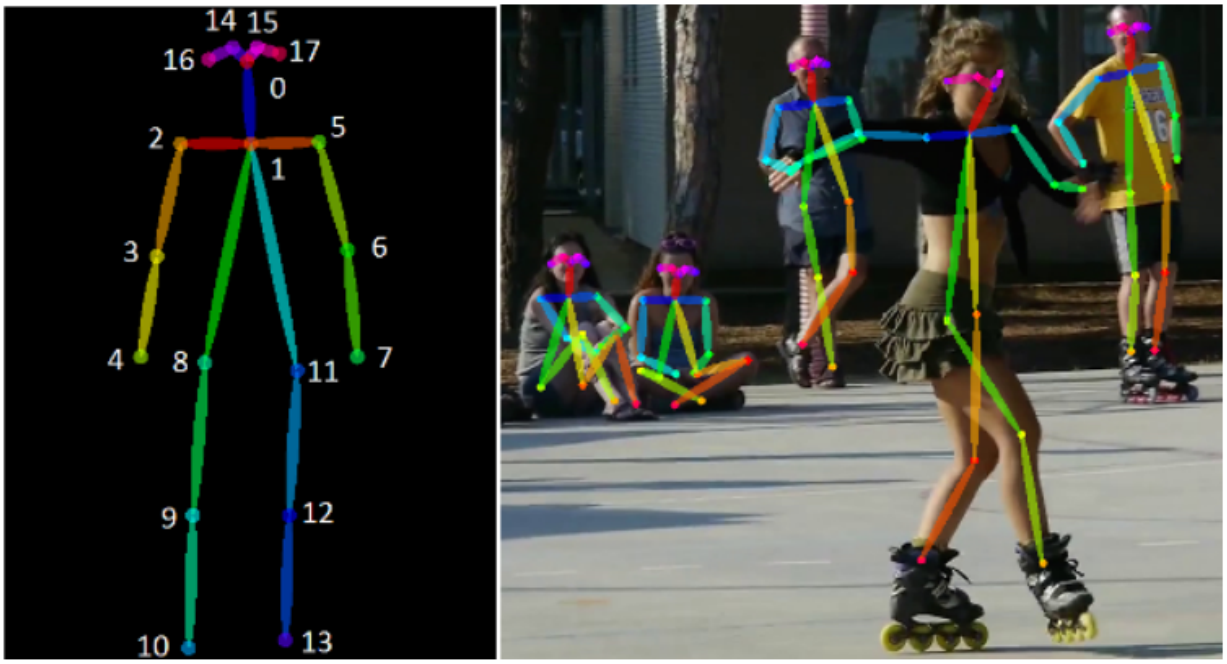


pav 2. Įvairių tipų pavyzdinės animacijos [SLT19a]

Minėta užduotis dažniausiai yra įvykdoma, remiantis anksčiau sužymėta anotacija ir ,remiantis ja yra naudojami tam tikri kompiuterinės grafikos metodai [CHZ14]. Šie metodai yra vadinami “specifinio-objekto” (angl. object-specific) metodai, todėl, kad jie naudoja specifinę aprašytą objekto informaciją. Tokio pobūdžio metodų veikimas priklauso grynai nuo pažymėtos anotacijos ar 3D modelių, ko pasekoje yra labai sudėtinga tokį metodą perprogramuoti skirtingiems objektams.

Algoritmai paremti giliais neuroniniais tinklais, sugeba nukonkuruoti minėtus algoritmus, todėl, kad jie sugeba išmokti judesį ir išvaizdą kaip du skirtingus konceptus.Tokie tinklai kaip “Generative Adversarial Network [GPM14] arba “Variable Auto Encoder”[DM14] yra

naudojami tam, kad išmoktų judėjimo modelius. Tačiau šie metodai dažniausiai priklauso nuo išmokintų modelių tam, kad ištraukų su objektu susijusią informaciją. Šie modeliai yra išmokinti iš žmonių surinktos aibės, kas išties riboja tinklo pritaikomumą kitokiem objektams. Atsižvelgiant į šias problemas buvo sukurtas pirmasis, nuo duomenų aibės nepriklausantis modelis “Monkey-Net” [SLT19b]. Šis tinklas užkoduoja su judesiu susijusią informaciją, naudojantis pagrindiniais taškais. Pagrindiniai taškai nusako specialų regioną, kurį įmanoma atpažinti nuotrauką didinant, mažinant, sukinėjant, perspalvojant bei sumažinant rezoliuciją(pav. 3).

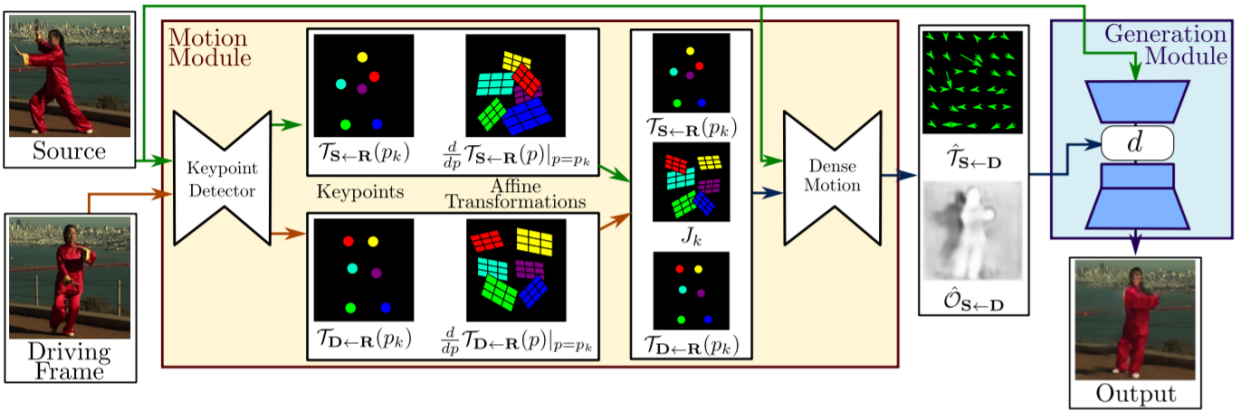


Pav 3. Žmoguje aptikti pagrindiniai taškai.

Testo metu, nuotrauka yra suanimuojama, remiantis rastomis pagrindinių taškų trajektorijomis vaizdo įraše. Pagrindinė “Monkey-Net” silpnybė yra prasta kokybė animuojant atvejais, kai judesys yra labai didelis. Tam, kad išspręsti šią problemą “First Order Motion” modelį yra naudojami išmokti pagrindiniai taškai, kurie yra transformuojami tam, kad atkurti sudėtingus veiksmus. Taip pat yra sprendžiama persidengimo problema. Jinai sprendžiama naudojant vaizdo generatorių, kuris automatiškai atkuria animuojamo objekto dalis, kurių nuotraukoje nesimato. Tai yra labai svarbi algoritmo dalis, tuomet kai vaizdo įraše yra atliekami sudėtingi judesiai.

5.2.1 Algoritmas

Algoritmo užduotis yra suanimuoti objektą, kuris yra pavaizduotas nuotraukoje **S**, remiantis panašaus tipo objekto judesiu, esančio vaizdo įraše **D**. Tinklo mokymui yra naudojama didelė aibė vaizdo įrašų, kuriuose yra tos pačios kategorijos objektai. Modelis yra išmokytas atgaminti mokymo aibėje esančius vaizdo įrašus, remiantis vienu kadru ir išmokto judesio reprezentacija. Tinklas gaudamas kadrų poras, kurios yra iš to pačio įrašo išmoksta užkoduoti judesį, kurį išreiškia kaip pagrindinių taškų transformacijas. Testo metu modelis sukuria poras kadrų, kuriuose yra nuotrauka (angl. source image) ir visi atskiri kadrai iš vaizdo įrašo. Algoritmo veikimo principą galima pamatyti pav. 4.



Pav. 4 "First order motion" modelis.[SLT19a]

Modelis susideda iš dviejų pagrindinių modulių: judesio matavimo ir nuotraukos generacijos. Judesio matavimo modulis yra atsakingas už judesio lauko (angl. "Motion field") nuspėjimą iš duoto kadro $D \in R^{3 \times H \times W}$, kurio dimensijos yra $H \times W$ iš vaizdo įrašo **D** į pradinę nuotrauką $S \in R^{3 \times H \times W}$. Judesio laukas vėliau yra naudojamas tam, kad išdėstyti bruožus, apskaičiuotus iš S**,** su objekto išsidėstymu, esančiu D. Judesio laukas yra funkcija $T_{S \leftarrow D}: R^2 \rightarrow R^2$,kuri atvaizduoja kiekvieną pikselio vietą iš Dsu atitinkančia vieta **S.** $T_{S \leftarrow D}$yra dažnai vadinama, kaip atbulinis optinis srautas (angl. Backward optical flow). Yra įsivaizduojama, kad yra abstraktus kadras R. Algoritmas bando nuspėti dvi transformacijas: iš R į $S(T_{S \leftarrow R})$ ir iš R į $D(T_{D \leftarrow R})$. Tai tiesiogiai niekada nėra apskaičiuojama ir todėl šio kadro vizualizacija yra neįmanoma. Šis pasirinkimas leidžia atskirai apdoroti D ir S. Tai yra labai patogu, nes testo metu modelis gauna poras kadrų, susidedančių iš nuotraukos ir vaizdo įraše esančių kadrų, kurie gali būti labai skirtingi vizualiai. Vietoj to, kad tiesiogiai būtų bandoma spėti $T_{S \leftarrow R}$ ir $T_{D \leftarrow R}$, judesio matuotojas atlieka 2 žingsnius.

Pirmame žingsnyje, yra gaunamos transformacijos iš aibės išmoktų pagrindinių taškų. Taškų pozicijos esančios D ir S yra atskirai randamos naudojantis, kodavimo-atkodavimo tinklu. Šis tinklas gražina taškų pozicijas ir parametrus, reikalingus atlikti transformaciją.

Antrame žingsnyje, tankaus judesio tinklas sujungia gautą išvestį ir yra gaunamas judesio laukas $T_{S \leftarrow D}$. Tinklas taip pat sugeneruoja taip vadinamą "Occlusion" kaukę $O_{S \leftarrow D}$, kuri nusako, kurios objekto dalys gali būti sukonstruojamos iš duotos nuotraukos, o kurios turi būti sugeneruojamos, remiantis kontekstu.

Galiausiai, generacijos modulis generuoja pradinėje nuotraukoje esantį objektą su judėjimu, esančiu pradiniame vaizdo įraše. Čia yra naudojamas generacijos tinklas G, kuris deformuoja pradinę nuotrauką remiantis, $T_{S \leftarrow D}$ir prideda nuotraukos dalis, kurių pradinėje nuotraukoje nėra.

5.2.2 Autorių eksperimentai

Šioje skiltyje bus pristatyti autorių padaryti eksperimentai.

5.2.2.1 Duomenų aibės

Autoriai atliko mokymus ir testus ant keturių skirtingų duomenų aibių, kuriuose yra įvairių objektų:

- "VoxCeleb" duomenų aibė [NCZ17] susideda iš 22496 vaizdo įrašų, kuriuose yra vaizduojami veidai. Paėmus vaizdo įrašą yra nustatoma veido pozicija pirmame kadre. Tuomet šis veidas yra sekamas iki tol, kol jis yra per toli nei pradinė pozicija. Tuomet vaizdo įrašas yra apkarpomas taip, kad būtų išimamas veidas ir nieko daugiau. Procesas yra kartojamas, kol apdorojami visi vaizdo įrašai. Įrašai, kurie turi mažesnę rezoliuciją nei 256 x 256 yra pašalinami, o likę vaizdo įrašai yra sumažinami iki minėtos rezoliucijos. Praėjus šias procedūras buvo gauti 12331 mokimui ir 444 testui skirti įrašai, su trukme nuo 64 iki 1024 kadrų.
- "Uva-Nemo" duomenų aibė [DSG12] yra veido išraiškų analizės aibė, kuri susideda iš 1240 vaizdo įrašų. Apdorojimas yra identiškas, kaip pirmosios duomenų aibės. Po filtravimo gauti 1116 įrašų yra naudojami mokymui ir 124 testui.
- "BAIR" duomenų aibė[EFL17] ,kurioje vaizdo įrašai susideda iš roboto, kuris judina įvairius daiktus, esančius ant stalo. Aibėje yra 42880 mokymuisi ir 128

testiniai vaizdo įrašai. Kiekvienas vaizdo įrašas susideda iš 30 kadrų ir turi 256 x 256 rezoliuciją. Kadangi aibė jau yra paruošta tai joks apdorojimas nereikalingas.

- “Tai-Chi-HD” aibė susideda iš “youtube” surinktų vaizdo įrašų su kovų menų tematika. Joje yra 252 vaizdo įrašų mokinimui ir 28 testavimui. Kiekvienas vaizdo įrašas yra išskaidomas į trumpus klipus, remiantis tuo pačiu principu, kaip “VoxCeleb” aibėje. Galiausiai, yra gauta 3049 vaizdo įrašų mokymui ir 285 testavimui, kur įrašo ilgis yra tarp 128 iki 1024 kadrų.

5.2.2.2 Metrikos

Įvertinti sugeneruotos animacijos kokybę praktikoje yra labai sunku, nes animuojant nuotrauką, realus rezultatas, su kuriuo būtų galima lyginti, neegzistuoja. Tam, kad įvertinti rezultatą buvo panaudotos metrikos, su kuriomis buvo vertintas “Monkey-Net”[SLT19b]. Buvo naudojamos tokios metrikos:

- L_1. kadangi testo metu mes turime vaizdo įrašus, kurie atitinka realybę, vienas iš būdų yra pamatuoti skirtumus tarp pikselio reikšmių, iš sugeneruoto ir realaus įrašų.
- “Average Keypoint Distance (AKD). “Tai-Chi”, “VoxCeleb” bei “Nemo duomenų aibei buvo naudojami išmokyti “keypoint” detektoriai tam, kad įsitikinti, kad įvesties įrašo judesys yra išsaugotas sugeneruotame įraše. “VoxCeleb” ir “Nemo” aibems naudojamas veido išraiškos detektorius[BT17]. “Tai-Chi” duomenų aibei naudojamas žmogaus stovėsenos matuotojas[CSW17]. AKD gaunamas apskaičiuojant vidutinį pikselių atstumą tarp aptiktų pagrindinių taškų realiame ir atkurtame įraše.
- “Missing Keypoint Rate” (MKR). “Tai-Chi” aibės atveju, žmogaus stovėsenos matuotojas, kiekvienam vaizdo įrašui gražina kiekį pagrindinių taškų rastų originaliame vaizdo įraše, bet nerastų sugeneruotame įraše procentas.
- “Average Euclidean Distance”(AED). Darbe yra apskaičiuojami išmoktais bruožais paremta metrika, kuri susideda iš AED apskaičiavimą tarp originalaus ir sugeneruoto įrašo. Tai parodo, kiek yra išsaugota objekto išvaizda iš pradinės nuotraukos. Šias metrikas skaičiuoja tam skirti išmokyti tinklai. “Nemo” ir “VoxCeleb” aibėms naudojamas veido išraiškos identifikacijos tinklas, o “Tai-Chi” aibei yra naudojamas tinklas skirtas viso žmogaus indentifikacijai.

