

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Lars Jesper Eller

Tarkvara videote automaatseks monteerimiseks
programmiga DaVinci Resolve

Bakalaureusetöö (9 EAP)

Juhendaja: Sven Aller

Tartu 2025

Tarkvara videote automaatseks monteerimiseks programmiga

DaVinci Resolve

Lühikokkuvõte

Käesoleva bakalaureusetöö eesmärk on analüüsida masinõppetööriistade potentsiaali filminduse valdkonnas ning seejärel luua tarkvaraline lahendus mõnda sündmust või ajaperioodi kokkuvõtva video kokku monteerimise lihtsustamiseks. Kasutaja sisestab vajalikud heli- ning videofailid ja sätestab parameetrid. Seejärel kasutab programm Pythoni heli- ja videotöötlusteeki librosa ja Mediapipe, et luua DaVinci Resolve tarkvara siseseid monteerimistoiminguid automatiseeriva skripti. Skripti abil loodud video klipid on lõigatud objektituvastuse ning helitempotuvastuse tulemuste põhjal.

Võtmesõnad

Tarkvara, DaVinci Resolve, Python, librosa, Mediapipe, masinõpe, skriptimine, helitöötlus, videotöötlus, videote monteerimine

CERCS:

P175 Informaatika, süsteemiteooria

Software for automatically editing videos using DaVinci Resolve

Abstract

This Bachelor's thesis aims to analyse the viability of machine learning tools in film creation and create software that simplifies editing a recapitulative video of an event or period. The user inputs the necessary audio and video files and parameters. The program then uses Python's librosa audio analysis library and Mediapipe video analysis library to create a script which automates the editing process within DaVinci Resolve. Clips within the video created with a script are cut according to the object detection and audio tempo detection results.

Keywords

Software, DaVinci Resolve, Python, librosa, Mediapipe, machine learning, scripting, audio analysis, video analysis, video editing

CERCS:

P175 Informatics, systems theory

Sisukord

Sissejuhatus.....	3
1. Töö teoreetilised alused.....	4
1.1 Filmide ja videote monteerimine.....	4
1.2 Masinõpe filmide ja videote monteerimise valdkonnas.....	4
2. Tehnoloogilised valikud.....	6
2.1 DaVinci Resolve.....	6
2.2 Python.....	6
3. Tarkvara põhiohanded ja arhitektuur.....	8
3.1 Tarkvara nõuded.....	8
3.2 Tarkvara üldine arhitektuur ja tehisintellekti roll.....	8
3.3 Kasutajaliides ning konfiguratsioon.....	9
3.4 Helifailitöötlus.....	10
3.5 Videofailitöötlus.....	11
3.6 Skripti loomine.....	12
3.7 Tarkvara protsesside ressursimahukus.....	13
4. Tarkvara ja loodud skripti kasutamise kirjeldus.....	15
4.1 Liides ja kasutaja töövoog.....	15
4.2 Skripti kasutamine DaVinci Resolve monteerimistarkvaras.....	21
5. Tagasiside.....	22
6. Võimalikud edasiarendused.....	23
Kokkuvõte.....	25
Viidatud kirjandus.....	27
Lisad.....	30
I. Tarkvaranõuded.....	30
II. Kasutusjuhend.....	30
III. Katsetamise käigus kasutatud sülearvuti ja lauaarvuti tehniline spetsifikatsioon.....	32
IV. Helifailitöötluse ressursikulu katsete tulemused.....	32
V. Videofailitöötluse ressursikulu katsete tulemused.....	33
VI. Tagasisideküsitluse lisainfo ja küsimused.....	33
VII. Litsents.....	35

Sissejuhatus

Üheskoos filmitööstuse digitaliseerumisega ning isiklike arvutite kasutajasõbralikkuse ja ligipääsetavuse kasvuga on viimase mitmekümne aastaga arenenud välja uus infotehnoloogia haru: digitaalne monteerimine (Fernandes, 2024). Monteerimistarkvara mitmekesine ja pidevalt arenev maastik lubab igal huvilisel oma soovidele ja oskustele vastavalt kasutada sobivat programmi, millega saab oma isiklikus seadmes kokku monteerida video (Fernandes, 2024). Sellegipoolest on see protsess alati keeruline või ajakulukas - igal lihtkasutajal ei pruugi olla motivatsiooni ega aega, et õppida selgeks keerulisemad toimingud, mille najal saaks valmis monteerida soovitud video.

Bakalaureusetöö eesmärk on luua tarkvaraline lahendus mõnda sündmust või ajaperioodi kokkuvõtva video loomise lihtsustamiseks. Vajalike sisendandmete (sh üks kuni mitu videofaili, üks helifail, soovitud lisaparameetrid) alusel leitakse sisendfailidest heli tempo ning videote meeleolukaimad hetked, olgu need rõõmsad näod või ilusad hetked. Seejärel genereeritakse nende põhjal videot monteeriva skripti ning see väljastatakse kasutajale üheskoos väikese juhendiga.

Põhifunktsionaalsuste täitmiseks kasutab tarkvara masinõppel põhinevaid tööriistu. Need sammud tagavad, et lõplikus videos on kaadrid asjakohased ning muutuvad helifailiga rööbiti. Tarkvara kõrvaldab kasutaja jaoks tülikaimad ning ajakulukaimad sammud video loomise protsessis, jättes kasutajale vaid soovitud videoklippide ja taustaheli leidmise ning video omaduste (nt helitugevus, kaadrivahetuse sagedus) osas otsustamise vaeva.

Töö sisuline osa on jagatud seitsmeks osaks: uurimuse teoreetilised alused, tehnoloogilised valikud tarkvara loomisel, tarkvara arhitektuur, tarkvara põhinõuded ning kirjeldus, tarkvara tulemustele kogutud tagasiside ning võimalikud edasiarendused. Töö lisade hulka kuuluvad tarkvaranõuded, kasutusjuhend, katsetamise käigus kasutatud riistvara spetsifikatsioon ning ressursikulu mõõtmise tulemused.

Töös valmimisel on kasutatud tekstirobotit juba koostatud koodi töötlemiseks - tehisintellekti kasutamise olemust täpsustatakse alampeatükis 3.2 Tarkvara üldine arhitektuur ja tehisintellekti roll.

1. Töö teoreetilised alused

Peatükis antakse ülevaade filmide ja videote monteerimisest ning masinõppe ja tehisintellekti arenevast rollist selles valdkonnas.

1.1 Filmide ja videote monteerimine

Andmekaitse ja infoturbe leksikon defineerib monteerimist ISO 5127 rahvusvahelise standardi järgi kui “filmivõtete lõikamine ja kokkupanek filmi lõpliku eksemplari saamiseks” (AKIT). Kuigi filmide monteerimine sai tõepoolest alguse lihtsa filmilindi füüsilise lõikamise ning kokkukleepimisega, on sellesse protsessi filmikunsti arenedes lisandunud lai valik muid võimalusi. Heliefektid, üleminekud, tehnikud visuaalsed kujutised ning paljud teised monteerimisvõimalused on tänapäeval üldlevinud nii filmitööstuses kui ka amatöörvideotes. Tänu eelmise sajandi lõpus alanud digitaalsete monteerimissüsteemide võidukäigule on videote monteerimine tavainimesele kättesaadavam ja lihtsam kui kunagi varem (Fernandes, 2024). Saadaval on lai valik monteerimistarkvara, mis infotehnoloogia arengutega üheskoos on viimase paarikümne aastaga kasvanud võimekaks, kuid sellegipoolest jõukohaseks tööriistade pagasiks. Tänu valdkonna rajaleidjatele on isegi võimalik igal lihtkasutajal oma isiklikkus arvutis kasutada Oscareid pälvinud filmide monteerimisel kasutatud tarkvara (Waters, 2019). Juhul, kui mõni komplekssem tarkvara (näiteks Avid Media Composer või Adobe Premiere Pro) käib kasutajale üle jõu, on tal siiski võimalik viia läbi elementaarsemad videote kokkupanekud ning muusika või teksti lisamised näiteks TikToki või Instagrami platvormidesse sisse ehitatud monteerimistööriistadega (Fernandes, 2024). Monteerimine on selle uurimuse keskmes: arendatava tarkvara põhiline eesmärk (nii Pythoni arenduskeskkonnas kui ka DaVinci Resolve tarkvara jaoks skripti luues) on monteerida sisendklippide põhjal terviklik video.

1.2 Masinõppe filmide ja videote monteerimise valdkonnas

Nagu nii mõneski infotehnoloogia valdkonnas, on ka filmitööstuse jaoks loodud tarkvaralahendused saanud viimasel ajal masinõppel põhineva tehisintellekti jõul töötavaid uuendusi. Kendra Ruczak on oma artiklis on välja toodud kolm valdkonna rajaleidjat tehisintellekti implementeerimises (2023):

1. NVIDIA, mis ainuüksi ühel konverentsil avalikustas ligikaudu 20 uurimust tehisintellekti kasutusvõimalustest reaalsajas animeerimiseks ning visuaalsete efektide genereerimiseks.

2. Blackmagic Design, mis integreeris oma populaarse monteerimistarkvara DaVinci Resolve'i versiooni 18.5 laia sortimendi tehisintellektil põhinevaid tööriistu - nende hulgas virtuaalsete valgusallikate loomine, automaatne heliklippide sortimine ja videoklipis leiduvate elementide ja heliallikate tark isoleerimine.
3. Adobe, mille Premiere Pro monteerimistarkvara sisse on nüüd ehitatud masinõppel põhinev tekstipõhine monteerimissüsteem - monteerijatel on võimalik lugeda imporditud klippide põhjal koostatud ära kirja, mida muutes ja millest võtmesõnu noppides valmib video ilma täiendava sisendita.

Enamus välja arendatud lahendustest keskenduvadki visuaalsete efektide loomisele (Argaw et al., 2022) ning vajavad asjatundlikku sisendit monteerimist tundvalt inimeselt. Selle töö eesmärk on aga luua lahendus, mis keskendub videote tegelikule kokkupanekule ilma olulise sekkumiseta kasutajalt.

2. Tehnoloogilised valikud

Peatükis tutvustatakse DaVinci Resolve monteerimistarkvara ning selle asjakohaseid funktsioone, samuti Pythoni programmeerimiskeelt ning selle uurimuses kasutatavaid võimalusi.