5.2.2.3 Rezultatai

Eksperimento metu tyrimai buvo atlikti naudojantis 3 algoritmus: “X2Face” [WKZ18],” Monkey-Net”[SLT19b], bei “First Order Motion”[SLT19a]..

	L_1	AKD	MKR	AED
X2Face	0.080	17.654	0.109	0.272
Monkey-Net	0.077	10.798	0.059	0.228
First Order Motion	0.063	6.862	0.036	0.179

1. Lentelė. Bandymų su Tai-Chi aibe, rezultatai

	L_1	AKD	AED
X2Face	0.078	7.687	0.405
Monkey-Net	0.049	1.878	0.199
First Order Motion	0.043	1.249	0.140

2. Lentelė. Bandymų su Vox Celeb aibe, rezultatai

	L_1	AKD	AED
X2Face	0.031	3.539	0.221
Monkey-Net	0.018	1.285	0.077
First Order Motion	0.016	1.119	0.048

3. Lentelė. Bandymų su Nemo aibe, rezultatai

	L_1
X2Face	0.031
Monkey-Net	0.018
First Order Motion	0.016

4. Lentelė. Bandymų su Bair aibe, rezultatai

pav. 5. Pavyzdinis rezultatas

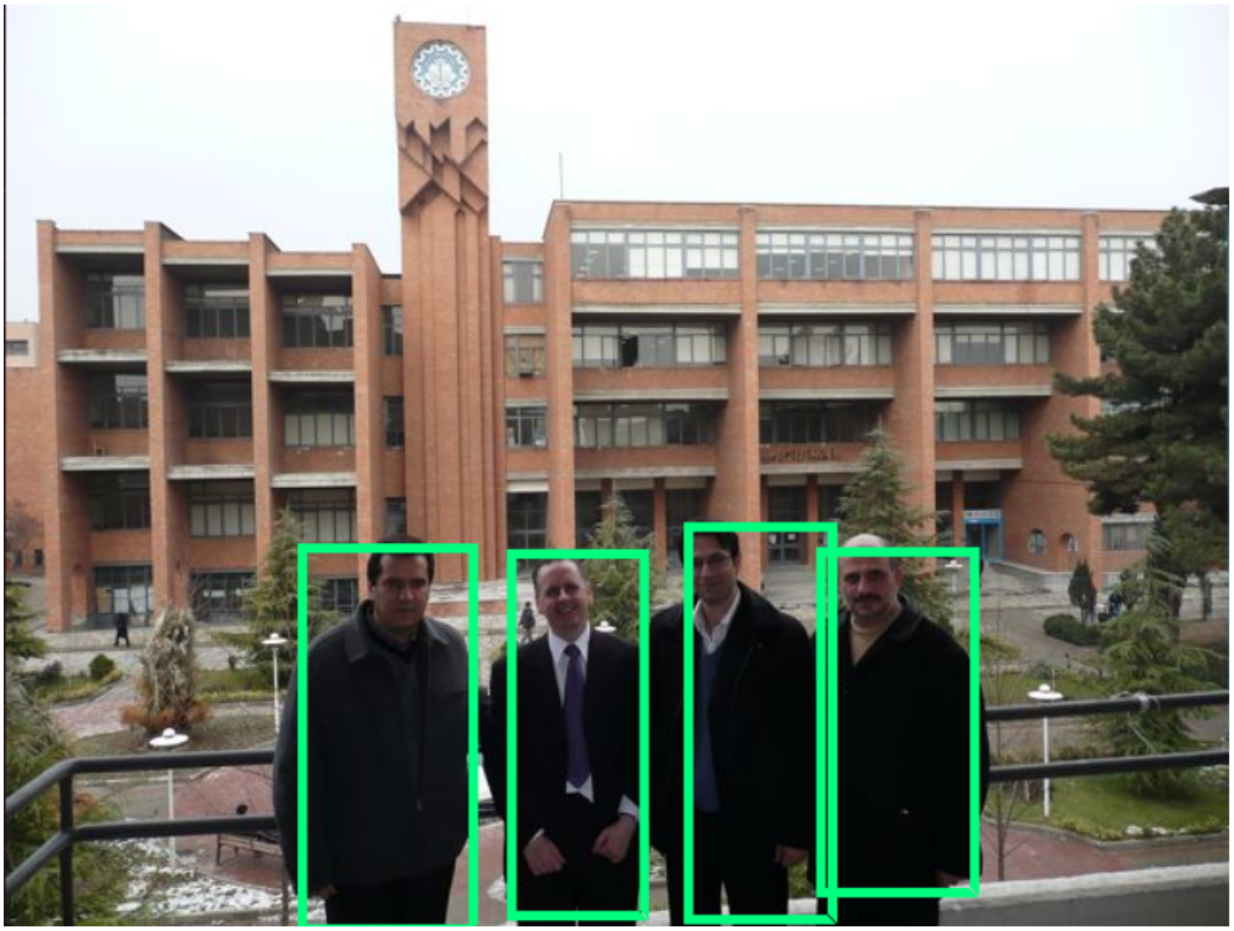
Pav. 6. Pavyzdinis rezultatas

5.2.3 Apibendrinimas

Iš rezultatų galima teigti, kad “First order motion” modelis yra akivaizdžiai geriausias. Šis modelis pralenkia konkurentus pagal visas metrikas. Tačiau tam, kad sėkmingai taikyti šį modelį, objektas vaizdo įraše ir nuotraukoje turi būti panašiose vietose. Tokios sąlygos yra praktiškai labai retos bet kokiame vaizdo įraše. Jeigu objektas yra skirtingose pozicijose, visas vaizdas tampa iškraipytas ir netikslus. Tam, kad taikyti šį modelį praktikoje, reikalingas didelis duomenų apdorojimas ir apkarpymas, kuris bus nagrinėjamas kituose skyriuose.

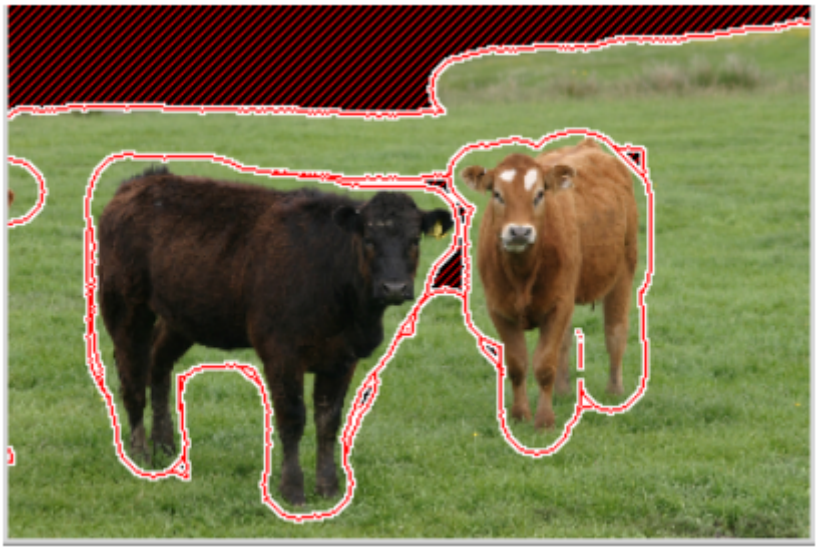
5.3 Animacijos paruošimas

Kadangi vaizdo įrašuose asmuo dažniausiai būna skirtingose pozicijose, todėl reikalingas būdas, kaip sekti asmenį su kuriuo norime atlikti keitimą. Pradinėje stadijoje bus daroma prielaida, kad žmogus vaizdo įraše yra vienas, tam kad supaprastinti pradinius eksperimentus. Tuomet reikalinga kitą sritis, kurią reikia išnagrinėti: asmens aptikimas vaizdo įraše. Šią užduotį sprendžia dvi pagrindinės neuroninių tinklų šakos: objektų segmentavimas ir objektų aptikimas. Šios sritys pagal reikšmę gali priminti sinonimus, tačiau jos yra skirtingos. Objekto aptikimo užduotis yra apibrėžti stačiakampį (angl. bounding box) aplinkui norimą aptikimo objektą (pav. 7)



Pav 7. Asmenų aptikimas.

Tuo tarpų objektų segmentavimas sprendžia užduotį, kuomet norima lygiai apibrėžti ieškomą objektą (pav. 8).



pav. 8 Žinduolių segmentavimas

Objektų aptikimas yra žymiai greitesnė operacija, tačiau nuo to nukenčia tikslumas. aptikimas yra plačiai naudojamas vietose, kur reikia greitai atpažinti kažkokį objektą, pavyzdžiui, kaip save

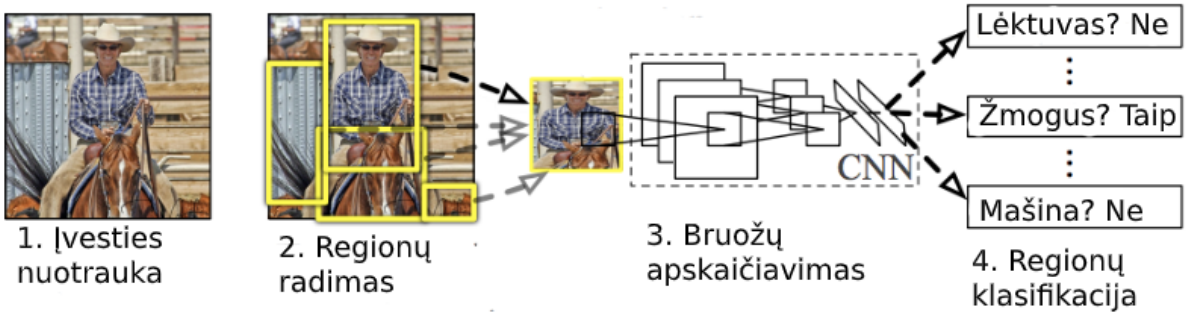
vairuojančiose mašinose. Tačiau segmentavimas turi pranašumą, kad visiškai tiksliau yra aptinkamas bei apibrėžiamas objektas. Todėl šiame darbe, kur greitis nėra pagrindinė užduotis, bus naudojamas objektų segmentacijos algoritmas. Tam, kad suprasti, kaip veikia tokie algoritmai, pirma bus trumpai aptarti baziniai modeliai ir tuomet bus pristatytas vienas iš tiksliausių algoritmų, kuris dabar egzistuoja “Mask R-CNN”.

5.3.1 Objektų aptikimas

Objektų aptikimas yra naudojamas įvairiose srityse: apsaugoje, save vairuojančiose mašinose, žmonių aptikime ir daugelyje kitų sričių. Pagrindinis skirtumas tarp klasifikacijos ir aptikimo yra toks, kad aptikime reikia ne tik pasakyti, kad nuotraukoje yra objektas, bet ir pasakyti tikslią jo vietą. Vienas iš didžiausių iššūkių šioje disciplinoje yra situacija, kai objektų nuotraukoje yra labai daug. Šios problemos negali išspręsti standartinis neuroninis tinklas todėl, kad išvesties sluoksnio ilgis nėra konstanta, nes nėra žinoma kiek ir kokių objektų gali būti nuotraukoje. Intuityvus sprendimas būtų išskaidyti nuotrauką į regionus ir naudoti konvoliucinį tinklą tam, kad klasifikuoti specifini regioną. Problema su šiuo sprendimu yra tokia, kad objektai gali būti įvairaus dydžio, gali patekti į tą patį regioną. Bandant taip išspręsti problemą, generuojamų regionų kiekis būtų milžiniškas. Procesas būtų arba labai lėtas, arba neįmanomas.

5.3.1.1 R-CNN

Tam, kad išspręsti šią problemą buvo sukurtas modelis “R-CNN” [GDD15](pav.9).



pav 9. “R-CNN” modelis

Pasiūlytame sprendime naudojamas algoritmas, kuris vietoj to, kad rinktų visus regionus, išrinktų 2000 geriausių. Šie 2000 regionų pasiūlymų yra sugeneruojami naudojantis atrankine paieška (angl. selective search)[USG13]. Šis algoritmas susideda iš 3 dalių:

1. Sugeneruoti pradinius regionus-kandidatus.
2. Naudojantis “Greedy” algoritmu sujungti panašius regionus į vieną didesnį.

3. Procedūra kartojama iki tol, kol pasiekiama 2000 galutinių regionų.

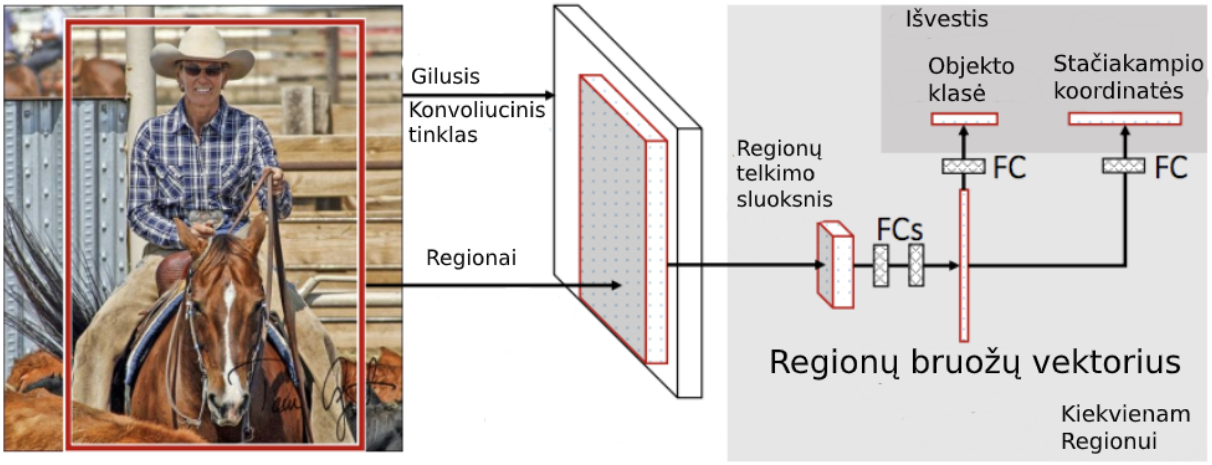
Gauti regionai perduodami į konvoliucinį tinklą, kuris gražina bruožų vektorių. Konvoliucinis tinklas perduoda regionus bei bruožus kitam sluoksniui, kuris bando nuspėti, kokios rūšies objektas yra regione. Kartu su objekto klasifikacija gražinamos regiono koordinatės. Šis algoritmas turi tris pagrindines problemas:

- Kadangi yra 2000 regionų, greitis vis tiek išlieka lėtas.
- Šis algoritmas nėra pritaikomas realiu laiku.
- Atrankinė paieška nėra neuroniniais tinklais paremtas algoritmas. Tai reiškia, kad jei regionų ieškojimo žingsnyje iškyla problemų, nėra būdų pagerinti rezultatų. To pasekoje gali būti sugeneruoti blogi regionai.