2.1 DaVinci Resolve

DaVinci Resolve on monteerimistarkvara, mille juured peituvad 1980. aastate alguses, kui arendati esimene da Vinci Classic värviparandustarkvara (Shaw, n.d.). Esialgu oli Resolve, mille esimene täisversioon tuli müügile 2004. aastal, otse sisse ehitatud spetsiaalsesse riistvarasse, mis odavaimas konfiguratsioonis maksis 200 000 dollarit (BusinessWire, 2010). Tänapäevaks on saanud tänu uutele omanikele ja ulatuslikule laienemisele sellest monteerimistarkvarast tööstuse standard, mida korraga kasutavad nii maailmatasemel filmitootjad kui ka üle kahe miljoni lihtkasutaja, kes kasutavad tarkvara tasuta versiooni (Waters, 2019). Peale tehisintellektitööriistade toetab DaVinci Resolve ka skriptide - teise programmi poolt ilma kasutaja sekkumiseta täidetavate käskude jadade (AKIT) - kasutamist. See funktsionaalsus pärineb esialgselt DaVinci Resolve omanike Blackmagic Design teiselt, visuaalefektide loomiseks mõeldud tarkvaralt Fusion ning toetab Lua ja Python keeli (Radet). DaVinci Resolve Studio, mis on tarkvara tasuline versioon, suudab skripte jooksutada ka välimisest kaustast, seda võib käitada tummvarana (ingl *headless software* ehk ilma graafilise kasutajaliideseta) ning kasutades *DaVinciResolve.py* moodulit on võimalik läbi viia monteerimistoiminguid läbi välimise Pythoni keskkonna (Radet, n.d; Ramani, 2024). Selle töö hakul oli võimalik neid tasulise versiooni funktsioone kasutada ka tasuta versioonis, kuid need eemaldati selle uurimuse kirjutamise ajal (Ramani, 2024). Kuna ligipääsetavus on selle rakendusliku uurimuse üks nõuetest (vt alampeatükki 3.1 Tarkvara nõuded), jäi tugi Resolve Studio funktsioonidele esialgu välja selle uurimuse skoobist ning tarkvara väljundiks jäi skript, mida kasutaja saab ise Resolve tarkvaras hiljem käivitada. Käesolevas töös kasutatakse DaVinci Resolve monteerimistarkvara just selleks, et selle laia skriptimistuge kasutades luua videoid kokku paneva skripti loomise tarkvara.

2.2 Python

Python on kõrgetasemeline programmeerimiskeel, mis loodi 1990. aastatel Guido van Rossumi poolt ABC keele järeltulijaks (History and License, n.d.). Seda peetakse üheks maailma populaarseimaks keeleks - kasutajad hindavad kergelt selle lihtsust, tuge mitme programmeerimislähenemise jaoks ja laia teekide valikut. Teek (ingl *library*) on kogumik

valmis kirjutatud ja sisse laetavat koodi, mida teised arendajad saavad enda programmeerimiskeskkonnas kasutada ilma, et nad seda uuesti kirjutama peaksid (Jawahar et al., 2023). Python on saavutanud suurt edu andmeteadeuse ja tehisintellekti valdkondades, sest sellele on aastatega arendatud hulganisti masinõpet ja andmeanalüüsi teostavaid teke - populaarseimate hulka kuuluvad NumPy, PyTorch, Scikit-learn ja teised (Jawahar et al., 2023). Lisaks võimaldavad mitmed pildi-ja videotötlusteed nagu OpenCV (Open Source Computer Vision) või Pillow teostada otse Pythonis lihtsaid monteerimistoiminguid ja kasutada sügavat masinõpet, et tuvastada kaadris objekte (Khatik, 2023). Samuti on asjakohased ka Pythoni helitötlusteed - 2015. aasta SciPy konverentsil avalikustatud librosa pakett võimaldab kasutajatel analüüsida, visualiseerida ja töödelda helifaile (McFee et al., 2015). Sellele tööle asjakohaseim librosa funktsionaalsus on tempoanalüüs, mis on kindlasti vajalik näiteks muusika sünkroonimiseks videoklippide üleminekutega (McFee et al., 2015). Eelnimetatud kategooriatesse kuuluvate tööriistade kasutamine võimaldab Pythoni keskkonnas valmistada ette video- ja heliklippe, mida saab seejärel DaVinci Resolve teegi abil loodud skriptiga automaatselt üheks videoks kokku panna.

3. Tarkvara põhiohanded ja arhitektuur

Peatükis defineeritakse esialgu tarkvara arenduse nõuded. Seejärel käsitletakse tarkvara ehitust ja selle töövoogu, samuti tehisintellekti rolli tarkvara kirjutamises. Edasi tutvustatakse lähemalt töövoogu osi ja nende eesmärki, kasutatud tehnoloogiaid ja kasutamata jäänud alternatiive. Hinnatakse ka iga arhitektuuri osa ressursimahukust.

3.1 Tarkvara nõuded

Töö kirjutamise käigus vormistas autor loodava tarkvara nõuded, mis on järgnevalt toodud tähtsuse järjekorras alustades tähtsaimast ning millest lähtuvalt tehti töö valmimise protsessis mitmeid otsuseid:

1. Väljundi asjakohasus - rakenduse abil valminud videod kajastavad sisendi kõige meeolukamaid hetki, st iga klipp käsitleb mõnda eeldefineeritud teemat (nendest lähemalt alampeatükis 3.5 Videofailitöötlus).
2. Ligipääsetavus - rakendus on kättesaadav nii paljudele kasutajatele kui võimalik. Rakenduse kasutamiseks ei tohiks olla eeldus tehniline pädevus või maksevõime.
3. Kasutajamugavus - rakendust mõista ning selle põhifunktsionaalsust kasutada on võimalikult lihtne. Rakenduse liides peaks olema üheselt mõistetav, lihtsalt navigeeritav ning visuaalselt sobilik.
4. Ressursisäästlikus - on oluline, et rakendus ei pärsi töö ajal märgatavalt jõudlust. Rakendus ei tohi pärast töövoogu läbimist jätta alles üleliigseid faile. Analüüsiks ei tohi kuluda ebavõrdeliselt pikk aeg.

Tarkvara kirjutati ning on loodud töötamiseks operatsioonisüsteemis Windows, kuna see oli töö kirjutajale kõige tuttavam ning ligipääsetavam. Versioon, mis töötaks teistes operatsioonisüsteemides, kuulub võimalike edasiarenduste alla (vt peatükki 7).

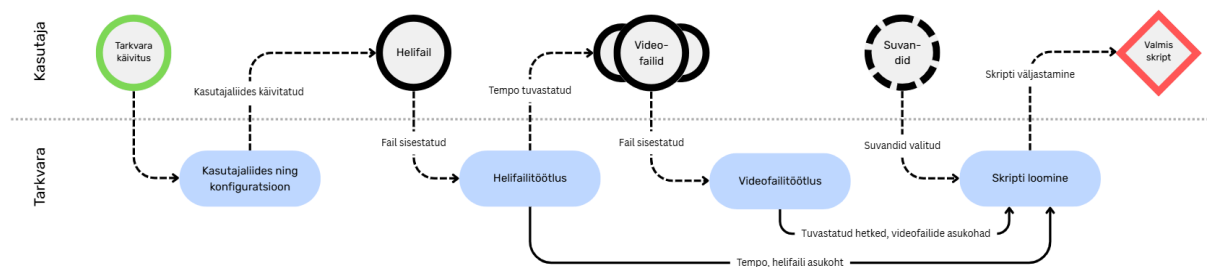
3.2 Tarkvara üldine arhitektuur ja tehisintellekti roll

Realiseeritud tarkvara koosneb teoreetiliselt kolmest põhiosast: helifailitöötluse tööriist, videofailitöötluse tööriist ning skripti genereerimise tööriist. Sellele lisanduvad kõrvalmoodulid, mis käsitlevad programmi konfiguratsiooni ning kasutajaliidest. Kokku koosneb tarkvara viiest moodulist: *main.py*, *config.py*, *media_processing.py*, *mediapipeline_utils.py* ning *resolve_script_generator.py*. Tarkvara töövoogu on järgmine:

1. kasutajaliidese ning videofailitöötluseks vajaliku mudeli valmendus;
2. sisendhelifaili tempo tuvastamine;

3. sisendvideofailides meeleolukaimate hetkede (nt inimese nägu, sõiduvahend, välistseen) tuvastamine;
4. tuvastatud tempo ja hetkede põhjal klippe lõikava ja kokku monteeriva skripti genereerimine;
5. genereeritud skripti väljastamine kasutajale koos juhendiga.

Tarkvara loomisel kasutati tehisintellekti rakendust Google Gemini (2025) programmeerimisprobleemide tuvastamise, koodile kommentaaride ning koodi refaktoreerimise (ingl *refactoring* - struktuuri lihtsustamine ja/või täiustamine, mis säilitab objekti välise käitumise) eesmärkidel (AKIT). Tekstiroboti tehtud muudatuste õigsust koodis kontrolliti ning tekstirobotit ei kasutatud programmi koodi sisu määravate otsuste tegemiseks või uurimuse kirjaliku osa sisu loomiseks.



Joonis 1. Tarkvara arhitektuur ja töövoog.

Tarkvara töövoog samade täitvad tarkvara osad on näidatud Joonises 1. Neid kirjeldatakse lähemalt järgnevatel alampeatükkidel.

3.3 Kasutajaliides ning konfiguratsioon

Tarkvara käivitamisel läbitakse esmalt kasutajaliidese ning videofailitöötluse mudeli ettevalmistus ning esitus kasutajale. Selleks läbitakse *main.py* moodulis kasutajaliidese loomine (jälgides *config.py* moodulis määratud konfiguratsiooni) ning seejärel leitakse *mediapipe_utils.py* mooduli abil kas kodukaustast või Google'i mudelite andmehoidlast Mediapipe objektituvastuse tööriist. Kasutaja töövoog läbimist ning selle tulemuste jälgimist kasutajaliideses toetavad *main.py* mitmed abifunktsioonid, mis uuendavad tekstivälju, kontrollivad nuppude olekut, loovad CSV-formaadis salvestatavaid analüüsi kokkuvõtteid, ning palju muud.

Kasutajaliides realiseeriti paketi tkinter abil. Laia toega ning kergkaalulist tkinterit loetakse standardseks Pythoni graafiliseks kasutajaliideseks (*Tkinter — Python Interface to Tcl/Tk*,

n.d.). Töö autori varasem kokkupuude selle paketi ja selle kokkusobivus töö nõuetega pakkuda kasutajasõbralikku ning ressursisäästliku kogemust (vt alampeatükki 3.1 Tarkvara nõuded) olid peamised tegurid selle kasuks otsustamisel.