5.3.1.2 Fast R-CNN

Tas pats autorius nusprendė išspręsti R-CNN modelio minėtas problemas ir sukūrė naujesnę savo modelio versiją “Fast R-CNN”[Gir15]. Principas yra panašus į R-CNN algoritmą, bet vietoj to, kad regionus perduoti konvoliucijų tinklui, mes perduodame visą nuotrauką, iš kurios tinklas sugeneruoja bruožų vektorių. Tuomet naudojantis bruožų vektorių identifikuojami regionai ir, naudojantis regionų telkimo sluoksniu, jie paverčiami į formą, kurią apdoroja pilnai sujungtas sluoksnis (angl. Fully Connected Layer). Tuomet spėjimas vyksta taip pat kaip ir “R-CNN” modely.

“Fast R-CNN” yra greitesnis todėl, kad konvoliucijų tinklas atlieka operaciją vieną kartą, o ne 2000 kaip “R-CNN” modely.

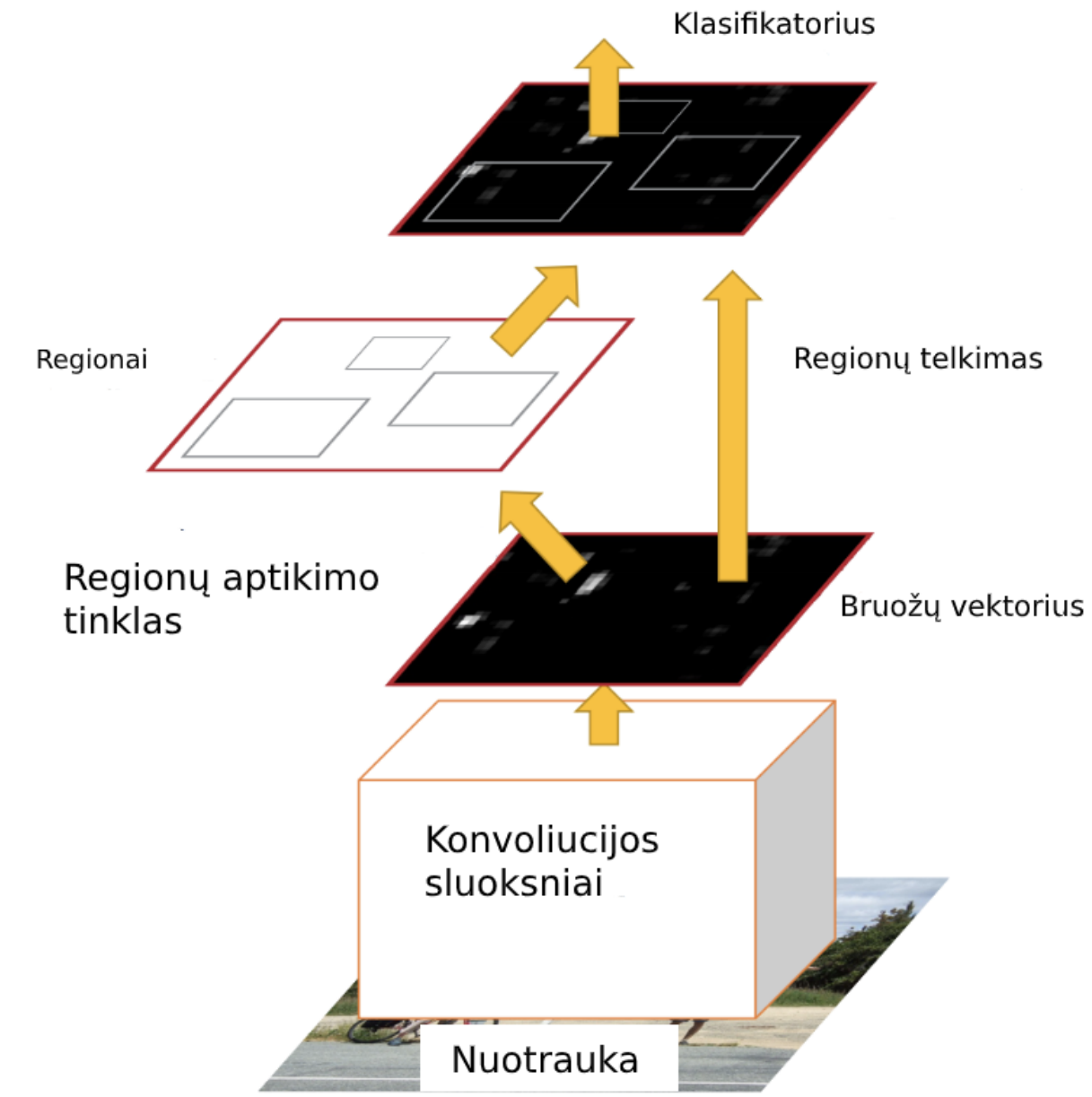


pav 10. Fast R-CNN modelis.

5.3.1.3 Faster R-CNN

Abudu pristatyti algoritmai (“R-CNN ir “Fast R-CNN”) naudoja atrankinę paiešką, kad rastų potencialius regionus. Atrankinė paieška yra lėta operacija, kuri labai sulėtina modelio veikimą. Tam, kad toliau patobulinti regionų radimą, buvo sukurta trečia “R-CNN” versija, pavadinimu “Faster R-CNN”[RHG15]. Šis algoritmas nenaudoja atrankinės paieškos, o leidžia tinklui išmokti atpažinti regionus.

Panašiai kaip “Fast R-CNN”, nuotrauka yra perduodama konvoliuciniui tinklui, kuris generuoja bruožų vektorių. Vietoj to, kad naudoti atrankinės paieškos algoritmą, atskiras tinklas bando atspėti tinkamus regionus. Tuomet gauti regionai yra paduodami į regionų telkimo sluoksnį ir eiga yra tokia pati, kaip “Fast-RCNN”. Ši algoritmo implementacija yra tiek pat tiksli, kaip senesnės versijos, tačiau yra ženkliai greitesnė. Šį algoritmą įmanoma taikyti realiu laiku.



pav 11. "Faster R-CNN" modelis

5.3.2 Regiono telkimo sluoksnis

Regiono telkimo sluoksnis (angl.) [HZR14] neuroniniuose tinkluose naudojamas objektų aptikimo užduotyje. Šis sluoksnis kaip įvestį gauna:

1. Fiksuoto dydžio bruožų vektorių, kuris buvo gautas iš kovoliucinio tinklo
2. $N \times 5$ dydžio matricą, kuri reprezentuoja rastus regionus, kur N yra regionų kiekis. Pirmasis stulpelis reprezentuoja nuotraukos identifikaciją, o likę keturi yra viršutinio kairiojo ir apatinio dešiniojo kampo koordinatės.

Tuomet kiekvienam regionui iš įvesties yra paimami atitinkami bruožai ir paverčiami į hiper parametruose nurodytą dydį. Dydžio keitimas susideda iš trijų žingsnių

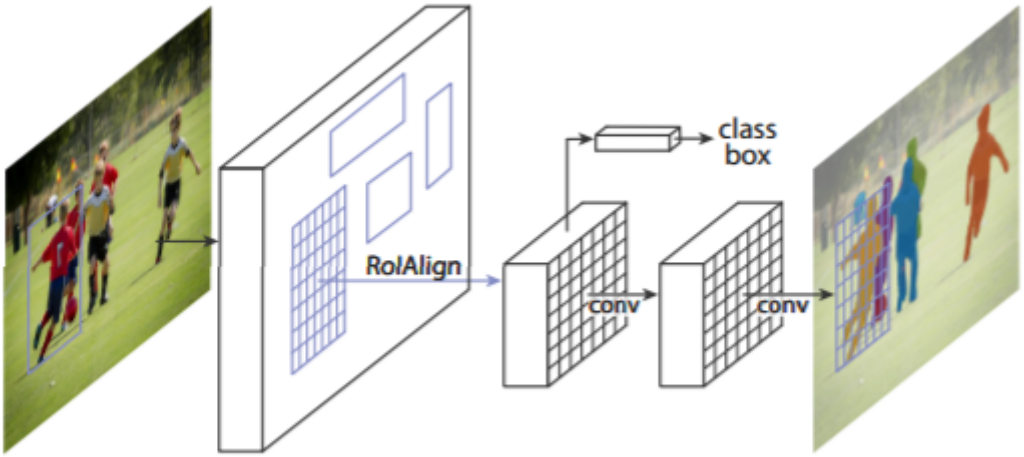
1. Regiono išskaidymas į vienodo dydžio skyrius.
2. Rasti didžiausią reikšmę skyriuose.
3. Rastas reikšmes patalpinti į išvesties vektorių.

Šis sluoksnis pagreitina tinklo veikimo laiką.

5.3.3 Mask RCNN

Objektų segmentacija yra sudėtinga užduotis dėl to, kad ją atlikti yra reikalingas teisingas visų objektų indentifikavimas nuotraukoj kartu su sėkmingų kiekvieno objekto segmentavimu. Šioje srityje yra sujungiami bruožai iš klasikinės neuroninių tinklų sričių, kaip objektų aptikimo, kur užduotis yra lokalizuoti objektą ir apibrėžti aplink jį stačiakampį ir semantinės segmentacijos , kur nagrinėjama, kaip klasifikuoti kiekvieną pikselį, į duotą kategorijų aibę.

Metodas, kuris bus nagrinėjamas yra vadinamas “Mask R-CNN” [KGP17]. Šis modelis yra patobulinta tinklo “Faster R-CNN” versija, kurioje yra pridedamas papildomas išvesties bruožas, kuris skirtas nuspėti segmentacijos kaukes kiekvienam rastam regionui. Segmentacijos kaukė yra pikselio tikslumu nusakomas filtras, kuris identifikuoja objektą. Kaukė yra randama papildomai taikant konvoliucinį tinklą kiekvienam rastam regionui. Abstraktų veikimo principą galima pamatyti pav. 12.



pav 12. Mask R-CNN modelis

“Mask R-CNN” pratęsia “Faster R-CNN”, pridedant papildomą šaką, kuri yra atsakinga už segmentacijos kaukių radimą kiekviename rastame regione. Kaukės radimo šaka yra mažas konvoliucinis tinklas, pro kurį praeina kiekvienas rastas regionas, kuriame yra nuspėjama kaukė pikselio tikslumu. “Faster R-CNN” nėra pritaikytas objektus rasti pikselio tikslumų. Regiono telkimo sluoksnis, atlieka operaciją, kuri sutraukia regionuose rastus bruožus į mažesnį vektorių, todėl nėra įmanoma objekto nusakyti pikselio tikslumu. Tam, kad išspręsti šią problemą, buvo

pasiūlyta perdaryti sluoksnį, kuriame nebūtų supaprastinti regiono bruožai. Šis sluoksnis vadinamas “RoIAlign”, sugeba išsaugoti originalias erdvines koordinates. Visų antra, “Mask R-CNN” algoritme yra atskiriamas kaukės bei klasės spėjimas. Klasės spėjimas yra visiškai nepriklausomas nuo kaukės ir tai reiškia, kad pats atpažinimas visiškai nėra modifikuotas nuo “Faster R-CNN” modelio.

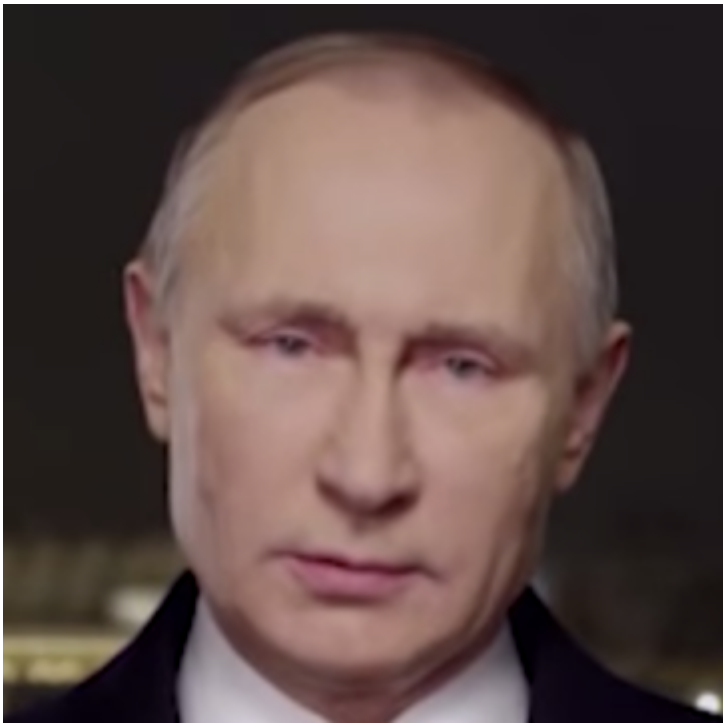
6. Autoriaus darbų demonstracijos

Šioje skiltyje bus išbandomas “First Order Motion” modelis. Programinis kodas yra viešai pasiekiamas iš “Github” svetainės. Autoriai yra paruošę eksperimentams skirtą modulį, kurį galima lengvai paleisti debesijos aplinkoje : “Google Colab”.