3.4 Helifailitöötlus

Käesoleva tarkvara teine samm töövoos on tuvastada kasutaja poolt valitud helifaili tempo. Tempo tuvastamine lubab tarkvaral lõigata videofailidest välja asjakohase pikkusega klipid, sest selle alusel saab arvutada, kui suure ajavahemiku tagant on üks löök ehk millal võiks olla üleminek ühelt videoklipilt teisele. Näiteks kui helifailis kõlava muusika tempo on 120 lööki minutis, siis on üks löök iga poole sekundi tagant. Sellest tulenevalt võiks klipp samuti muutuda iga poole sekundi tagant. Praktikas on selline videoklipp aga liiga lühike ning lõpuks sattuvad skripti hoopiski 2 kuni 16 lööki pikad videoklipid (sellest rohkem alampeatükis 4.1 Liides ja kasutaja töövoog, kus kirjeldatakse monteerimisstiilide tööd).

Helifailitöötluse läbiviimiseks on moodulis *media_processing.py* funktsioon *detect_bpm_and_offset* - see tuvastab tempo ning leiab ka helifaili alguses oleva vaikuse pikkuse, et selle saaks helikliipi algusest eemaldada ja seeläbi löögid hiljem kaadrivahetustega ajaliselt vastavusse viia. See funktsioon võtab sisendiks kasutaja poolt antud failitee ning tagastab tempo, ühe löögi pikkuse sekundites, alguses oleva vaikuse pikkuse sekundites ning kogu helifaili pikkuse sekundites.

Tempo tuvastamise funktsiooni realiseerimiseks katsetati Madmom, Essentia ning librosa tarkvarapakette. Katsetamise tulemusena otsustati librosa kasuks kahe peamise eelise tõttu võrreldes alternatiividega. Esimene eelis on librosa ajakohasus: selle töö kirjutamise ajal hiljutisem ametlik väljalase librosa meeskonna poolt ilmus 2025. aasta märtsis (McFee et al., 2025). Madmomi viimane ametlik väljalase ilmus 2018. aasta novembris (Johannes Kepler University Department of Computational Perception, 2018), mistõttu on tema funktsionaalsus uutematel Pythoni versioonidel poolik. Librosa teine eelis on selle ühilduvus Windows operatsioonisüsteemidega: kuigi Essentia tarkvarapakett on samuti aktiivses arenduses, on sellel otsene tugi (ilma krosskompilaatorit kasutamata) ainult Linux ja macOS operatsioonisüsteemide kasutamiseks arvutiteks (UPF Music Technology Group, n.d.). Kuna käesolev programm valmis töötamiseks operatsioonisüsteemis Windows, siis polnud Essentia kasutamine töö eesmärkidega kooskõlas. Librosa ulatuslik dokumentatsioon ning selles leiduv tempo kalkuleerimisele pühendatud alammodul (McFee et al., 2025) olid samuti otsustavad faktorid.

3.5 Videofailitöötlus

Tarkvara teine peamine masinõppel põhinev samm on videotest meeolukate hetkede tuvastamine. Selle funktsiooni eesmärk on tagada, et tarkvara poolt kokku pandud video oleks täidetud asjakohaste hetkedega, millel on näha inimeste nägusid, loodust, hobiesemeid, loomi ja muud sellist.

Videofaile töötleb mooduli *media_processing.py* funktsioon *detect_video_moments*. See funktsioon võtab sisendiks ühe video failitee ja helialalüüsi käigus leitud löögi pikkuse. Esialgu määrab minimaalseks analüüsi käigus leitud klipi pikkuseks 2 löögi pikkust (samal põhjusel, mis kirjeldatud alampeatükis 3.4 Helifailitöötlus, loe rohkem peatükist 3.6 Skripti loomine) ning seejärel kutsub abifunktsiooni *_detect_scenes_mediapipe*, millele ta annab sisendiks video failitee ning kasutuses oleva objektituvastuse tööriista. Abifunktsioon leiab konfiguratsioonimoodulist *config.py*, kui tihti ta objekte tuvastama peab (vaikimisi 0.1 sekundit) ning kui lähestikku olevaid hetked ta kokku liidab (vaikimisi 0.2 sekundit). Seejärel kutsub funktsioon *_detect_scenes_mediapipe* omakorda *mediapipe_utils.py* mooduli funktsiooni *classify_frame_mediapipe*, et kategoriseerida kaader. Sisendiks annab ta OpenCV formaadis pildi ning objektituvastuse tööriista. Tuvastatavad kategooriad on järgmised:

- Inimese stseen, kus on näha inimene;
- Sõiduki stseen, kus on näha mõni sõiduvahend (nt auto, jalgratas, lennuk);
- Loomade stseen, kus on näha mõni tuvastustööriistale tuttav loom (nt koer, kass, elevant);
- Siseruumide stseen, kus on näha mõni tuvastustööriistale tuttav siseruumides leiduv ese (nt telefon, kaisukaru, voodi, vaas);
- Välistingimuste stseen, kus on näha mõni tuvastustööriistale tuttav välistingimustes leiduv ese (nt valgusfoor, puu, surfilaud);
- Muu stseen, juhul kui kaadris ei olnud ükski eelnevalt kirjeldatud stseenidest.

Funktsioon *classify_frame_mediapipe* tagastab kõige tõenäolisema kategooria või, kui ülaltoodud kategooriatesse stseen ei sobinud, siis muu stseeni kategooria. Funktsioon *_detect_scenes_mediapipe* paneb seejärel ühes ajavahemikus tuvastatud hetked kokku ning tagastab ühe video kõik tuvastatud hetked listina. Funktsioon *detect_video_moments* seejärel jagab need hetked inimeste hetkedeks ja esemete hetkedeks. Seejärel tagastab ta need kahes listis. Kogu protsess korratakse iga sisendiks antud videofailiga.

Tarkvara realiseerimise algusfaasides leidsid näotuvastuseks kasutust OpenCV Haar kaskaadide klassifikaator (*OpenCV: Cascade Classifier*, n.d.) ning Google'i poolt loodud Mediapipe projekti näotuvastustööriist (*Face Detection Guide*, n.d.). Sellele lisaks katsetati põgusalt PyTorch'i torchvision teeki, mis üheskoos ResNet eeltreenitud mudeliga oli alus stseenide ja muude esemete tuvastusele (*Models and Pre-trained Weights — Torchvision 0.22 Documentation*, n.d.). Tarkvara realiseerimise käigus ilmnnes Mediapipe tööriistade laiem kasutusvõimekus, mistõttu käsitletakse kogu näo- ning stseenituvastust lõplikus tarkvaraversioonis läbi Mediapipe objektituvastustööriista. Mediapipe'i objektituvastustööriist kasutab omakorda EfficientDet-Lite0 mudelit, mis võtab sisendiks 320 korda 320 pikslit resolutsioonis pildi ning tuvastab sellel ühe tõenäoliseima kuni kaheksakümnest objektist (*Object Detection Task Guide*, n.d.). Kuna töö nõuete hulka kuulub nii väljundi asjakohasus kui ka kasutajamugavus (vt alampeatükki 3.1 Tarkvara nõuded), siis kasutab tarkvara olemasolevatest mudelitest kõige tasakaalustatumat - “it strikes a balance between latency and accuracy” (*Object Detection Task Guide*, n.d.). See tagab, et tarkvara jõudluse arvelt ei tõsteta liialt mudeli täpsust ning ka vastupidi.

3.6 Skripti loomine

Failitöötluste tulemuste alusel kirjutatakse skript, mille saab kasutaja hiljem sisse lugeda tarkvaras DaVinci Resolve monteerimisprojekti. Selleks kasutatakse moodulit *resolve_script_generator.py*, mis sätestab *replib* mooduli abil limiidid sisendite pikkusele, et turvaliselt käsitleda suuremaid sisendandmeid, ning milles on defineeritud ainus funktsioon *create_script*. See funktsioon võtab sisendiks kõik eelnevad väljundid - helifaili tempo, ühe kaadri pikkus, leitud meeolukad hetked, sisendite failiteed, helifaili alguses oleva vaikuse pikkuse ning kasutaja poolt valitud monteerimisstiili (sellest lähemalt alampeatükis 4.1 Liides ja kasutaja töövoog). Lisaks on sisendiks veel heli- ja videoanalüüside edukat läbimist kinnitavad Boole'i muutujad (AKIT). Esmase vaikimisi kaadrisageduse, monteerimisstiilide ja muu konfiguratsiooni ning sisendite limiitidega representatsioonide tekitamisel täidab funktsioon muutujas *python_script_template* oleva skripti malli. Skripti malli koostamisel kasutati Raymond Radet' kohandatud DaVinci Resolve skriptimise dokumentatsiooni (n.d.). Skripti mall koosneb Resolve tarkvara sees vajaminevast konfiguratsioonist, abifunktsioonidest ajaühikute teisendamiseks ning põhifunktsioonist *create_resolve_timeline*, milles läbitakse Resolve tarkvaras olevas Pythoni programmeerimiskeskonnas järgnevad sammud:

1. Resolve tarkvara rakendusliidese ehk *API* (AKIT) võtmine;

2. Uue monteerimisajajoone (ingl *timeline*) loomine;
3. Sisendfailide lisamine projekti meediakogumisse (ingl *media pool*);
4. Videoklippide ettevalmistamine leitud hetkede ning valitud monteerimisstiili alusel;
5. Videoklippide järjestuse osaline juhuslikustamine;
6. Videoklippide lisamine ajajoonele;
7. Helifaili alguse ja lõpu lõikamine vastavalt helifaili alguses oleva vaikuse pikkusele ja kogu lõpliku video määratud pikkusele ning selle lisamine ajajoonele;
8. Algselt videotest kõlava heli kustutamine ajajoonelt.

Eeltoodud viies ning kaheksas samm on uurimuse autori isiklikud loominguks otsused ning nende suvanditeks muutmine kuulub programmi võimalike edasiarenduste alla (vt peatükki 6 Võimalikud edasiarendused).

Skripti malli viimane osa on kirjeldatud põhifunktsiooni käivitamine. Malli täitumisel salvestatakse skript kasutaja poolt valitud asukohta ning tagastatakse informatiivne hüplikaken.

Peale *replib* ning *tkinter* teekide ei kasuta see moodul ühtegi välist tehnoloogiat.

3.7 Tarkvara protsesside ressursimahukus

Esmalt hinnati liidese ja konfiguratsiooni ressursikulu. See on minimaalne - Windowsi tegumihaldur raporteeris nii lauaarvutil kui ka sülearvutil tarkvara vaikeolekus kuni 107-megabaidilist muutmälukulu, mis on ligi 75 korda väiksem kui muutmälukogus standardses 8-gigabaidilise muutmäluga arvutis (Lauaarvuti ning sülearvuti tarkvaraspetsifikatsioonidega saab tutvuda lisa 3).