6.1 Pirmoji demonstracija

Šioje demonstracijoje bus generuojamas pats paprasčiausias variantas: žmogaus kalbėjims. Autoriai yra patalpinę tinklo konfiguracijas, kurias galima pasiimti, kad nereikėtų tinklo mokyti rankiniu būdu. Naudojama konfiguracija buvo gauta mokinant tinklą naudojantis “VoxCeleb” duomenų aibe. Pirmajame pavyzdyje būtent iš ten ir yra paimti duomenis. Įvesties nuotrauka, bei vaizdo įraša galime pamatyti pav. 13 ir pav. 14.

pav. 13 Vaizdo įrašas (driving video)



pav. 14 Įvesties nuotrauka (source image)

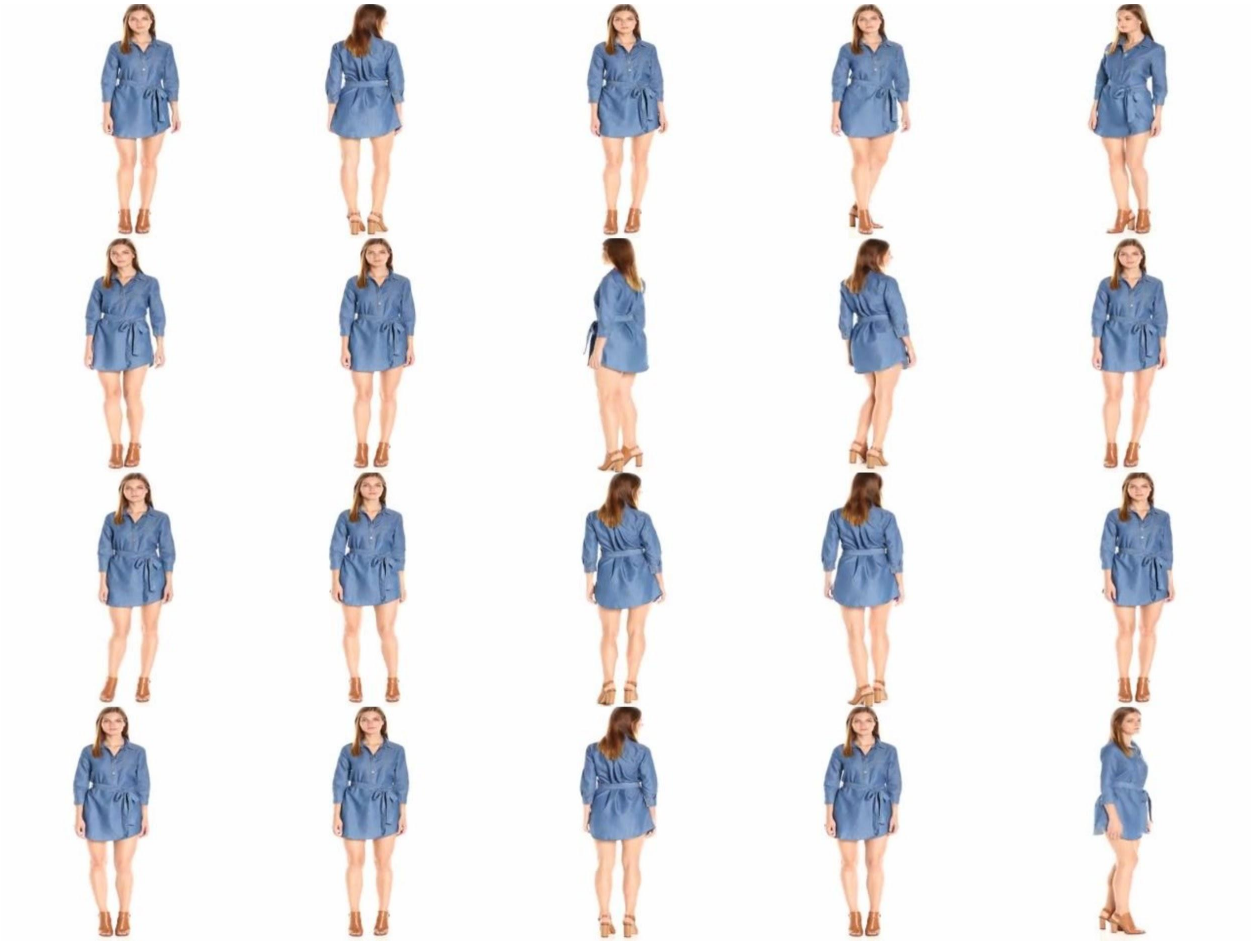
Kaip matoma, asmenys, esantys tiek vaizdo įraše, tiek nuotraukoje yra labai panašiose pozicijose ir panašaus dydžio. Rezultate (pav. 15) yra matoma atkurta animacija. Verta paminėti, kad asmenys yra tose pačiose pozicijose, bet fonas yra tas pats, kuris yra originalioje nuotraukoje. Nors nuotraukoje galvos dalies, kuri yra vaizdo įraše nėra, tinklas sugeba generuoti, tą galvos dalį.

pav. 15 Rezultatas

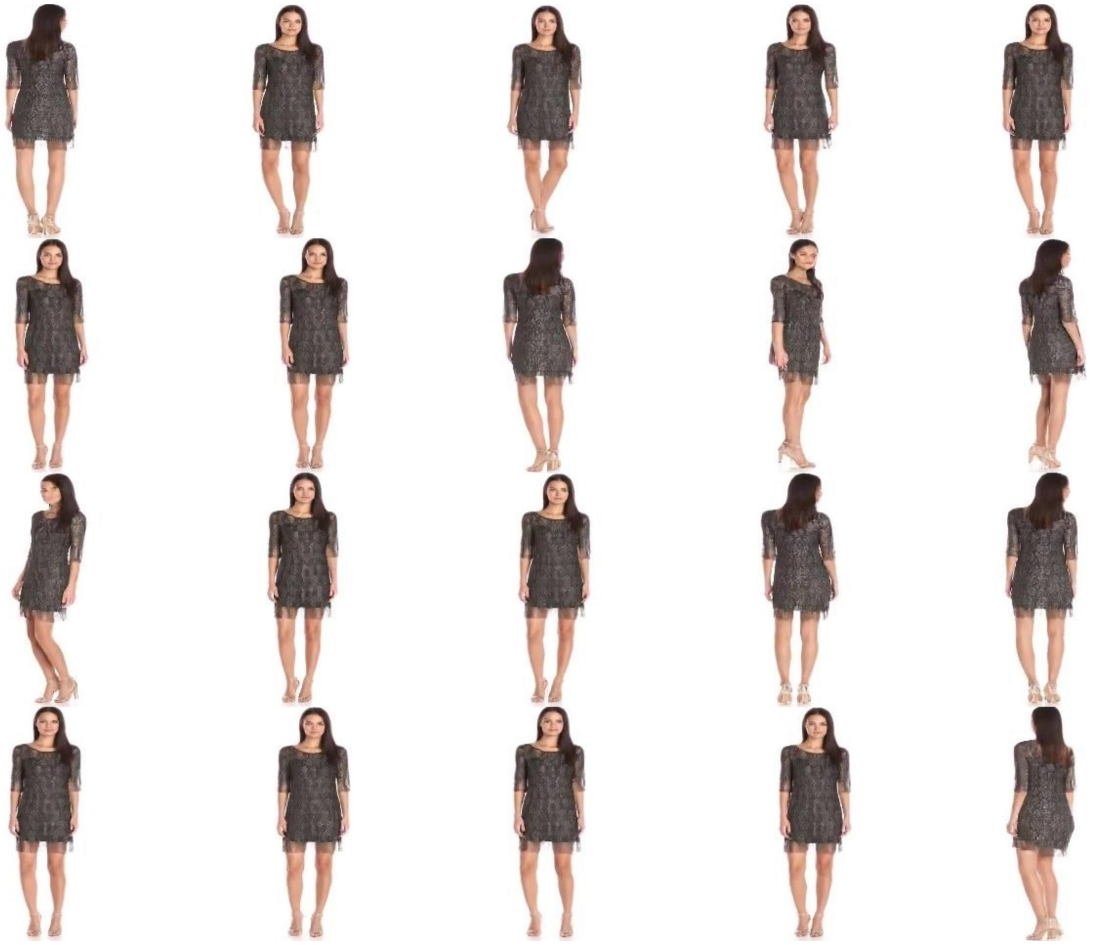
6.3 Antroji demonstracija

Šioje demonstracijoje yra atkuriamas visas žmogus. Tam bus panaudota “Fashion Video Dataset” duomenų aibėje esančių rūbų demonstracija.Įvesties nuotrauką pavaizduota pav 16, o vaizdo įrašas pav 17.

pav 16. Įvesties nuotrauka (source image



pav. 17 Vaizdo įrašas (driving video)



pav 18. Rezultatas

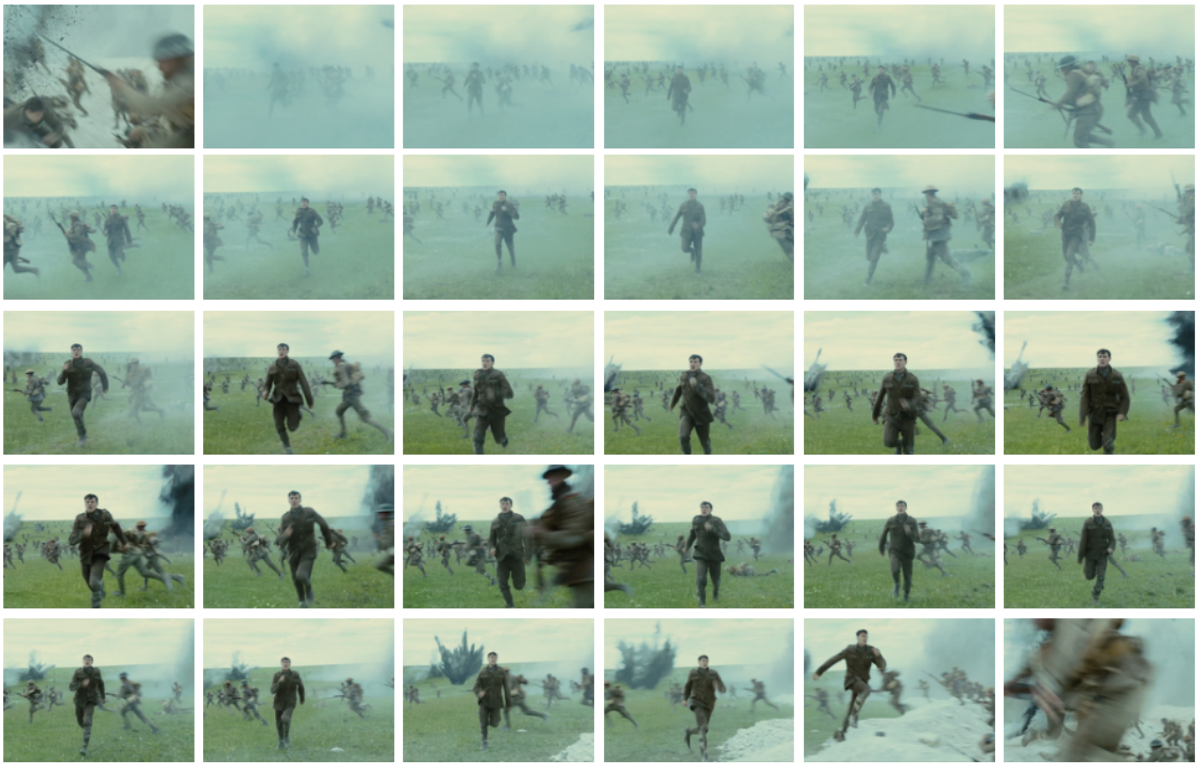
7. Praktinė dalis

Pilno kūno animaciją galima sugeneruoti tuomet, kada yra puikios sąlygos: tiek nuotrauka, tiek vaizdo įrašas turi būti tokio pačio dydžio, toje pačioje pozicijoje, turi turėti kuo imanoma paprastesnį foną. Tačiau daugelyje vaizdo įrašų, pavyzdžiui, kaip filmuose, asmens pozicija kadre keičiasi, asmens dydis kadre irgi nėra pastovus ir fonas sudarytas ne iš balto fono, kaip buvo matyta pateiktame pavyzdyje. Taigi, tam, kad generuoti animacija naudojantis duomenų aibe, kuri nėra tam pritaikyta, reikia išspręsti daugelį naujų problemų, kurios nebuvo aprašytos “First order Motion” autoriaus darbe. Šioje praktinėje dalyje bus sprendžiamos visos minėtos problemos.

7.1 Pirmasis eksperimentas

Tam, kad supaprastinti darbą, bus naudojama filmo ištrauka, kurioje norimas sukeisti personažas yra visados matomas. Vaizdo įrašas, kuris bus naudojamas, yra filmo “1917” (*pav.*

19) ištrauka, kurioje personažas bėga per lauką. Šiame vaizdo įraše personažo pozicija kadre pastoviai keičiasi, kamera yra atitolinama ir priartinama, todėl ir jo dydis yra pastoviai besikeičiantis. Taip pat personažas yra užstojamas kitų personažų.



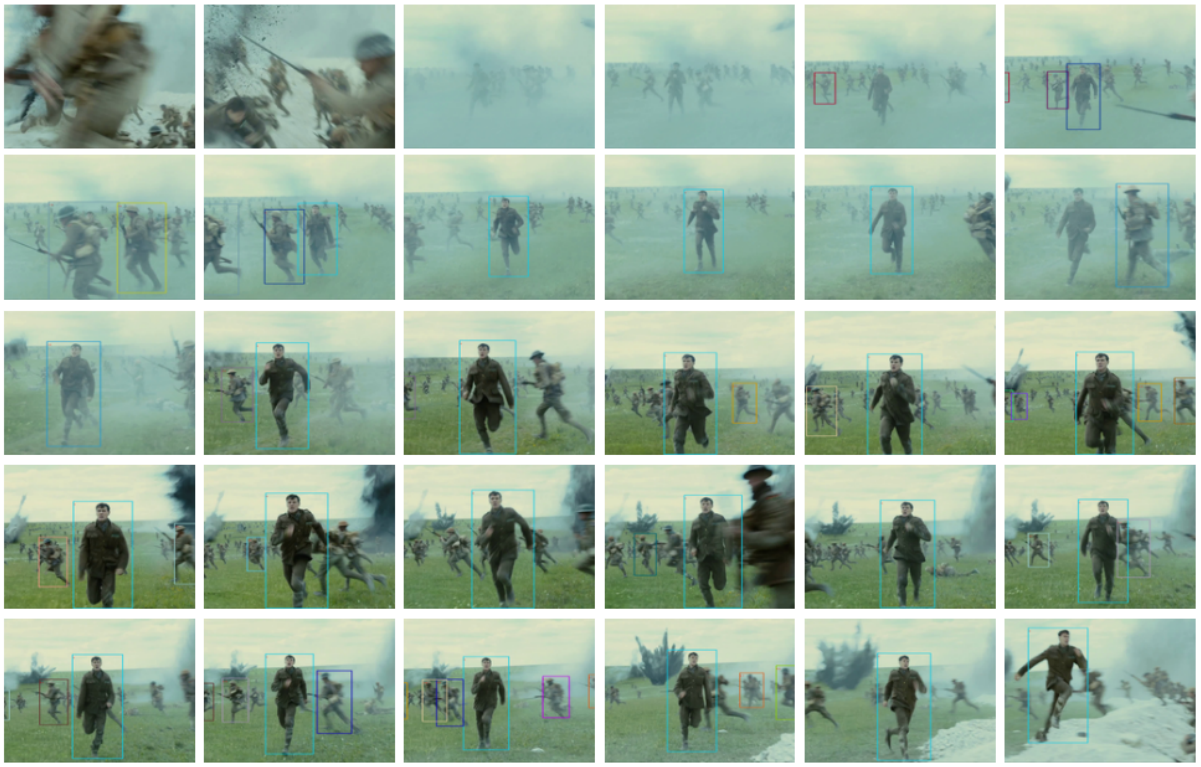
pav. 19 Filmo "1917" ištrauka.

Tam, kad pritaikyti "First order motion" modelį, reikia vaizdo medžiagą paruošti taip, kad algoritmas visą laiką tokiam pačiam formate matytų norimą pakeisti personažą. Tam bus reikalingas asmens sekimas, asmens iškirpimas, fono išėmimas ir daugelis kitų paruošimo procedūrų. Pagrindinė žingsnių seka būtų tokia:

1. Aptikti pagrindinį personažą vaizdo įraše ir gauti koordinates stačiakampio, kuris visados gaubtų jį.
2. Gavus stačiakampį iškirpti personažo siluetą.
3. Išimti foną ir sudaryti vaizdo įrašą, kuriame personažas judėtų baltame fone
4. Naudojantis "First order motion" modeliu, sugeneruoti animaciją naujam personažui.
5. Pasinaudojus pirmame žingsnyje gautomis koordinatėmis, naują animaciją įklijuoti, ko pasekoje bus padarytas "deepfake" vaizdo įrašas.

7.1.1 Objekto sekimas

Iš 19 pav. matosi, kad personažas pastoviai keičia pozicijas kadre, todėl reikalingas objekto sekimas.



19 pav. Filmo ištrauka su atpažintu personažu.