Nii sülearvutis kui ka lauaarvutis katsetati helifailitöötluse ressursikulu 3 korda Mahavoki looga "Tunnete keel" ning 3 korda Pink Floyd'i looga "Echoes", kusjuures esimene lugu on võrdlemisi standardse pikkusega (3 minutit ja 40 sekundit) ning teine lugu väga pikk (24 minutit ja 11 sekundit). Esimese loo puhul oli mõlemas arvutis ligikaudne muutmälu keskmine kulu alla 400MB ning ajakulu helifaili analüüsimiseks 5 sekundit. Protsessorikoormus oli esimest lugu analüüsidel sülearvutil keskmiselt 8 protsenti ning lauaarvutil 3 protsenti. Teise ning pikema loo puhul küündis muutmälukulu mõlemas arvutis keskmiselt ligikaudu 1200MB suuruseks. Ajakulu oli siis sülearvutis 14-18 sekundit ning lauaarvutis 12 sekundit. Protsessorikoormus oli sülearvutil keskmiselt 10% ning lauaarvutil 5%. Kuna teise loo pikkusest tulenevalt on selle ressursikulu pigem erijuhtum ning

keskmised protsessori- ning muutmälukoormused ei häirinud oluliselt arvutite jõudlust, siis võib pidada neid tulemusi positiivseteks. Helianalüüsi ressursikulu katsetamise täpsemate tulemustega saab tutvuda lisas 4.

Videofailitöötamise ressursikulu katsetati mõlemal riistvaral kaks korda. Selleks kasutati kahte erinevat videofailide kogumit, mille resolutsioonid olid vastavalt 480p ja 1080p. Mõlema kogumi videofailide kaadrisagedus oli 30 kaadrit sekundis ning kogupikkus 2 minutit ja 51 sekundit. Esimese failikogumi puhul oli mõlemas arvutis ligikaudne muutmälu keskmine kulu alla 300MB ning ajakulu helifaili analüüsimiseks ligi 1 minut. Protsessorikoormus oli esimest kogumit analüüsides sülearvutil keskmiselt 21 protsenti ning lauaarvutil 7 protsenti. Teise ning kõrgema resolutsiooniga failikogumi puhul ei tõusnud muutmälukulu kummaski arvutis oluliselt, see suurenes vaid 100MB võrra. Analüüsi ajakulu oli nende videotega mõlemas arvutis pisut üle 2 minuti. Protsessorikoormus oli sülearvutil keskmiselt 41% ning lauaarvutil 13%. Siinkohal võib järeldada, et videoanalüüsist tingitud protsessorikoormus on piisav, et potentsiaalselt mõjutada väiksema jõudlusega arvutite töökiirust. Vaatamata sellele on ligikaudne keskmine ajakulu ühe sekundi video analüüsimiseks ka kõrgema resolutsiooniga sisendi puhul alla sekundi, ehk arvuti töökiirus on häiritud vähem aega, kui võtaks sisendiks antud videofaile otsast lõpuni läbi vaadata. Seega võib pidada neid tulemusi rahuldavaks. Videoanalüüsi ressursikulu katsetamise täpsemate tulemustega saab tutvuda lisas 5.

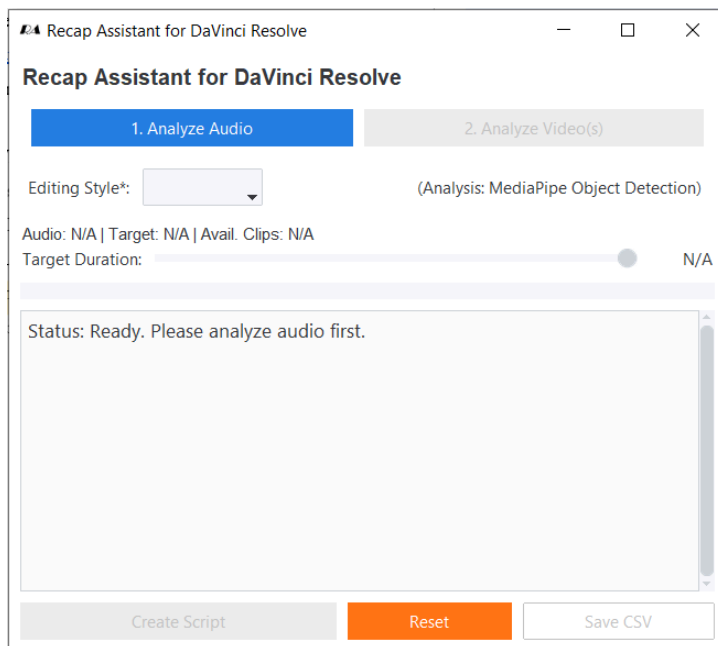
Skriptimooduli ressursimahukus oleneb suuresti sisendandmete suurusest, kuid skripti loomisele ei kulu kunagi üle paari sekundi ning protsessori ja muutmälu ressursikulu on samuti tühine.

4. Tarkvara ja loodud skripti kasutamise kirjeldus

Peatükis tutvustatakse valminud tarkvara Recap Assistant for DaVinci Resolve kasutajaliidest, demonstreeritakse selle töövoogu kasutaja perspektiivist ning kuidas tarkvara töö tulemusena loodud skripti DaVinci Resolve monteerimistarkvaras kasutada. Rakenduse lähtekood on saadaval GitHubis lingil <https://github.com/jespereller/RecapAssistant>.

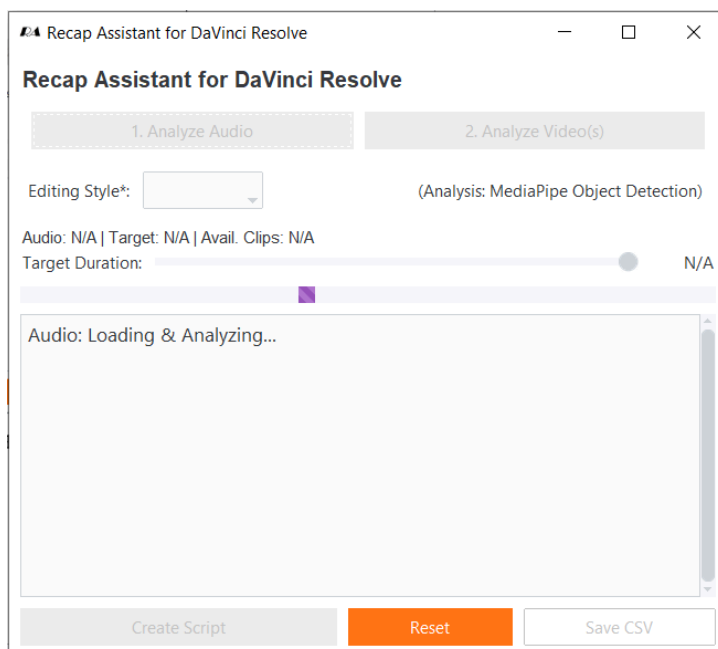
4.1 Liides ja kasutaja töövoog

Tarkvara liides on kirjutatud inglise keeles, et seda saaks kasutada võimalikult lai rahvusvaheline kasutajaskond. Sellegipoolest on saadaval eestikeelne juhend (juhend on nii inglise kui ka eesti keeles loetav lisas 2) ning keele tarkvarasisene valik kuulub võimalike edasiarenduste hulka, mis on kirjeldatud 7. peatükis. Tarkvara käivitamiseks avab kasutaja RecapAssistantForDaVinciResolve.exe programmi (kui ta kasutab kokku pakitud programmifaili) või käivitab *main.py* mooduli (kui tal on arendamise käigus kasutatud projektikaust kõikide moodulitega). Avaneb kasutajaliides, mis koosneb ühest töövoo läbimisel uuenduvast lehest. Tarkvara pealkirja all on klõpsatav nupp helianalüüsi alustamiseks ning mitteklõpsatav nupp videoanalüüsi alustamiseks, rippmenüü monteerimisstiili valimiseks ning videoanalüüsis kasutatavat mudelit nimetav informatiivne tekst. Nende elementide all paikneb esialgu mitteaktiivne liugur (ingl *slider*) (AKIT) ning sellega kaasnevad tekstiväljad video pikkuse valimiseks. Liuguri all on edenemisriba (ingl *progress bar*) (AKIT). Edenemisriba all on suurem tekstiboks koos veoriba (ingl *scroll bar*) funktsionaalsusega (AKIT). Liidese kõige alumised elemendid on kolm nuppu, millest esimene loob skripti (algul mitteaktiivne), teine võimaldab lähtestada liidese (alati aktiivne) ning kolmas salvestab CSV-formaadis analüüsi tulemusi (samuti algul mitteaktiivne). Veoribaga tekstiboksis uueneb tarkvara töövoogu jooksul tekst pidevalt kasutaja juhendamise ning informatsiooni edasiandmise eesmärgil. Tarkvara edukal käivitamisel teavitab see kasutajat kasutusvalmidusest ning juhendab kasutajat helifaili analüüsima (vt joonist 2).



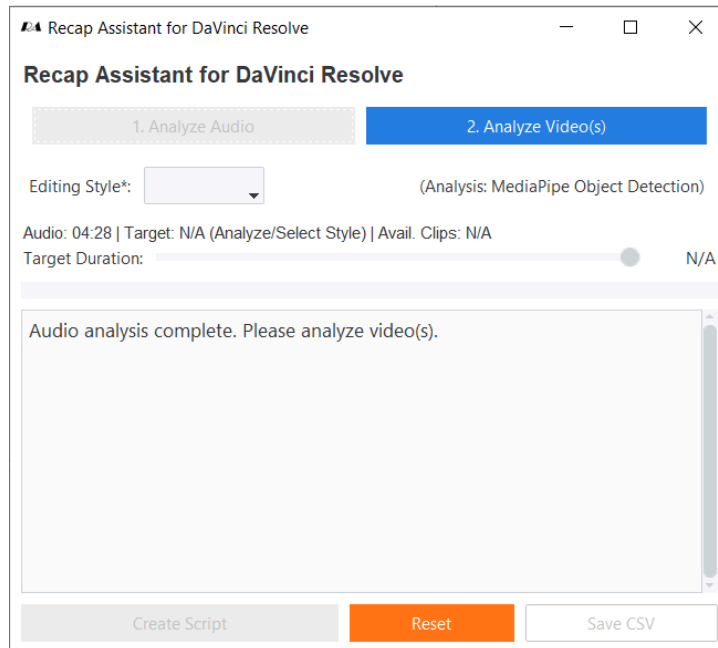
Joonis 2. Kasutajaliides tarkvara edukal käivitamisel.