Kadangi objekto kaukės radimas yra labai brangi operacija, pirma bus aptinkama personažo pozicija kadre bei stačiakampis, kuris gaubia šį asmenį. Šiai užduočiai panaudotas “FairMOT” neuroninis tinklas[YCX20]. Algoritmas kiekvienam aptiktam asmeniui priskiria “etiketę”(angl. label). Naudodamasi ja mes galime identifikuoti poziciją to pačio personažo per visą vaizdo įrašą. Tačiau vaizdo įraše yra momentas, kuomet pro personažą prabėga kitas žmogus, todėl algoritmas pameta išsaugotą etiketę ir priskiria jam kitą. Kadangi objekto aptikimas nėra pagrindinis šio tyrimo eksperimentas, paprastumo dėlei toliau bus nagrinėjami tik tie kadrai, kuriuose personažas yra aptiktas po prasilenkimo su kitu personažu. Taip pat verta paminėti, kad personažas nėra aptinkamas visuose kadruose. Kadangi algoritmas nesugeba 100% sekti personažo visą laiką, mes gautus rezultatus suskirstysime į epizodus. Epizodas bus sudarytas iš kadro rinkinio, kuriame visi kadrai yra su atpažintu personažu.

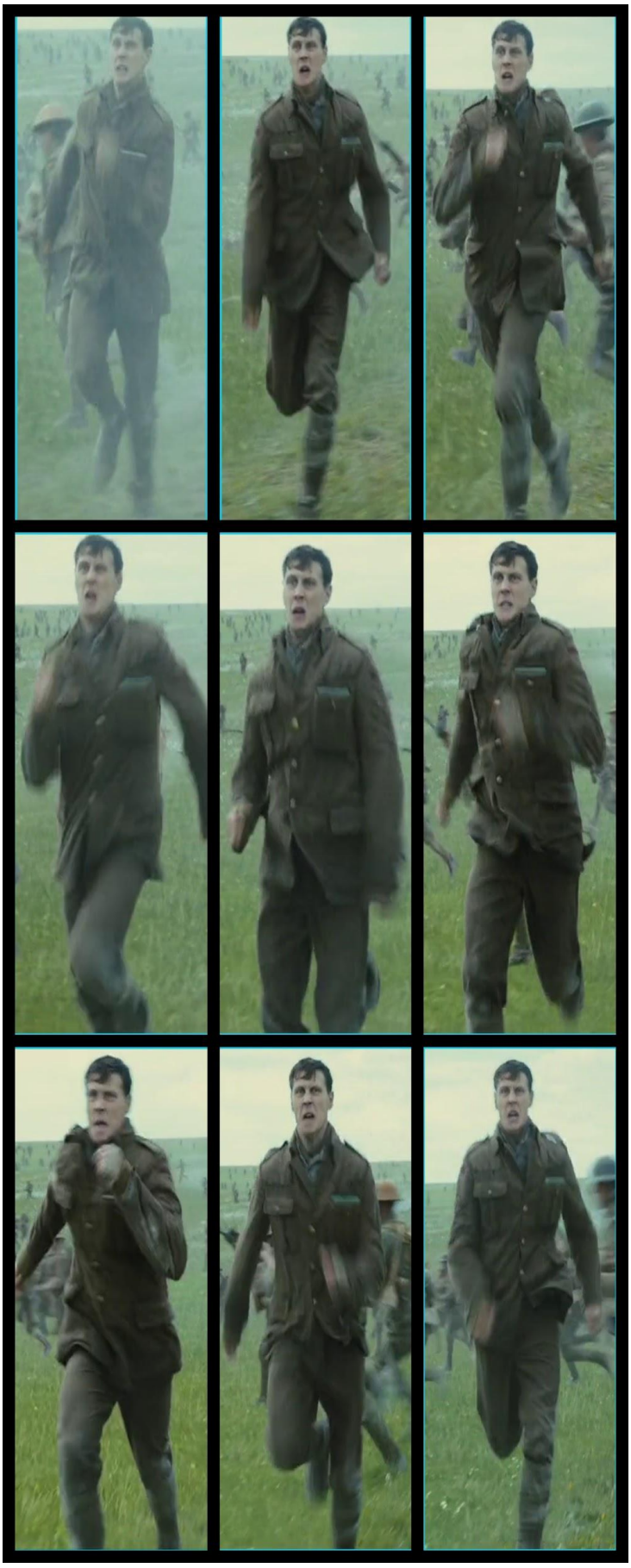
Iš viso gauname 10 epizodų. Šių epizodų dydžius galime pamatyti lentelėje 5.

Epizodo numeris	Kadrų kiekis
1	48
2	1
3	8
4	1
5	140
6	7
7	36
8	16
9	2

Tolimesniem bandymam naudosime penktą epizodą, dėl to, kad jame yra daugiausia kadrų.

7.1.2 Tinkamo formato vaizdo įrašo sudarymas

Paėmus penkto epizodo kadrus, kiekvienam kadrui yra iškerpamas pagrindinis personažas(pav. 20).



pav. 20 Iškirptas personažas kadre

Iš paveikslėlio matome, kad personažas kadre būna skirtingo dydžio, todėl iškirptų nuotraukų dydis skiriasi. Tam, kad sėkmingai galėtume sugeneruoti animaciją. mums reikia šias nuotraukas turėti to pačio formato. Tam, kad tai pasiekti, padarysim atitinkamo dydžio baltus rėmus. Visų pirma, susirasime nuotrauką, kurios rezoliucija yra pati didžiausia. Didžiausia rasta nuotrauka yra 220 kadras, kurios rezoliucija yra 352x913. Taigi, generuojama minėtos nuotraukos rezoliucijos baltą nuotrauką ir visus kadrus įklijuojame į ją taip, kad balti rėmai užpildytų likusią vietą. (pav. 21)

pav 21. Centruoti iškirpti kadrai

Turėdami šiuos rezoliucijos kadrus, galima sukonstruoti vaizdo įrašą, kuriame personažas visada yra viduryje.

7.1.3 Fono pašalinimas

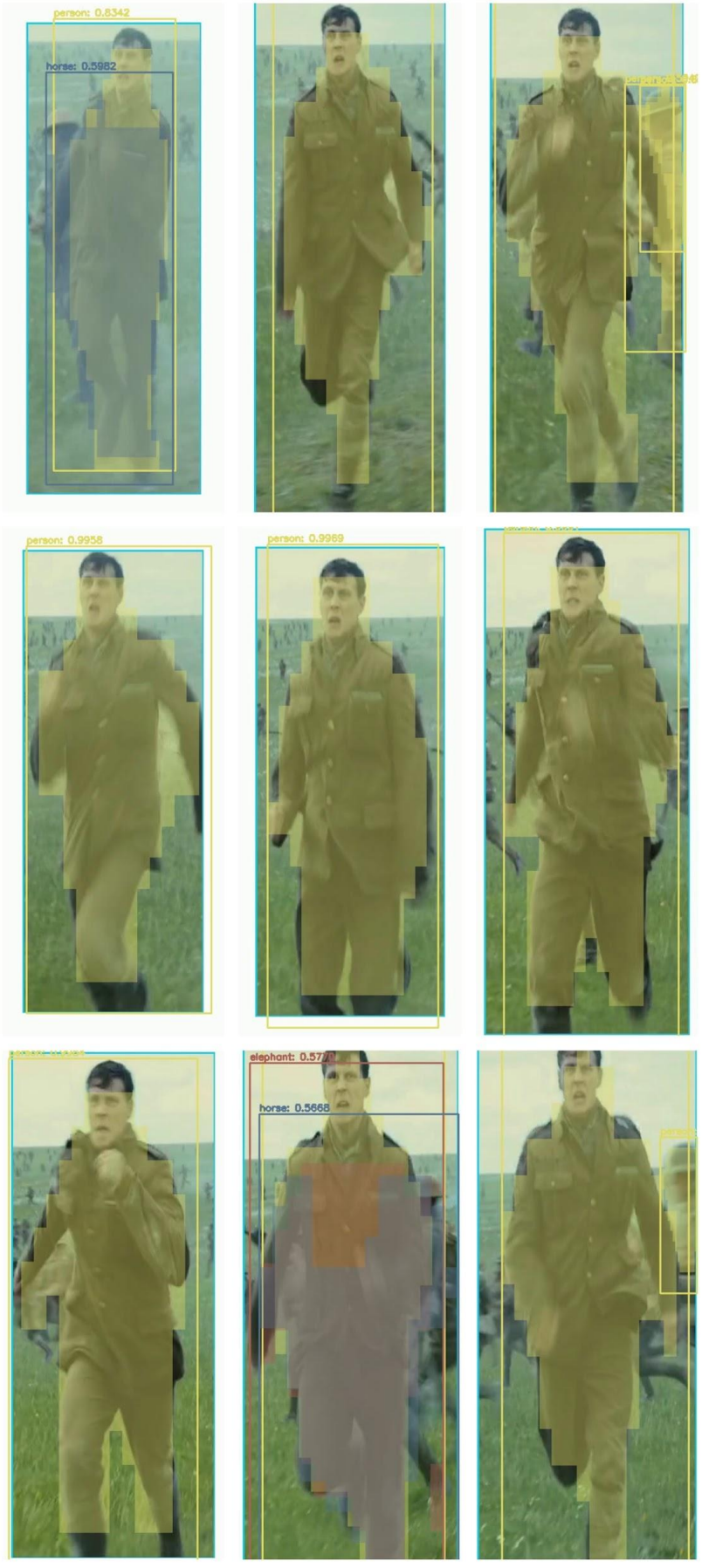
Tam, kad pašalinti foną, reikia rasti personažo kaukę (angl. mask). Kad pasiekti šį tikslą, bus išbandomi du algoritmai: Mask-RCNN ir U2net.

7.1.3.1 Mask R-CNN eksperimentas

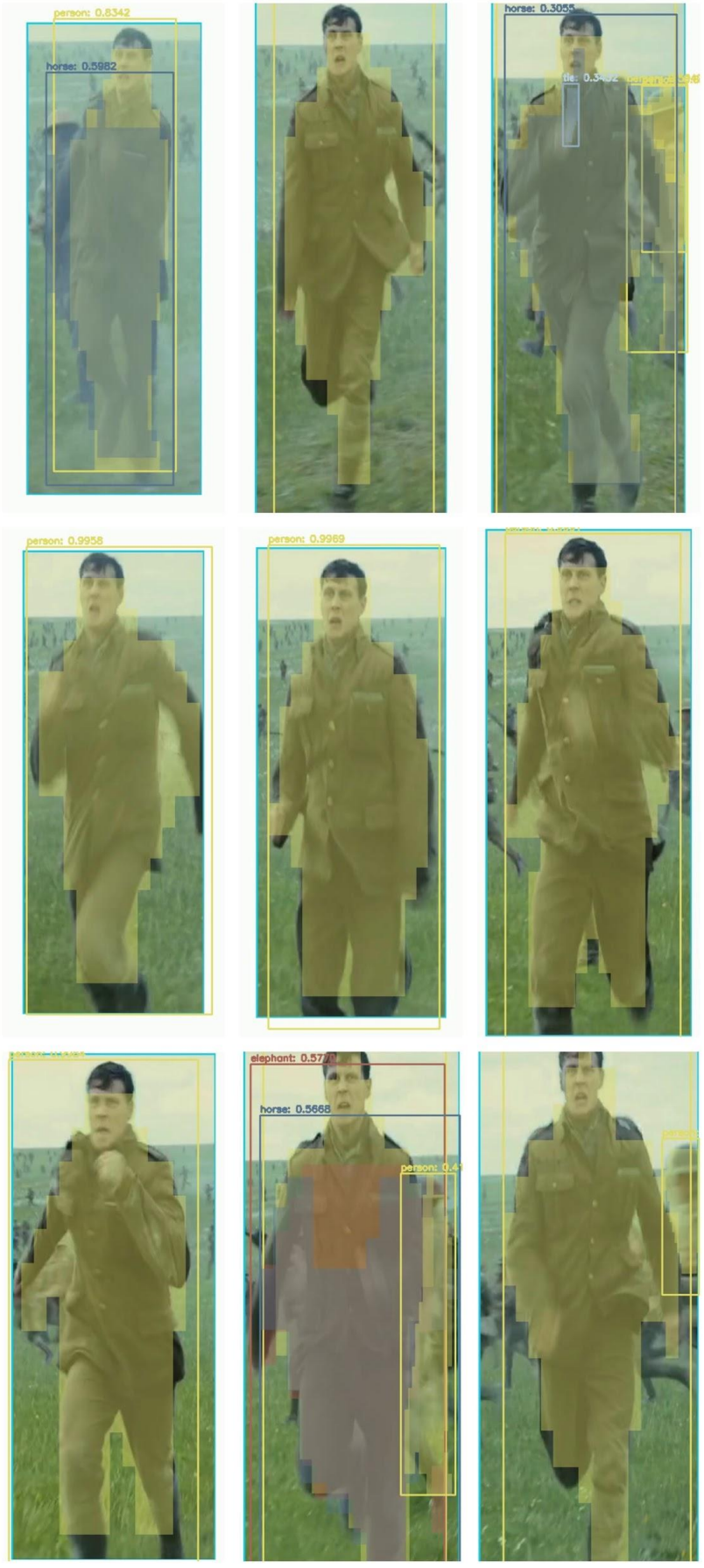
Pirmasis algoritmas bando suklasifikuoti kiekvieną pikselį ir, naudodamiesi jais, mes galime iškirpti asmens siluetą. Taigi, naudosime jau apmokytą Mask-RCNN modelį. Algoritmo veikimo principas:

1. Užkraunami išmokyto modelio svoriai.
2. Užkraunamas pats modelis.
3. Nuskaitomas vaizdo įrašas.
4. Imame pirmą/sekantį kadrą. Jei sekančio kadro nėra, išsaugome vaizdo įrašą su pažymėtais objektais.
5. Randame stačiakampius, gaubiančius norimus atpažinti objektus.
6. Imame pirmą/sekantį atpažintą objektą. Jei objektų nėra, einame į ketvirtą žingsnį.
7. Apskaičiuojame tikimybę, kad čia yra norimas objektas.
8. Klasifikuojame kiekvieną pikselį ir gauname objekto kaukę.
9. Einame į 6 žingsnį.

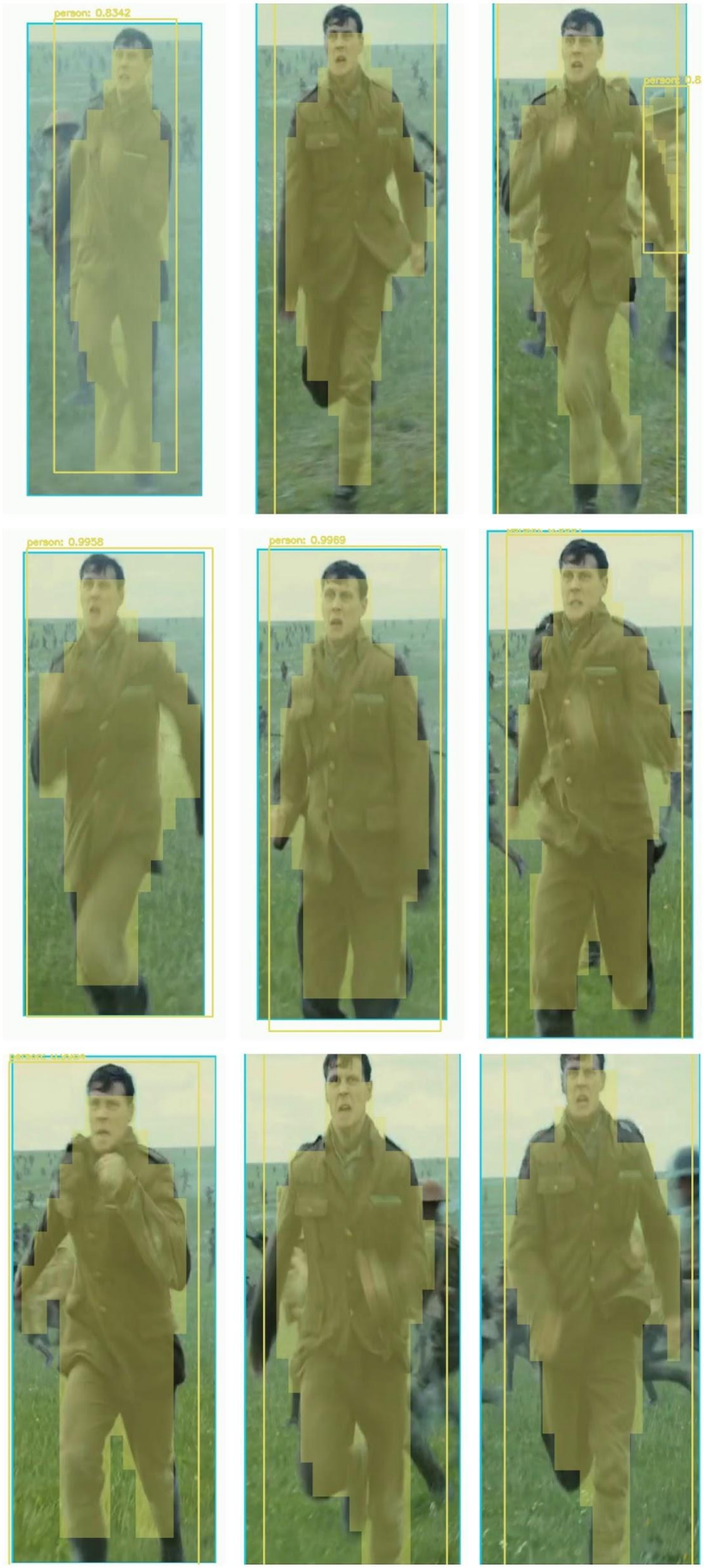
Šis modelis yra mokytas su COCO duomenų aibe, kurioje yra 80 objektų rūšių. Algoritmo implementacija reikalauja pasitikėjimo konstantos, kaip argumento. Ši konstanta nusako, kiek procentaliai algoritmas turi būti įsitikinęs, kad priskirti pikselį tam tikrai klasei. Testas padarytas su trim variacijomis - 0.25, 0.5, 0.75 . Rezultatus galima pamatyti 22, 23, 24 pav.



pav 22. Rezultatas su pasitikėjimo konstantą - 0.5



pav 23. Rezultatas su pasitikėjimo konstantą - 0.25



pav 24. Rezultatas su pasitikėjimo konstantą - 0.75

Iš rezultatų galima teigti, kad didesnė pasitikėjimo konstanta duoda tikslesnius rezultatus. Pirmuose dviejuose bandymuose yra aptinkami objektai, kurių iš tikrųjų nėra: tokie kaip arklys arba dramblys. Tačiau netgi su aukšta pasitikėjimo konstanta, kaukė nėra tiksli, nes kai kurios personažo dalys nėra teisingai klasifikuojamos. Pavyzdžiui, plaukai ne patenka į kaukę visuose kadruose. Tai rezultatas, su kuriuo negalima tęsti eksperimento, nes animacijos generavimas bus visiškai netikslus.

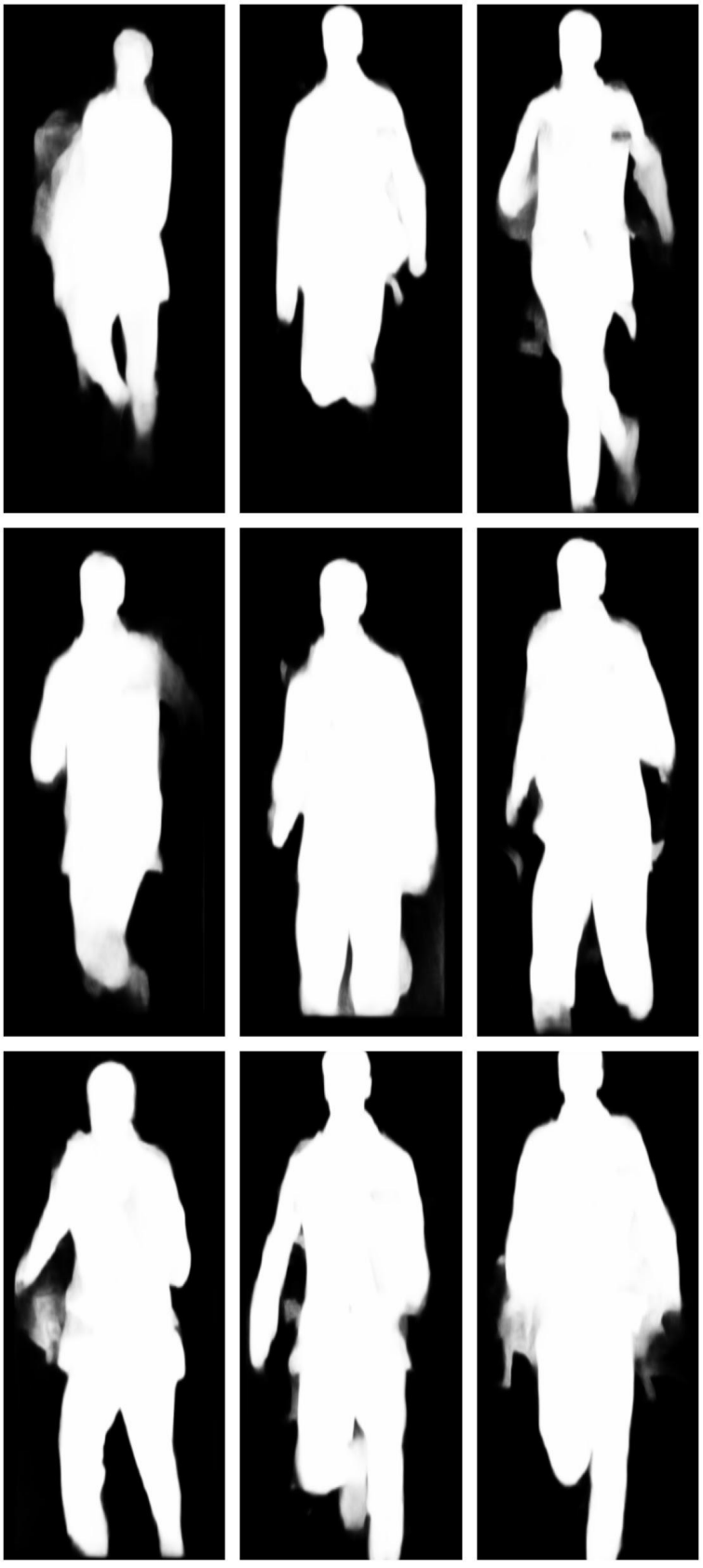
7.1.3.2 U2Net eksperimentas

U2Net [QZH20] tinklas originaliai buvo sukurtas rasti ir išskirti pagrindinio objekto kontūrus(angl. Salient Object Detection). Juos radus, galima lengvai pašalinti foną.

Algoritmo veikimo principas:

1. Užkraunami išmokyto modelio svoriai.
2. Užkraunamas pats modelis.
3. Nuskaitomas vaizdo įrašas.
4. Imamas pirmas/sekantis kadras. Jei kadro nėra, vaizdo įrašas išsaugomas su pažymėtais objektais.
5. Randamas pagrindinio objekto kontūras.
6. Kadras nuspalvinamas juodai, išskyrus tuos pikselius, kurie patenka į kontūrą.
7. Einama į 4 žingsnį.

Verta paminėti, kad šis tinklas yra sukurtas bendriniams atvejams. Tai reiškia, kad jis nebuvo mokamas, kad specifiškai atpažintų mašiną, žmogų ir t.t.. Šis algoritmas išskiria pagrindinį objektą kadre ir ieško būtent jo kontūrų. Tai sukeltų problemų jei paiešką darytume pilname kadre, kadangi mūsų personažas ne visados būna pagrindinis objektas jame. Tačiau anksčiau minėtos paruošimo procedūros užtikrina, kad algoritmas bandys atpažinti būtent norimą personažą. Rezultatus galima pamatyti 25, 26 pav.



pav 25. U2net rasti kontūrai.

pav. 26. Personažas su išimtu fonu.

Iš paveikslėlių matome, kad personažo kontūrai yra išimami visai tiksliai. Aišku, yra ir netikslumų, tokių, kaip fono detalės, prikibusios prie personažo arba kažkuri personažo dalis nepatenka į kontūrą. Tačiau didžioji dalis iškirpto personažo yra tiksli. Taigi, tolimesniam eksperimentui naudosime U2net algoritmo rezultatą.

Taigi, šių eksperimentu metu, mes paėmėm vaizdo įrašą ir sudarėm tinkamo formato vaizdo įrašą (angl. driving video). Turėdami šį vaizdo įrašą, galime pradėti eksperimentus su “First order motion” modeliu.

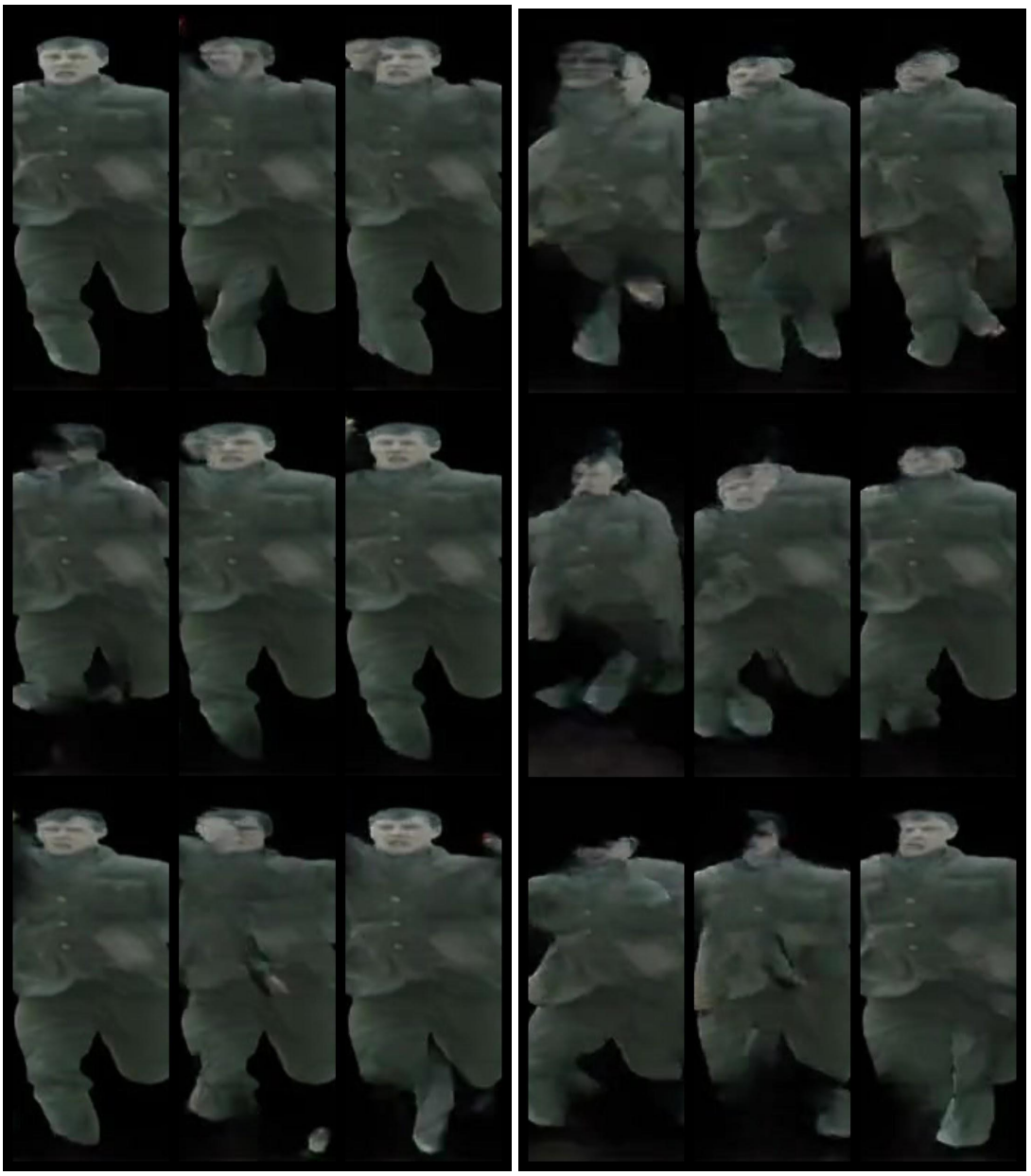
7.1.4 Animacijos kūrimas

“First order motion” modelio demonstracinė versija yra patalpinta “Google colab” platformoje. Naudojantis ja ,galima įkelti savo duomenis bei pabandyti sukurti animaciją. Demonstracinėje versijoje galima pasirinkti 7 skirtingas svorių konfiguracijas. Jos padarytos mokant modelį ant tam tikros duomenų aibės. Iš šių 7 aibių testai bus daromi tik su tomis duomenų aibėmis, kuriose personažai matomi pilnai kadre. Tai reiškia, kad eksperimentas bus daromas su dviem aibės “Tai-Chi” bei “Fashion Video Dataset”. Šiuose testuose bus bandoma išsiaiškinti ar “First order motion” modelis, išmokytas ant panašaus tipo aibės, sugeba prisitaikyti prie duomenų, kurie iš principo yra panašūs(kadre yra pilnai matomas žmogus), tačiau tuo pačiu ir skirtingi. Minėtose duomenų aibėse vaizdas yra labai švarus, fonas paprastas, o tuo tarpu mūsų vaizdo įraše yra tokie niuansai, kaip dūmai, mažėjantis/didėjantis personažas, kartais fonas susilieja su personažu.