Klõpsates nupule helianalüüsi alustamiseks avaneb hüpikaken, milles on kasutajal võimalik mistahes kaustast valida helifaili formaadis MP3, WAV, FLAC, AAC või OGG. Helifaili valides algab helianalüüs automaatselt - edenemisriba alustab tööd ning tekstiboks teavitab kasutajat analüüsi toimumisest (vt joonist 3).



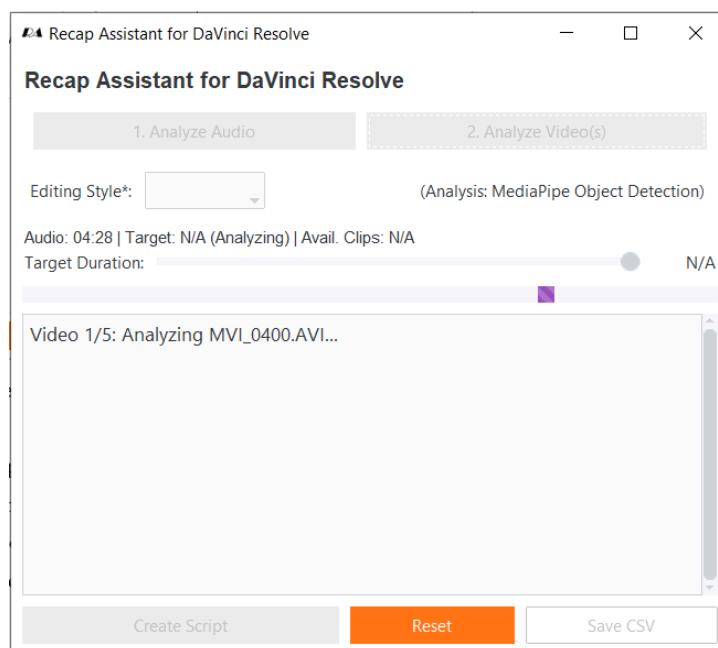
Joonis 3. Kasutajaliides helianalüüsi toimumise hetkel.

Helianalüüsi edukal lõppemisel edenemisriba lõpetab töö, tekstiboks teavitab analüüsi lõppemisest kasutajat ning juhendab kasutajat videofaili analüüsima (vt joonist 4).



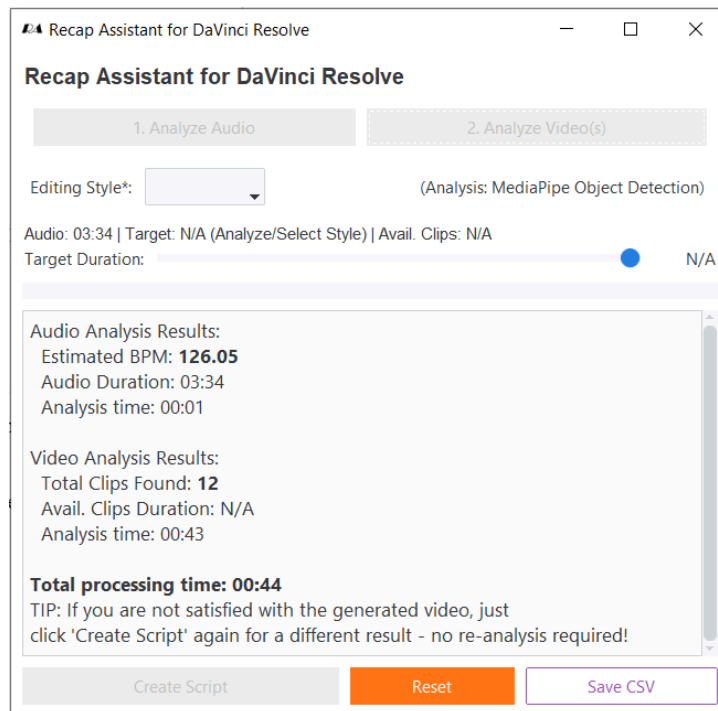
Joonis 4. Kasutajaliides helianalüüsi edukal lõppemisel.

Klõpsates nupule videoanalüüsi alustamiseks avaneb hüpikaken, milles on kasutajal võimalik mistahes kaustast valida ühe või mitu videofaili formaadis MP4, AVI, MOV, MKV või WEBM. Videofaili(de) valides algab videote analüüs automaatselt - edenemisriba alustab taas tööd ning tekstiboks teavitab kasutajat videote analüüsi toimumisest (vt joonist 5).



Joonis 5. Kasutajaliides videoanalüüsi toimumise hetkel.

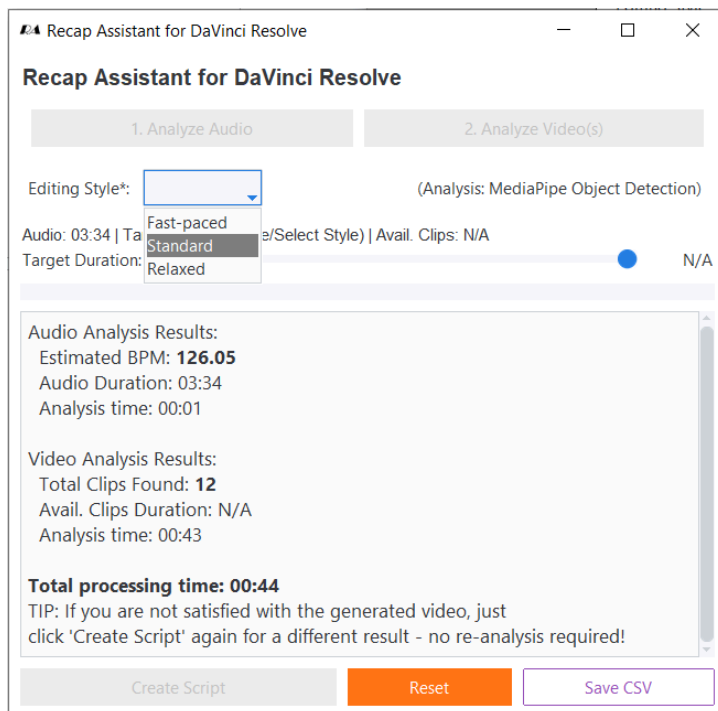
Videoanalüüsi edukal lõppemisel lõpetab edenemisriba töö ning tekstiboksis esitatakse kasutajale lühike kokkuvõtte heli- ja videoanalüüsi tulemustest, kusjuures on näha ka analüüside peale kulunud aeg. Samuti avaneb võimalus salvestada analüüsi tulemused CSV-formaadis, klõpsates selleks aktiveeritud nupule liidese alumises paremas nurgas (vt joonist 6).



Joonis 6. Kasutajaliides videoanalüüsi edukal lõppemisel.

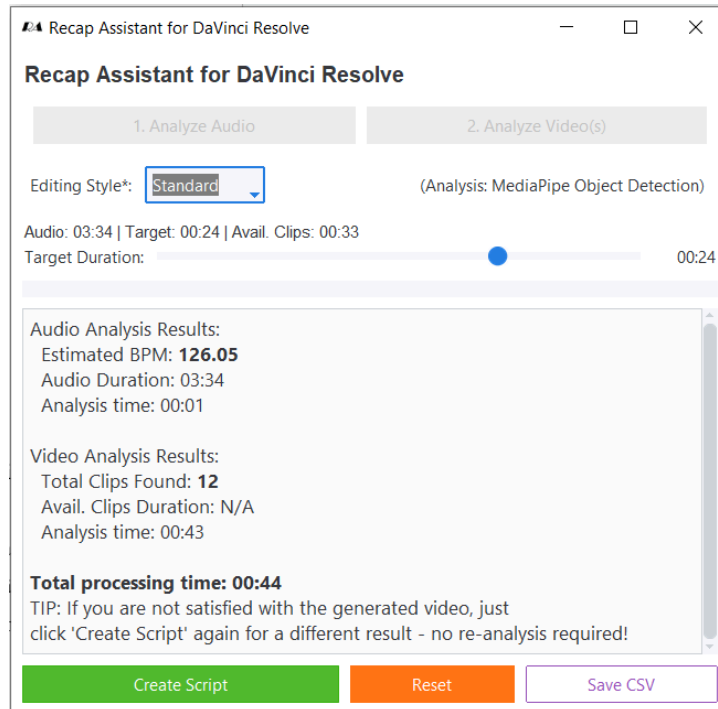
Enne skripti loomist on vaja valida monteerimisstiil - selleks avab kasutaja monteerimisstiili rippmenüü, kus on tal valik kolme eeldefineeritud stiili vahel (vt joonist 7):

1. rahulik (ingl *relaxed*) monteerimisstiil, mille puhul on videoklipid neli, kaheksa või kuusteist korda pikemad kui ühe löögi pikkus (ühe löögi pikkuse tuvastasime helifaili analüüsis), kusjuures suurem rõhk on pikematel videoklippidel.
2. standardne (ingl *standard*) monteerimisstiil, mille puhul on videoklipid kaks, neli, kaheksa või kuusteist korda pikemad kui ühe löögi pikkus, kusjuures suurem rõhk on keskmise pikkusega videoklippidel.
3. kiire (ingl *fast-paced*) monteerimisstiil, mille puhul on videoklipid kaks, neli või kaheksa korda pikemad kui ühe löögi pikkus, kusjuures suurem rõhk on lühematel videoklippidel.



Joonis 7. Kasutajaliides, kus hetkel on rippmenüüst valitud standardne monteerimisstiil.

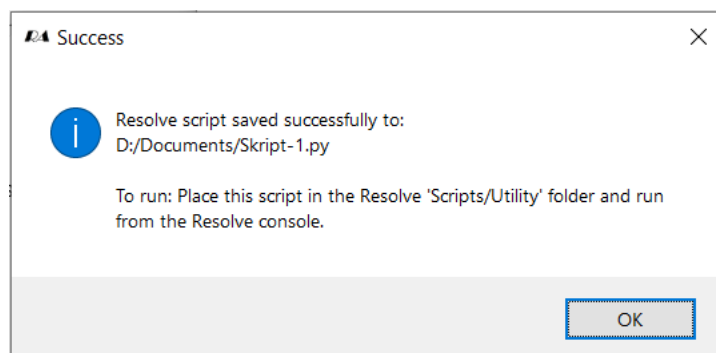
Kui stiil on valitud, avaneb võimalus luua skript. Lisaks on võimaldatud liuguril valida video soovitud pikkus - liuguri maksimaalne pikkus on kas analüüsi käigus leitud hetkede kogupikkus valitud stiilis või helifaili pikkus, olenevalt sellest, kumb pikkus on väiksem (vt joonist 8).



Joonis 8. Kasutajaliides, kus on liuguril valitud video soovitavaks pikkuseks 24 sekundit, kusjuures maksimaalne pikkus on 33 sekundit.