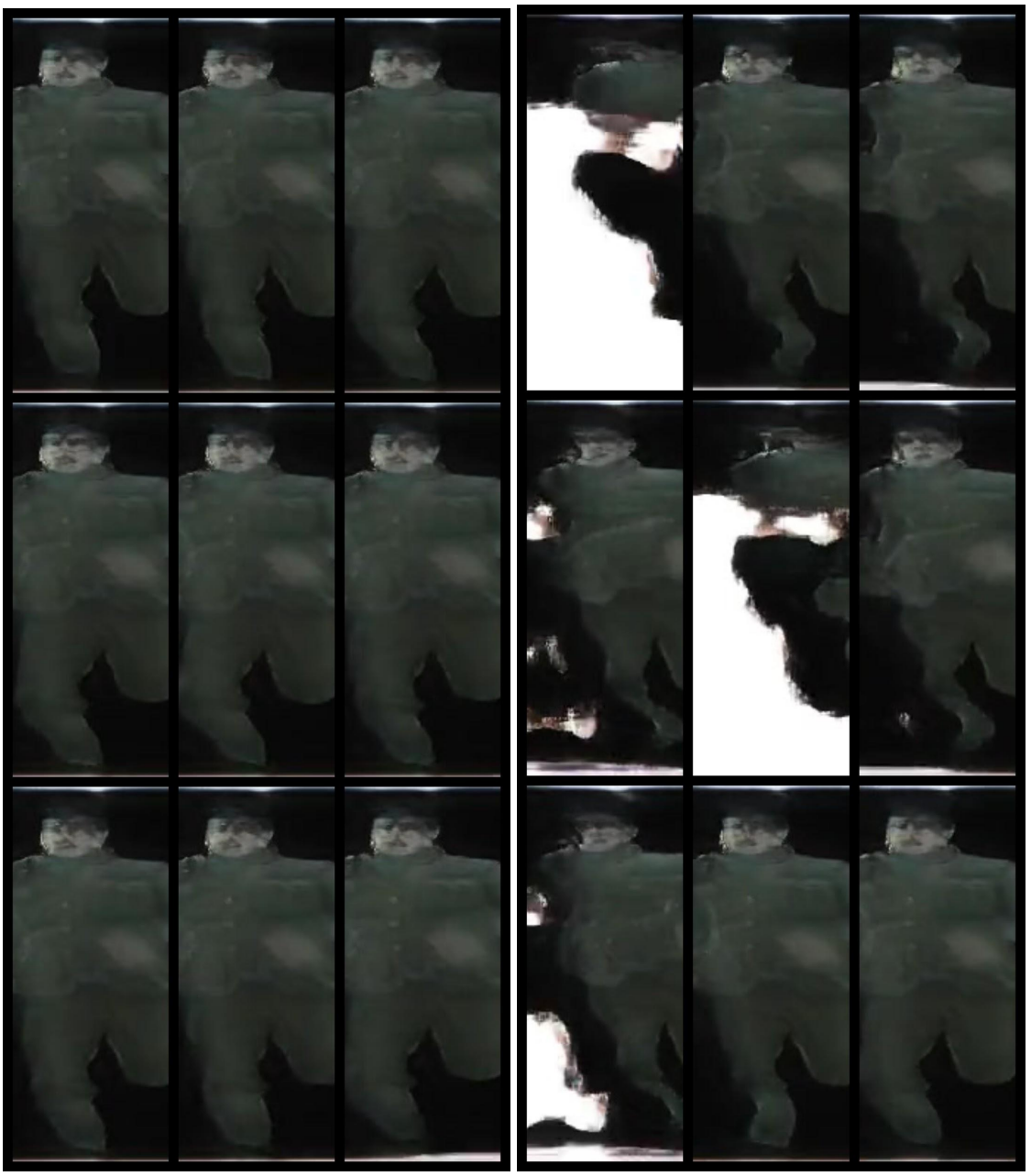
Kad pažiūrėti ar algoritmas sugeba veikti, ant kitos aibės nei yra mokytas, supaprastinsim darbą ir pažiūrėsim kaip modelis generuoja animaciją iš kadro, kuris yra išimtas iš minėtos animacijos. Taigi, kaip nuotrauką (angl. source image) naudosime kadrą iš įrašo (pav. 27).

pav 27. Įvesties nuotrauka (angl. source image)

Šiame eksperimente yra du kintamieji: svoriai ir koordinačių generavimo stilius. Svoriai bus paimti iš jau išmokyto modelio arba iš “Fashion video dataset” arba iš “Tai-chi”. Koordinačių generavimo stiliai yra 2: reliatyvus ir absoliutus. Reliativiame generavime bandoma animaciją objektui sugeneruoti nekeičiant jo dydžio, o absoliučiam, objektas, kuris animuojamas yra padidinamas/praplečiamas priklausomai nuo vaizdo įraše esančio personažo proporcijų. Taigi iš viso bus daromi 4 eksperimentai. Rezultatai pavaizduoti paveikslėliuose 28 ir 29.



pav. 28. Animacijos generavimas naudojantis "Tai-chi" duomenų aibės svoriais (kairėje reliatyvios koordinatės, dešinėje absoliučios)



pav. 29. Animacijos generavimas, naudojantis "Fashion video dataset" duomenų aibės svoriais. (kairėje reliatyvios koordinatės, dešinėje absoliučios)

Kad išmatuoti algoritmo rezultatus, bus naudojamos dvi pagrindinės metrikos : AKD ir MKR, kurios plačiau aprašytos 5.2.2.2 skyreli.

	Tai-chi	Fashion
Reliatyvios koordinatės	21.45	20.13
Absoliučios koordinatės	22.01	31.55

lentelė 6. Vidutinis esminių taškų atstumas (angl. Average keypoint distance) tarp originalaus ir animuoto vaizdo įrašo)

	Tai-chi	Fashion
Realityvios koordinatės	0.25	0.16
Absoliučios koordinatės	0.26	0.33

lentelė 7. Procentas prarastų esminių taškų (angl. Missing keypoint rate) tarp originalaus ir animuoto vaizdo įrašo)

Remiantis rezultatais galima teigti, kad geriausiai pasirodė modelis, kuris buvo mokytas su “Fashion video dataset” svoriais ir vaizdo įrašas generuojamas absoliučiomis koordinatėmis (išsaugant vaizdo įraše esančio personažo proporcijas). Rezultatas toli gražu neprimena fotorealistiškos žmogaus animacijos. Verta atkreipti dėmesį, kad modelio autoriai yra iškėlė problemą dėl animacijos vertinimo. Analizuojant vaizdo medžiagą. galima pamatyti, kad, nors skaičiai taip nerodo, tiksliausiai animacija buvo kuriama “Tai-chi” aibės mokytu modeliu su realityviom koordinatėmis. Šioje modelio konfigūracijoje buvo matomas tam tikrų kūno dalių judėjimas. Tai gali simbolizuoti, kad, nors formatas ir skiriasi, algoritmas mokytas su žmonėmis, kurie atlieka sudėtingus judesius, suteikė algoritmui abstraktų supratimą apie žmogaus judėjimą.

7.3 Antrasis eksperimentas

Iš pirmojo eksperimento rezultatų galima teigti, kad norint kokybiškai sukurti animaciją, tinklą reikia mokyti ant tam tikrų duomenų. Tam, kad gauti tokius duomenys, anksčiau aprašyti algoritmai bus taikomi ant viso filmo, taip sukuriant duomenų aibę, kuria naudojantis bus galima išmokyti tinklą.

7.3.1 Duomenų aibės paruošimas

Duomenų aibė bus generuojama iš minėto filmo “1917”. Šis filmas susideda iš 171312 kadrų (1 valanda ir 58 minučių trukmės). Tam, kad gauti tinkamą medžiagą tinklo mokymui, bus taikomos pirmame eksperimente aprašytos procedūros.

7.3.1.1 Epizodų radimas

Epizodais vadinsime kadrų rinkinį, kuriame yra x kadrų iš eilės, kuriuose yra aptinkamas žmogus. “FairMot” žmonių atpažinimo ir sekimo algoritmui, kaip įvestį bus naudojamas visas filmas. Algoritmas modifikuotas taip, kad vietoj vaizdinės išvesties gražintu JSON formato dokumentą, kurioje bus saugomi įrašai tokiu formatu:

1. Kadro numeris
2. Sekamo asmens ID
3. Stačiakampio koordinatės, kurios nusako kurioje kadro vietoje yra aptiktas žmogus.

Turint išvesties failą, yra sugeneruojamas epizodų sąrašas. Tai yra padaroma įteruojant per JSON failą ir tikrinant šalia esančius kadrus, kuriuose yra atpažintas tas pats asmuo. Po šios procedūros yra gauti 36143 epizodai. Verta paminėti, kad didelį epizodų skaičių lemia kelios aplinkybės:

- Pasirinkto filmo tematika. Šis filmas yra unikalus tuo, kad jisai yra nufilmuotas pastoviai sekant personažus. Todėl filme yra retas kadras, kuriame nebūtų asmuo.
- FairMot tinklas turi trumpą atmintį ir gali tam pačiam žmogui priskirti jau kitą ID numerį, jei, pvz. žmogų trumpam užstoja koks nors objektas.
- Yra daug filmo vietų, kuriose yra keli asmenys viename kadre. Tinklas juos atskiria, todėl kai kuriose filmo vietose vienam kadre gali būti sekami 5 atskiri asmenys.

Kadangi pagrindinė šio darbo užduotis yra sugeneruoti žmogaus animaciją, epizodai, kurie yra labai trumpi, nėra naudingi. Todėl yra nuspręsta epizodus prafiltruoti ir tolimesniam darbui naudoti tik tuos, kurie yra minimum 30 kadrų ilgio (~1 sekundė). Po filtravimo epizodų lieka 193.

7.3.1.2 Epizodų pavertimas į vaizdo įrašus

Sekantis žingsnis eksperimente yra iš epizodų sukurti vaizdo įrašus. Visų pirma, visas filmas yra išskaidomas į kadrus, t.y. vietoj mp4 formato video failo yra gaunama 171312 png formato nuotraukų. Naudojantis atfiltruotų epizodų informacija yra iškerpami kiekviename epizode atpažinti asmenys ir iškirpti kadrai yra patalpinami aplankuose, kurie simbolizuoja epizodo numerį. Kadangi vaizdo įrašą galima sukonstruoti tik tuo atveju kai kadrai yra to pačio dydžio, yra taikoma pirmame eksperimente naudota procedūra:

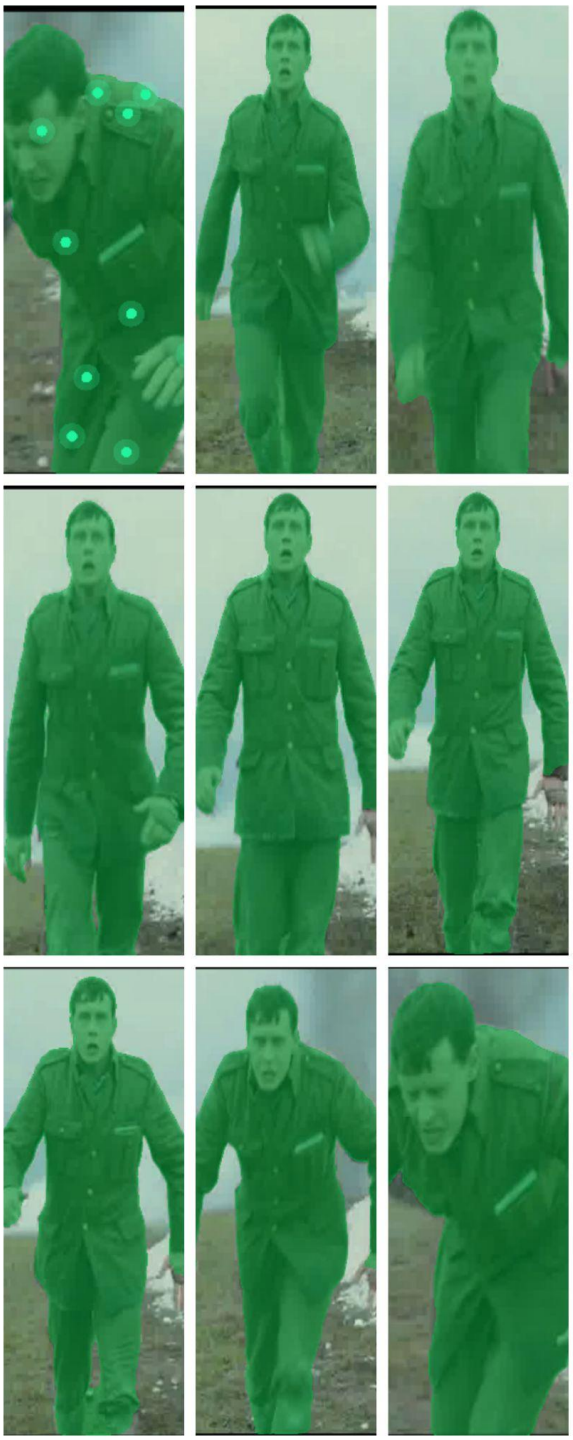
1. Įteruojama per atskiro epizodo kadrus ir išrenkamas pats didžiausias kadras
2. Antra kartą įteruojama per atitinkamo epizodo kadrus ir kiekvienam kadrui yra pridedamas tokio dydžio rėmas, kad atitiktų didžiausio kadro matmenis.

Taigi, kiekvieno epizodo kadrai dabar yra tos pačios dimensijos, ko pasekoje galima sukonstruoti vaizdo įrašą, kuriame asmuo yra viduryje.

7.3.1.3 Fono pašalinimas

Sekantis žingsnis eksperimente yra pašalinti foną, paliekant jame tik personažą. Šiam procesui buvo panaudotas U-2net tinklas. Tačiau tinklas sunkiai sugebėjo atpažinti personažą, todėl daugelyje įrašų fonas susilieja su personažu. Tam, kad patikrinti ant kiek gerai buvo iškirptas personažas, šie įrašai buvo duoti kaip įvestis “FairMot” tinklui. Tik 6 įrašuose iš 193, buvo aptiktas asmuo kiekviename kadre. Atsižvelgiant į netikslų atpažinimą nuspręsta ieškoti kito būdo.

Tam, kad pašalinti foną, buvo pasirinktas pusiau automatinis būdas. Naudojantis programine įranga “Green Screen AI”, rankiniu būdu buvo pašalintas fonas. Šioje programoje yra įkraunamas vaizdo įrašas ir rankiniu būdu yra žymimi regionai kadre, kurie turi būti sekami. Sužymėjus tinkamus regionus, programa automatiškai juos seka visame likusiame vaizdo įraše. Tuomet programa automatiškai uždeda vienspalvį foną ir palieka kadre tik sekamą objektą. Buvo išimti įrašai, kuriuose fono nustatyti praktiškai neįmanoma. Tokie įrašai, kuriuose aptiktas personažas yra labai nutolęs nuo kameros objektyvo, ko pasekoje asmuo yra labai susiliejes su aplinka arba įrašai, kuriuose personažas yra labai tamsiame fone, ko pasekoje sunku aptikti, kur baigiasi pats personažas ir prasideda fonas. Po filtravimo liko 126 įrašai. Šiuose įrašuose “Fairmot” tinklas atpažįsta personažą kiekviename kadre, todėl neuroninio tinklo mokymo proceduroje bus naudojama būtent ši aibė.



pav. 30. Sekamo objekto aptikimas naudojant "Green Screen AI" programinę įrangą. Viršutiniame kairiajame kampe matosi ranką pažymėti pagrindiniai regionai, kuriais naudojantis objektas yra sekamas visame įraše.

pav. 31. Fono išėmimo rezultatas

7.3.2 Neuroninio tinklo mokymas

Turint aibę, galima pradėti tinklo mokymą. Iš 126 įrašų aibės, 70% yra priskiriama mokymo aibei ir 30% priskiriama testinei aibei. Taip pat visų įrašų dydis yra suvienodinamas iki 384x384 dimensijos. Mokymas bus vykdomas “Google Colab” platformoje. Šioje platformoje yra suteikiama Tesla K80 vaizdo plokštė, kurią ir naudosime mokymui. Kadangi mokymo laikas yra labai ilgas (~6 paros), bus panaudota autorių rekomenduojama konfigūracija. Neuroninio tinklo mokymas vyksta tokiu principu:

1. Imamas vaizdo įrašas.
2. Iš kadro randami pagrindiniai taškai.
3. Iš viso vaizdo įrašo yra gaunama judesio informacija.

4. Tinklas bando atkurti vaizdo įrašą remiantis pagrindiniais taškais bei judesio informacija.

Bus išbandomos dvi mokymo konfigūracijos: standartinė ir mokymas su deformacija (angl. Animation via disentanglement). Mokymas su deformacija sprendžia problemą, kuomet tinklas rekonstruodamas vaizdo įrašus žmogaus formą įsisavina iš vaizdo įrašo, o ne įvesties kadro. Šis rezultatas pasiekiamas atsitiktinai deformuojant sekantį kadrą. Tai priverčia tinklą, asmens formą išmokti iš įvesties kadro, o ne vaizdo įrašo. Taipogi verta paminėti, kad mokymas su deformacija yra vykdomas naudojantis standartinio mokymo metu gautais svoriais, t.y. tai yra ne atskiras, o papildomas mokymas.

7.3.3 Neuroninio tinklo mokymo rezultatas

Tinklo mokymas užtruko 130 valandas. Papildomai mokymas su deformacija užtruko 30 valandų. Tam, kad įvertinti neuroninio tinklo rezultatus, bus naudojamos tos pačios metrikos, kaip pirmame eksperimente. Tam, kad patikrinti kaip tinklas sugeba rekonstruoti vaizdo įrašą, testas bus atliekamas taip:

1. Imamas sekantis įrašas iš testinės aibės. Jei įrašų nėra einama į 6 žingsnį.
2. Imamas pirmas kadras ir gaunama judesio informacija.
3. Tinklas rekonstruoja vaizdo įrašą.
4. Apskaičiuojamos metrikos tarp originalaus įrašo ir sugeneruoto.
5. Einama į pirmąjį žingsnį.
6. Apskaičiuojamas visų gautų metrikų vidurkis

Verta paminėti, kad standartiniame mokyme yra naudojamas tik absoliučių koordinačių generavimo stilius, dėl to, kad yra žinoma, kad personažas, kuris yra pirmame kadre(source image), yra tas pats, kuris bus su animuotame vaizde. Mokyme su deformacija, koordinačių stilius yra nei absoliutus, nei reliatyvus, kadangi yra atsitiktinai deformuojamas.

	Standartinis mokymas	Mokymas su deformacija
AKD	19.6895	19.4420
MKR	0.1348	0.1388

lentelė 8. Suvidurkintos metrikos gautos lyginant originalius įrašus su sugeneruotais

Rezultatas yra ženkliai geresnis nei pirmame eksperimente. Galima teigti, kad tinklo mokymas ant specifinio tipo duomenų iš ties padėjo tinklui geriau rekonstruoti vaizdo įrašus.

7.3.4 Animacijos kūrimas

Viršuje pateikti rezultatai nenusako, kaip algoritmas sugeba rekonstruoti vaizdo įrašą, kai įvesties kadras nėra iš to pačio vaizdo įrašo. Tam, kad tai patikrinti reikia ištestuoti algoritmą su įvesties kadru, kuris nėra išimtas iš vaizdo įrašo. Šiam eksperimentui buvo atrinkti trys vaizdo įrašai, kuriuose personažas atlieka judesius iš matymo kampų: priekio, šono ir nugaros. Įvesties kadrai taip pat bus trys asmenys, kurie stovi iš atitinkamų kampų.

pav. 32 Įvesties kadrai.

pav 33. Vaizdo įrašai naudojami animacijoje

Eksperimente bus 2 kintamieji: tinklo svoriai ir koordinačių generavimo stilius. Kadangi tinklas buvo mokomas naudojantis dviem konfiguracijomis (su deformacija ir be), testas bus atliekamas su atitinkamais metodais gautais svoriais. Taip pat verta paminėti, kad yra pridėtas ir naujas koordinačių generavimo stilius - deformacijos. Šis koordinačių generavimo stilius yra galimas tik su tinklo svoriais, kurie buvo mokyti su deformacija. Naudosime tas pačias metrikas, kaip ir pirmame eksperimente. Sugeneravus realistišką animaciją, ją galimą būtų įkirpti į originalų vaizdo įrašą, taip pasiekiant personažo pakeitima.

	Standartinis mokymas/Absoliučios koordinatės	Standartinis mokymas/Realiatyvios koordinatės	Mokymas su deformacija/Absoliučios koordinatės	Mokymas su deformacija/Realiatyvios koordinatės	Mokymas su deformacija/deformuotos koordinatės
AKD	15.210	23.968	13.536	23.008	16.002
MKR	0.093	0.060	0.085	0.054	0.080

lentelė 9. Sukurtos animacijos rezultatas (personažas iš nugaros)

	Standartinis mokymas/Absoliučios koordinatės	Standartinis mokymas/Realiatyvios koordinatės	Mokymas su deformacija/Absoliučios koordinatės	Mokymas su deformacija/Realiatyvios koordinatės	Mokymas su deformacija/deformuotos koordinatės
AKD	16.941	52.539	18.167	49.976	22.0321
MKR	0.058	0.246	0.051	0.216	0.106

lentelė 10. Sukurtos animacijos rezultatas (personažas iš priekio)

	Standartinis mokymas/Absoliučios koordinatės	Standartinis mokymas/Realiatyvios koordinatės	Mokymas su deformacija/Absoliučios koordinatės	Mokymas su deformacija/Realiatyvios koordinatės	Mokymas su deformacija/deformuotos koordinatės
AKD	37.422	58.744	36.924	69.923	37.660
MKR	0.159	0.325	0.158	0.324	0.187

lentelė 11. Sukurtos animacijos rezultatas (personažas iš šono)

Remiantis rezultatais galima padaryti šias išvadas:

1. Jokia svorių/generavimo kombinacija nebuvo vienareikšmiškai geriausia. Bendrai geriausius rezultatus pavyko gauti su deformacijos mokymo svoriais ir absoliučiomis

koordinatėmis. Tai leidžia teigti, kad geriausios konfiguracijos pasirinkimas priklauso nuo duomenų.

2. Rezultatai gerokai blogesni nei autorių gauti rezultatai su paprastesnėmis duomenų aibėmis. Pirmoji priežastis yra duomenų kiekis: autorių pasirinktos duomenų aibės turi kaip minimum dvigubai daugiau įrašų. Antroji priežastis yra duomenų sudėtingumas. Epizodai gauti iš filmo “1917” yra labai skirtingi. Asmenys yra iškerpami skirtingais atstumais nuo kameros, o tai reiškia, kad įrašo kokybę, suvienodinus rezoliuciją, yra skirtinga. Patys personažai tarp kadrų ženkliai skiriasi, tiek fiziniais bruožais, tiek aprangomis. Galiausiai kadruose asmuo ne visada matomas pilnai.
3. Rezultatai labai priklauso nuo įvesties įrašo. Tai galima pamatyti lyginant AKD metrikas tarp eksperimentu iš priekio ir eksperimentu iš galo. Įrašas iš priekio prasideda personažui ištiesinant nugarą. Tinklas sunkiai galėjo suprasti šį judesį, todėl esminių taškų atstumas buvo vidutiniškai ženkliai didesnis nei generuojant animaciją iš nugaros, kuomet iš pradinio kadro matosi pilnas asmens kūnas. Blogiausios AKD metrikos yra eksperimente, kur personažas iš šono. Tai galėjo lemti faktas, kad asmuo, esantis vaizdo įraše, buvo su šalmu, kuprine ir šautuvu, ko pasekoje tinklas nesugebėjo tinkamai perteikti žmogaus stovėsenos.
4. Geriausios svorių/generavimo kombinacijos pasiekė geresnius rezultatus, negu pirmajame eksperimente. Taip pat verta paminėti, kad atvirkščiai, negu pirmame eksperimente, pirmas kadras nėra iš to pačio įrašo. Tai reiškia, kad mokymas, nors ir su sudėtingais duomenimis, davė naudą.

7.4 Išvados

Eksperimentų metu buvo išnagrinėtas “First order motion” modelis, kuris sugeba generuoti animaciją tiesiog iš nuotraukos ir duoto vaizdo įrašo. Pirmojo eksperimento metu buvo įsitikinta, kad modelis turi būti mokytas su specifiniais duomenimis, tam, kad galėtų sugeneruoti realistišką rezultatą.Iš gautų rezultatų matoma, kad tinklo animacijos galimybės labai priklauso, nuo duomenų aibės, su kuria buvo mokoma. Iš pažiūros, tiek vaizdo įrašas, kuris buvo generuojamas, tiek įrašai, esantys duomenų aibėje, su kuria autoriai atliko mokymą yra panašūs (pilnai matomas žmogus atlieka nesudėtingus veiksmus). Tačiau iš vaizdo generacijos rezultato galima teigti, kad reikalingi duomenys, kurie labiau atitiktų animuojamo personažo tematiką.

Labai svarbų vaidmenį šiame tyrime atlieka duomenys. Autorių kurti modeliai buvo mokyti bei testuoti ant aibių, kurios visos yra labai vizualiai panašios ir asmenys stovi tose

pačiose vietose, bei atlieka labai panašius veiksmus. Tyrimo metu, buvo sukurta duomenų aibė, kuri dalinai automatinėmis procedūromis buvo sugeneruota iš filmo, kuriame personažai yra skirtinguose kadro vietose, skirtingai nutolę nuo objektyvo ir matomi iš skirtingų kampų. Naudojantis “FairMot” neuroniniu tinklu buvo išspręsta asmens pozicijos kadre problema. Naudojantis “Green Screen AI” programine įranga, buvo išimtas fonas, taip padarant duomenis panašesnius į tuos, kuriuos naudoja autoriai. Pilnai automatizuoti proceso nepavyko dėl fono išėmimo problemos. U2net tinklas nesugebėjo tinkamai iškirpti fono iš didžiosios dalies gautų vaizdo įrašų.

Antrojo eksperimento metu, “First order motion” modelis, buvo mokomas ant naujai sugeneruotos duomenų aibės. Tačiau tiek mokymo metu gautos metrikos, tiek metrikos, gautos animuojant įrašą su įraše neegzistuojančiu asmeniu, neprilygsta autorių gautiems rezultatams. Duomenų kiekis bei faktas, kad duomenys tarpusavyje yra labai skirtingi, lėmė, kad tinklas, nesugebėjo animacijos generuoti kokybiškai. Atsižvelgus į pirmo ir antro eksperimento metu gautus rezultatus, galima teigti, kad mokymo metu gauti svoriai lėmė geresnius rezultatus, nei autorių gauti svoriai gauti mokant su ženkliai didesne ir tarpusavyje panašia aibe.

Šiame darbe buvo parodyta, koks galėtų būti procesas, norint daryti asmenų sukeitimą sudėtingoje vaizdo įrašo aplinkoje. Verta paminėti, kad tai yra pirmasis mokslinis darbas, kuriame nuotraukos animacija yra nagrinėjama sudėtingos vaizdo medžiagos kontekste. Jei antrasis eksperimentas, būtų pasiekęs autorių rezultatus, tai būtų leide filme sukeisti personažus, sugeneruotą animaciją įklijuojant į originalų įrašą. Apibendrinant galima teigti, kad nuotraukos animacijos kūrimo algoritmai yra labai priklausomi nuo duomenų aibės. Jų pritaikymas, norint sukeisti asmenis vaizdo įraše yra labai ribotas. Tiksliam asmenų sukeitimui reikalinga didelė duomenų aibė, kuriame asmenys yra vienodi ir daro panašius veiksmus.

Literatūra

[DO19] D. O'Sullivan. *Congress to investigate deepfakes as doctored Pelosi video causes stir*. CNN 2019.

[RP19] R. Petrana. *Harrison Ford is the star of Solo: A Star Wars Story thanks to deepfake technology*. Polygon, 2019.

[DJ19] D. Jesse. *Chinese Deepfake App Zao Goes Viral, Faces Immediate Criticism Over User Data And Security Policy*. 2019.

[SLT19a] A. Siarohin, S. Lathuiliere, *S. Tulyakov, E. Ricci, N. Sebe. First Order Motion Model for Image Animation*. 2019.

[CHZ14] C. Cao, Q. Hou, ir K. Zhou. *Displaced dynamic expression regression for real-time facial tracking and animation*. 2014.

[GPM14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.Courville ir Y. Bengio. *Generative adversarial nets*. 2014.

[DM14] D. P. Kingma ir M. Welling. *Auto-encoding variational bayes*. 2014.

[Gir15] R. Girshick *Fast R-CNN.* 2015.

[SLT19b] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, ir N. Sebe. *Animating arbitrary objects via deep motion transfer*. 2019.

[GDD15] R. Girshick, J.Donahue, T. Darrell ir J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5).* 2015.

[USG13] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers ir A.W.M. Smeulders. *Selective Search for Object Recognition*. 2013.

[RHG15] S. Ren, K. He, R. Girschick ir Jian Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015.

[HZR14] K. He, X. Zhang, S. Ren, irJ. Sun. *Spatial pyramid pooling in deep convolutional networks for visual recognition*. 2014.

[EHM18] P.Esser, J. Haux, T. Milbich, B. Ommer. *Towards Learning a Realistic Rendering of Human Behavior*. 2018.

[LMB14] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, ir P. Dollár. *Microsoft COCO: Common Objects in Context.* 2014.

[WKZ18] . Wiles, S. Koepke, ir A. Zisserman. *X2face: A network for controlling face generation using images, audio, and pose codes*. 2018.

[CSW17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. 2017.

[NCZ17] A. Nagrani, J. S. Chung, and A. Zisserman. *Voxceleb: a large-scale speaker identification dataset*. 2017.

[DSG12] H. Dibeklioglu, A. A. Salah, irT. Gevers. *Are you really smiling at me? spontaneous versus posed enjoyment smiles*. 2012.

[EFL17] F. Ebert, C. Finn, A. X Lee, ir S. Levine. *Self-supervised visual planning with temporal skip connections*. 2017.

[BT17] A. Bulat and G. Tzimiropoulos. *How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks).* 2017.

[KGP17] K. He, G. Gkioxari, P. Dollar, R. Girschick. *Mask R-CNN.* 2017.

[YCX20] Y. Zhang, C. Wang, X. Wang, W.Zeng, W.Liu. *FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking 2020*.

[QZH20] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, Martin Jagersand. "*U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection" 2020.*