Klõpsates skripti loomise nupule tekib hüpikaken, kus saab valida skripti nime ning salvestamise asukoha kettal. Selleks, et DaVinci Resolve skripti leiaks, on vaja see kas otse salvestada või hiljem toimetada spetsiaalsesse kausta, mille leiab DaVinci Resolve installeerimiskaustast Fusion skriptide alamkaustast. Tarkvara kasutusjuhendis (loetav lisas 2) on täpsem juhend, kuidas selle kausta üles leida.

Nagu on mainitud ka nõuandena tekstiboksis, siis on võimalik järjest mitu korda pärast ühte analüüsi skript luua ning need kõik on veidi erinevad - seeläbi on kasutajal võimalik valida mitme väljundi hulgast nende silmis kõige kvaliteetsema.



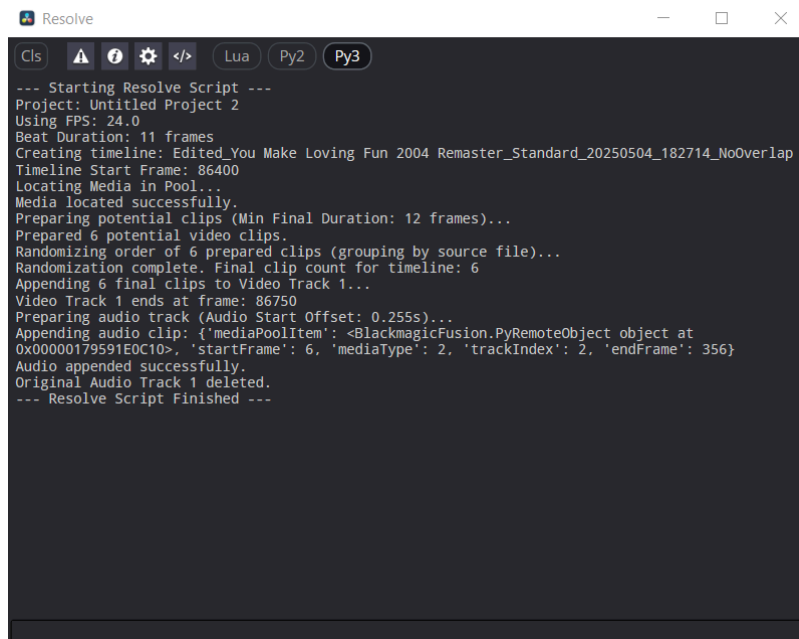
Joonis 9. Skripti edukat loomist kinnitav hüpikaken.

Kui skripti asukoht ja nimi on kinnitatud, kinnitab hüpikaken, et skript on edukalt loodud (vt joonist 9). Tarkvara töövoog on läbitud.

4.2 Skripti kasutamine DaVinci Resolve monteerimistarkvaras

Valmistatud skripti kasutamiseks peab kasutaja käivitama monteerimistarkvara DaVinci Resolve ning alustama uue projekti. Kui skript on kasutaja poolt toimetatud eelpool välja toodud õigesse kausta, siis ilmub see tööriistaribal leitavas skriptide rippmenüüs. Selle valides alustab skript tööd ning loeb sisse samad sisendfailid, lõikab videotest välja videotöötluse käigus leitud hetked ning paigutab need üheskoos sisendhelifailiga ajajoonele. Edaspidi on kasutajal soovi korral võimalik teha täiendavaid monteerimistoiminguid ning lõpuks video ka eksportida.

Kasutajal on võimalus tutvuda skripti tööga avades tööriistaribalt konsooli. Sinna tekib skripti jooksumises logi selle toimingutest (vt joonist 10).



```
Resolve
Cls [Warning] [Info] [Settings] [Code] [Lua] [Py2] [Py3]
--- Starting Resolve Script ---
Project: Untitled Project 2
Using FPS: 24.0
Beat Duration: 11 frames
Creating timeline: Edited_You Make Loving Fun 2004 Remaster_Standard_20250504_182714_NoOverlap
Timeline Start Frame: 86400
Locating Media in Pool...
Media located successfully.
Preparing potential clips (Min Final Duration: 12 frames)...
Prepared 6 potential video clips.
Randomizing order of 6 prepared clips (grouping by source file)...
Randomization complete. Final clip count for timeline: 6
Appending 6 final clips to Video Track 1...
Video Track 1 ends at frame: 86750
Preparing audio track (Audio Start Offset: 0.255s)...
Appending audio clip: {'mediaPoolItem': <BlackmagicFusion.PyRemoteObject object at 0x00000179591E0C10>, 'startFrame': 6, 'mediaType': 2, 'trackIndex': 2, 'endFrame': 356}
Audio appended successfully.
Original Audio Track 1 deleted.
--- Resolve Script Finished ---
```

Joonis 10. DaVinci Resolve konsool, milles on näha skripti toimingute logi.

DaVinci Resolve tarkvara tegevustega on täpsemalt võimalik tutvuda lisas 2 tarkvara kasutusjuhendis.

5. Tagasiside

Tagasiside koguti teenuses Google Forms. Tagasisideküsitlus koosnes kolmest osast:

1. Kolme eri stiilis tarkvaraga loodud videoga tutvumine ja eelistuse valimine;
2. Inimese tehtud videoga tutvumine ja võrdlemine;
3. Üldised küsimused.

Esimeses osas “Kolme eri stiilis videoga tutvumine ja eelistuse valimine” paluti vastajatel tutvuda kolme tarkvara poolt loodud skripti alusel loodud videoga, kusjuures iga video oli loodud erineva monteerimisstiiliga: rahulik (ingl *relaxed*), standardne (ingl *standard*) ja kiire (ingl *fast-paced*). Esimeses küsimuses paluti vastajatel valida nendest kolmest stiilist enda lemmik. Enim pälvis vastanute heakskiidu kiire ja standardne stiil, kusjuures ainult kaks vastanut väljendasid huvi aeglasema stiili vastu. Täpsustanud vastanud nentisid, et kiire ja standardne stiil tootsid “elulisema” väljundi ning aeglasema stiili puhul “ei ilmnenud efekti, et klippide vahetumine toimuks nagu rütmi peal.” Vastustest tulenevalt kuulub võimalike edasiarenduste alla rahulikuma monteerimisstiili kohandamine. Sellegipoolest tunti vastanute hulgas huvi iga monteerimisstiili vastu ehk monteerimisstiilide valik on rahuldav.

Teise osa “Inimese tehtud videoga tutvumine ja võrdlemine” hakul paluti küsitletavatel tutvuda samadest sisendfailidest koosneva videoga, mille oli monteerinud uurimuse autor enne uurimusega alustamist. Seejärel paluti hinnata viiepunktilisel skaalal tarkvara abil tehtud videote kvaliteeti võrreldes uurimuse autori tehtud videoga, kusjuures hinne “1” tähendas, et tarkvara abil tehtud video on palju ebakvaliteetsem, ning hinne “5” tähendas, et tarkvara abil tehtud video oli sama kvaliteetne. Samuti paluti antud hinnet põhjendada lühikese võrdleva kokkuvõttega. Kõige populaarsem hinne oli “4”, selle hinnangu andsid pisut üle poole vastanutest ning seda põhjendati arvamustega, et tarkvara abil koostatud video meeleolu kannatas ainult pisut võrreldes inimese tehtud videoga. Hinnati tarkvaraga tehtud videote ühtlast rütmi, “robootilisuse” puudumist ning “lahedust”, kuid palju kommenteeriti tarkvara tehtud videotest puuduvat kronoloogilisust. Seda põhjustab alampeatükis 3.6 Skripti loomine kirjeldatud skripti loomise funktsioonis leiduv 5. samm - videoklippide järjestuse osaline juhuslikustamine. Tagasisidest tulenevalt kuulub selle sammu valikuliseks tegemine programmi võimalike edasiarenduste alla. Hinde “3” andsid ligikaudu 27,3% vastanutest ning seda hinnet põhjendati samuti kronoloogilisuse puudumisega. Sellega põhjendas oma hinnet ka üksik hinne “2” andnud vastanu. Üks vastanu andis hinne “5”, leides, et tarkvaraga loodud video oli “sujavam ja meeleolukam”, ning inimese tehtud video oli “hüplik ning kohati igav”.

Teise osa tagasisidele hinnangut andes on vajalik arvestada ka ajakulu: kui ise otsast lõpuni tehtud 30-sekundilise video loomiseks kulus autoril vähemalt veerand tundi keskendunud tööd, siis samasuguse, aga tarkvara abil tehtud, video analüüside lõpetamiseks kulus tarkvaral 2 minutit ja 6 sekundit. Videolooje protsessi juba paar korda läbinud kasutaja on võimeline ülejäänud toimingud (sh skripti salvestamine ja lugemine tarkvarasse DaVinci Resolve, video eksportimine) läbima alla minutiga - seega on võimalik tarkvara kasutades videolooget kiirendada selle näite puhul kuni viiekordselt, ja seda ise klippe pingsalt analüüsivõimata. Selle lihtsustatud töövooga saada tulemuseks vastanute silmis peaaegu samaväärse video on autori silmis väga hea tulemus ning seeläbi on ka tagasiside positiivne.

Tagasiside kolmas osa “Üldised küsimused” uuris tagasisidestajate eeldatavat analüüsi ajakulu ning huvi sellise programmi vastu. Küsitletavatele anti ette nähtud videote parameetrid (sh originaalklippide resolutsioon ja kogupikkus, analüüsi käigus leitud hetkede kogupikkus) ning uuriti, kaua oleks vastanu valmis ootama tarkvara analüüsi lõppemist, et tarkvara kasutamine oleks neile “mugav ning väärtuslik”. Enamus vastanutest (ligikaudu 81,8%) leidsid, et nad on valmis ootama kuni 5 minutit või rohkemgi. See on samuti positiivne tulemus, sest video tegelik analüüsi-ajakulu oli pisut üle 2 minuti ehk parajalt alla vastanute ootuste. Lisaks uuriti, kas vastajatel on huvi programmi tulevikus kasutada - ligikaudu 72,7% vastanutest vastasid jaatavalt ning ülejäänud vastasid “Ei tea”. Jaatavaid vastuseid põhjendati tarkvara näilise mugavuse ja kasulikkusega. Mitmed vastanud toonitasid videote monteerimise huvi olemasolu, kuid oskuse puudumist. “Ei tea” valinud vastanud nimetasid üldist huvi puudumist “mälestuse jäädvustamise” või videote monteerimise vastu, aga ei välistanud vajadusel programmi kasutamist. Tagasiside andnud inimeste üldine huvi tarkvara vastu oli autori silmis positiivne ning rõõmustav tulemus.

Tagasisidevormi puudutava lisainformatsiooniga ning selle küsimustega saab tutvuda 6. lisa.

6. Võimalikud edasiarendused

Tarkvara loomise käigus ning töö tulemusi ning tagasiside analüüsidest tuvastati mitu võimalikku edasiarendust.

Alampeatükis 3.1 defineeritud programmi nõuete hulgas kõige olulisem on väljundi asjakohasus. Tagasiside andnud väljundiga tutvunute kommentaarides kõlas kõige enam mure, et videote asjakohasust nõrgestas klippide silmnähtavalt suvaline järjekord. Seda suvalist järjekorda määrab skripti loomise funktsioonis leiduv 5. samm, ning tagasiside arvestades on esmatähtsaim edasiarendus eemaldada järjekorra juhuslikustamine funktsioon. Edaspidi saaks videoklippide juhuslikustamise lisada tagasi programmi valikulise suvandina.

Skriptiloomes funktsiooni 8. samm käsitleb samuti autori isiklikku loomingulist otsust - originaalvideofailide heli kustutamist videost. Selle otsuse kasuks räägib veel asjaolu, et skriptimise läbi pole hetkel tarkvaras DaVinci Resolve võimaldatud originaalheli vaiksemaks sätestamist ning kaks maksimaalsel valjudusel koos esitatavat heli rikuksid vaatamiskogemust. Võimaliku edasiarendusena oleks paindlikuma väljundi nimel kasulik lisada programmi suvand (nt märkeruut ehk ingl *checkbox*) originaalheli säilitamiseks (AKIT). Sellisel juhul saab kasutaja monteerimistarkvara sees hiljem käsitsi helitasemeid vastavalt soovile sättida.

Vastavalt tagasiside 1. osa vastustele võib pidada võimalikuks edasiarenduseks ka rahulikuma ehk ingl *relaxed* monteerimisstiili kohandamist, kuna selle puhul kommenteeriti, et heliga rööbiti videoklippide üleminekute efekti ei esinenud.

Tarkvara nõuetest teine olulisem oli ligipääsetavus. Esimene ning üldisem edasiarendus ligipääsetavuse nimel võiks olla versioonide loomine teistele operatsioonisüsteemidele nagu näiteks Linux või MacOS. Ligipääsetavuse parandamiseks on veel võimalik edasiarendus lisada valik video monteerimise täielikuks automatiseerimiseks, kasutades DaVinci Resolve skripti asemel mõnda Pythoni videotötlusteedi. See teeks üldise kasutamise ligipääsetavamaks ning sobiks enim lõviosale potentsiaalsest kasutajakonnast, kuna ei peaks juurde installeerima lisatarkvara. Lisaks likvideeriks see mitmed mured seoses DaVinci Resolve skriptimisega, nt eelnevalt mainitud helitasemete sättimine. See-eest ohverdaks selline valik võimaluse teha järeltoiminguid, mis DaVinci Resolve projektis on laiahaardelised. Võimalik edasiarendus oleks ka tasuta DaVinci Resolve Studio

automatiseerimisfunktsioonidele toe lisamine, juhul kui mõni potentsiaalne kasutaja on selle endale soetanud.

Programmi nõuete alla kuuluva kasutajamugavuse nõude paremaks täitmiseks võiks tulevikus olla võimalik valida programmi kasutajaliidese keelt. Seeläbi saaksid eesti või mõnda muud keelt kõnelevad potentsiaalsed kasutajad mõista kasutajaliidest ilma juhendi poole pöördumata.

Programmi ressursisäästlikuse tõstmiseks on potentsiaalne arendussuund anda kasutajale võimaluse valida mõne kergekaalulisema esemetuvastuses kasutatud mudeli kasuks. Siinkohal võiks alustada Mediapipe enda väiksema mudeliga ning seejärel uurida ka teisi teenusepakkujaid või isegi luua oma mudel.

Viimase mõttena tasub hoida silma peal ka tehisintellekti arengul - kui avalikustatakse mõni tehisintellekti tööriist videote mõtestatumaks analüüsiks, mille väljundi saaks mugavalt sisestada käesolevasse Pythoni programmi, siis on see hea võimalus esemetuvastusmudelid millegi hoopis võimsamaga asendada ja seeläbi veelgi kasulikumat video monteerimist lihtsustavat tarkvara pakkuda.

Kokkuvõte

Bakalaureusetöö eesmärk oli arendada programm kokkuvõtva video kokku monteerimise lihtsustamiseks. Programmi põhifunktsionaalsusteks said sisendfailide põhjal läbi viidud heli- ning videoanalüüs, mille järel tagastati kasutajale videot monteeriv Pythoni skript monteerimistarkvara DaVinci Resolve siseseks kasutamiseks.

Enne rakenduse arendusega alustamist tutvuti videote monteerimise ajaloo ning hiljutiste arengutega. Seejärel viis loo autor end kurssi DaVinci Resolve monteerimistarkvara poolt toetatud Pythoni skriptimise võimaluste ja puudujääkidega. Analüüsiti Pythoni värvikat masinõppeteekide maailma ning tuvastati selle uurimuse jaoks kõige väärtuslikumad.

Programmi arendamise käigus visandati tarkvara algeline struktuur ning seejärel realiseeriti helifaili tempo tuvastuseks vajalik librosa teeki kasutav helianalüüsi funktsioon. Järgmisena valmis videoanalüüsi funktsioonide kogum, mis kasutas erinevate tööriistade katsetamise järel lõpuks Google'i Mediapipe esemetuvastustööriista. Lõpuks kirjutati automaatselt monteeriv DaVinci Resolve skript ning see muudeti malliks skripti loomise funktsioonis, mis lisas skripti eelneva heli- ja videoanalüüsi tulemused. Tarkvara valmides ning esimeste tulemuste saades mõõdeti ka tarkvara protsesside ressursikulu, mis oli kokkuvõtvalt nii suurema kui ka väiksema jõudlusega arvutitel rahuldav.

Kui tarkvara oli oma esitataval kujul valmis, kirjeldati selle autori poolt seatud nõudeid ning esitleti võimalikku kasutaja töövoogu koos illustreerivate joonistega kasutajaliidese eri etappidest. Põgusalt tutvustati skripti kasutamist DaVinci Resolve monteerimistarkvaras.

Seejärel kogus töö autor struktureeritud tagasiside - küsitletavad tutvused nii tarkvara abil kui ka inimese käe läbi tehtud sarnaste videotega ning võrdlesid neid, samuti jagasid nad tagasisidet oma muude ootuste kohta tarkvaraga seoses. Enamjaolt oli tagasiside positiivne (kiideti tarkvara skriptide väljundite võrdlemisi tugevat asjakohasust ning meeleolukust) ning selle analüüsi käigus tuvastati mitu võimalikku edasiarendust.

Töö lõpusirgele jõudes tuvastas töö autor edaspidiseid arendusvõimalusi, mida kirjeldati töö nelja põhinõude alusel. Arendusvõimaluste hulka kuulusid klippide juhuslikustamise eemaldamine, erinevate operatsioonisüsteemide jaoks versioonide loomine ning palju muud. Võimalikud edasiarendused võeti kokku mõttega, et lähitulevikus on ehk hoogsasti arenevast tehisintellekti maailmast pärineva uue tööriistaga tarkvara veelgi täiendada.

Viidatud kirjandus

AKIT = Andmekaitse ja infoturbe leksikon. Cybernetica AS. <https://akit.cyber.ee/>.

Fernandes, J. (2024, November 18). From Analogue to Digital: How Video Editing has Evolved. *Medium*.

<https://medium.com/@joaofernandes97/from-analogue-to-digital-how-video-editing-has-evolved-f807d6c4aa6e> (vaadatud 08.12.2024)

Waters, C. (2019, January 27). The Aussie Tech Company Behind the Oscar Nominees. *The Sydney Morning Herald*.

<https://www.smh.com.au/business/small-business/the-aussie-tech-company-behind-the-oscar-nominees-20190124-p50tg1.html> (vaadatud 13.05.2025)

Ruczak, K. (2023). AI: New Applications for Post Production: Innovative Features for VFX, Animation, Video Editing, Audio Editing, and beyond. *Computer Graphics World*, 46(3), 22–23.

Kroll, N. (2015, October 15). *Why I now believe DaVinci Resolve is the best editing platform on the market*. Noam Kroll.

<https://noamkroll.com/why-i-now-believe-davinci-resolve-is-the-best-editing-platform-on-the-market/> (vaadatud 08.12.2024)

Shaw, K. (n.d.). *History for Colorists*. Finalcolor.

<https://www.finalcolor.com/history-for-colorists/> (vaadatud 08.12.2024)

Radet, R. (n.d.). *DaVinci Resolve Scripting API Doc v19.1*. GitHub.

<https://gist.github.com/X-Raym/2f2bf453fc481b9cca624d7ca0e19de8> (vaadatud 08.12.2024)

Ramani, S. (2024, July 19). *You will need the Studio version to run scripts from the terminal. For the free version, the Console window* [Comment on the online forum post “*scriptapp(‘Resolve’) returns None*”]. Blackmagic Forum.

<https://forum.blackmagicdesign.com/viewtopic.php?f=12&t=202325#p1065063> (vaadatud 05.05.2025)

History and License. (n.d.). Python Documentation. <https://docs.python.org/3/license.html> (vaadatud 08.12.2024)

- Sundaram, J., Gowri, K., Devaraju, S., Gokuldev, S., Jayaprakash, S., Anandaram, H., Manivasagan, C., & Thenmozhi, M. (2023). An Exploration of Python Libraries in Machine Learning Models for Data Science. In *Advances in computational intelligence and robotics book series* (pp. 1–31).
<https://doi.org/10.4018/978-1-6684-8696-2.ch001>
- Khatik, K. (2023, March 11). Advanced Image and Video Processing Techniques Using Python | Computer Vision & Deep Learning Applications | Medium. *Medium*.
<https://medium.com/@kapildevkhatik2/advanced-image-and-video-processing-techniques-using-python-549fb1cf224e> (vaadatud 08.12.2024)
- McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and Music Signal Analysis in Python. *Proceedings of the Python in Science Conferences*, 18–24. <https://doi.org/10.25080/majora-7b98e3ed-003>
- McFee, B., McVicar, M., Faronbi, D., Roman, I., Gover, M., Balke, S., Seyfarth, S., Malek, A., Raffel, C., Lostanlen, V., Van Niekirk, B., Lee, D., Cwitkowitz, F., Zalkow, F., Nieto, O., Ellis, D., Mason, J., Lee, K., Steers, B., . . . Südholt, D. (2025). librosa/librosa: 0.11.0. *Zenodo*. <https://doi.org/10.5281/zenodo.15006942> (vaadatud 13.04.2025)
- Google. (2025). Google Gemini (2.5 Pro) [suur keelemudel]. <https://gemini.google.com/>
- tkinter* — *Python interface to Tcl/Tk*. (n.d.). Python Documentation.
<https://docs.python.org/3/library/tkinter.html> (vaadatud 05.05.2025)
- Johannes Kepler University Department of Computational Perception. (2018, November 14). *Madmom: Python audio and music signal processing library*. GitHub.
<https://github.com/CPJKU/madmom> (vaadatud 13.04.2025)
- UPF Music Technology Group. (n.d.). *Installing Essentia — Essentia 2.1-beta6-dev documentation*. <https://essentia.upf.edu/installing.html> (vaadatud 13.04.2025)
- OpenCV: Cascade Classifier*. (n.d.).
https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html (vaadatud 05.05.2025)

Face detection guide. (n.d.). Google AI for Developers.

https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector (vaadatud 05.05.2025)

Models and pre-trained weights — Torchvision 0.22 documentation. (n.d.).

<https://pytorch.org/vision/stable/models.html#classification> (vaadatud 05.05.2025)

Object detection task guide. (n.d.). Google AI for Developers.

https://ai.google.dev/edge/mediapipe/solutions/vision/object_detector (vaadatud 05.05.2025)

Lisad

I. Tarkvaranõuded

Tarkvara mõlema versiooni kasutamiseks on vajalik riistvara, mis kasutab operatsioonisüsteemi Microsoft Windows (teadaolevalt töötab versioonil Windows 10 ja Windows 11).

EXE-faili käivitamiseks lisanduvad tarkvaranõuded puuduvad.

Projektikaustast *main.py* mooduli käivitamiseks on vajalik Pythoni keskkond versiooniga 3.11.0 või uuem. Programmi käivitamiseks on vajalikud moodulid *tkbootstrap*, *opencv-python*, *librosa* ja *mediapipe*. Hangi moodulid käsuga `pip install ...`, näiteks `pip install tkbootstrap`.

II. Kasutusjuhend

EESTIKEELNE KASUTUSJUHEND:

1. Ava *RecapAssistantForDaVinciResolve.exe* (võtab pisut aega...)
2. Klõpsa nupul '1. Analyse Audio' ning vali helifail oma arvutist. Helianalüüs algab automaatselt
3. Helianalüüsi lõppedes avaneb valik '2. Analyse Video(s)'. Klõpsa sellel nupul ning vali üks või mitu videofaili oma arvutist. Videoanalüüs algab automaatselt
4. Oota, kuna videoanalüüs võtab pisut aega. Selle lõppedes avaneb lühikokkuvõtte analüüside tulemustest. Samuti tekivad probleemide korral hüplikaknad, mis kirjeldavad probleemi olemust. Probleemi korral lähtuda kasutajajuhendi 7. sammust ning uuesti proovides välja jätta probleeme tekitanud sisendfaili(d)
5. Vali 'Editing Style*' rippmenüüst endale sobiv monteerimisstiil - relaxed on rahulikum stiil, standard on tavapärase stiil ning fast-paced on kiiremas tempos stiil.
6. Stiili valides avaneb valik 'Create Script' - klõpsa sellel ning salvesta skript suvalisse kausta VÕI otse DaVinci Resolve skriptide kausta (sellest lähemalt allpool)
7. Soovi korral on võimalik mistahes sammu ajal klõpsata nupule 'Reset', et kasutajaliides lähtestada ning uuesti alustada (NB! Hetkeversioonis ei lõpeta 'Reset' nupule klõpsamine taustaprotsessi - mõne suurema analüüsi tühistamise puhul on ehk mõistlik programm siiski taaskäivitada)
8. Soovi korral on analüüsi lõppedes kasutajal salvestada CSV-formaadis kokkuvõtte analüüsi tulemustest klõpsates nupul 'Save CSV'. Salvesta see mistahes kausta.

KUIDAS LEIDA DAVINCI RESOLVE SKRIPTIDE KAUSTA?

DaVinci Resolve skriptide kaust asub üldjuhul failiteel:

```
'C:\Users\<TEIE KASUTAJA>\AppData\Roaming\Blackmagic Design\DaVinci Resolve\Support\Fusion\Scripts\Utility'
```

Kuna AppData on peidetud kaust, on lihtsaim viis sellele ligi pääsemiseks otsida Windowsi otsingust (tegumiribal vasakul, Windows logo kõrval) '%appdata%' ning avada esimene otsingutulemus.

Alternatiivne skriptide kaust asub failiteel:

```
'C:\ProgramData\Blackmagic Design\DaVinci Resolve\Fusion\Scripts\Utility'
```

Kuna ProgramData on peidetud kaust, pääseb sellele sarnaselt ligi: otsi Windowsi otsingus '%programdata%'.

KUIDAS KASUTADA SKRIPTI DAVINCI RESOLVE TARKVARA SEES?

1. Ava DaVinci Resolve
2. Klõpsa nupul 'New Project', et avada uus projekt VÕI ava olemasolev projekt
3. Ava ülevalt tööriistaribalt rippmenüü 'Workspace' --> 'Scripts' --> Vali enda skript tekkinud hõljukboksist
4. Skript peaks automaatselt lugema sisse sisendfailid ning kokku panema ajajoone. Skripti tööd saab jälgida konsoolist, mille saab avada valides 'Workspace' --> 'Console'
5. Siinkohal võib soovi korral tutvuda ajajoonega, tõsta klippe ümber, lisada efekte ning sooritada muid monteerimistoiminguid
6. Kui video on valmis, saab valida ülevalt paremalt alamtööriistaribalt valiku 'Quick Export' ning valida omale meelepärase väljundi. Video salvestatakse kasutaja valitud kausta

USER INSTRUCTIONS IN ENGLISH:

1. Open RecapAssistantForDaVinciResolve.exe (takes a little while...)
2. Click the '1. Analyze Audio' button and select an audio file from your computer. The audio analysis will begin automatically
3. When the audio analysis is complete, the '2. Analyzing video(s)' option will become available. Click on this button and select one or more video files from your computer. The video analysis will begin automatically
4. Wait, as the video analysis will take some time. When it is finished, a brief summary of the analysis results will open. In case of any problems, pop-up windows will appear that describe the nature of the problem. In case of any problems, follow step 7 of the user instructions and try again without the input file(s) that caused the problem
5. Select the editing style that suits you from the 'Editing Style*' drop-down menu
6. When you select a style, the 'Create Script' option will open - click on it and save the script to any folder OR directly to the DaVinci Resolve scripts folder (more on this below)
7. You can click the 'Reset' button at any time to reset the user interface and start over (NB! In the current version, clicking the 'Reset' button does not stop the background process - for longer analyses, it may be best to close and re-open the program)
8. If desired, at the end of the analysis, the user can save a summary of the analysis results in CSV format by clicking the 'Save CSV' button. Save it to any folder.

HOW TO FIND THE DAVINCI RESOLUTION SCRIPT FOLDER?

The DaVinci Resolve scripts folder is usually located at:

'C:\Users\<YOUR USER>\AppData\Roaming\Blackmagic Design\DaVinci Resolve\Support\Fusion\Scripts\Utility'

Since AppData is a hidden folder, the easiest way to access it is to search for '%appdata%' in Windows Search (on the taskbar on the left, next to the Windows logo) and open the first search result.

The alternative scripts folder is located at:

'C:\ProgramData\Blackmagic Design\DaVinci Resolve\Fusion\Scripts\Utility'

Since ProgramData is a hidden folder, it can be accessed similarly: search for '%programdata%' in Windows search.

HOW TO USE A SCRIPT WITHIN DAVINCI RESOLVE SOFTWARE?

1. Open DaVinci Resolve
2. Click on the 'New Project' button to open a new project, OR open an existing project
3. Open the 'Workspace' drop-down menu from the top toolbar --> 'Scripts' --> select your script from the hover box.
4. The script should read the user input files and assemble the timeline. The script's process can be followed from the Workspace console, which can be opened by selecting 'Workspace' --> 'Console'
5. Now, you can view the timeline, move clips around, add effects, and perform other editing operations if desired.
6. When the video is ready, you can select 'Quick Export' from the sub-toolbar at the top right and choose the output you prefer. The video will be saved to the folder selected by the user.

III. Katsetamise käigus kasutatud sülearvuti ja lauaarvuti tehniline spetsifikatsioon

Arvuti	Protsessor (CPU)	Muutmälu (RAM)	Andme- kandjad	Graafika- protsessor (GPU)
Sülearvuti	4 tuuma, max. sagedus 4.2GHz	16.0 GB	500GB SSD	Integreeritud protsessorisse
Lauaarvuti	8 tuuma, max. sagedus 4.7Ghz	16.0 GB	250GB SSD, 2TB HDD @7200rpm	Eraldi graafikakaart

IV. Helifailitööluse ressursikulu katsete tulemused

Arvuti	Helifaili suurus (MB)	Helifaili pikkus (s)	Ligikaudne keskmine CPU koormus (%)	Ligikaudne keskmine muutmälu kulu (MB)	Ajakulu (s)
Sülearvuti	4.27	220	8	300	5
Sülearvuti	4.27	220	7	400	5
Sülearvuti	4.27	220	8	300	5
Lauaarvuti	4.27	220	3	300	5
Lauaarvuti	4.27	220	4	400	5
Lauaarvuti	4.27	220	3	400	5
Sülearvuti	21.5	1451	10	800	18
Sülearvuti	21.5	1451	11	1100	14
Sülearvuti	21.5	1451	10	1100	14
Lauaarvuti	21.5	1451	4	1400	12
Lauaarvuti	21.5	1451	5	1200	12
Lauaarvuti	21.5	1451	5	1100	12

V. Videofailitötluse ressursikulu katsete tulemused

Arvuti	Videofailide resolutsioon	Videofailide kaadrisagedus (fps)	Videofailide kogupikkus (s)	Ligikaudne keskmine CPU koormus (%)	Ligikaudne keskmine muutmälukulu (MB)	Videoanalüüsi ajakulu (s)	Ligikaudne keskmine ajakulu ühe sekundi analüüsimiseks (s)
Sülearvuti	480p	30	171	21	250	56	0.33
Lauaarvuti	480p	30	171	7	300	65	0.38
Sülearvuti	1080p	30	171	41	350	147	0.86
Lauaarvuti	1080p	30	171	13	450	132	0.77

VI. Tagasisideküsitluse lisainfo ja küsimused

Tagasisidevorm loodi keskkonnas Google Forms. See ei käsitlenud vastanute isiklikku informatsiooni, aga soovi korral said vastajad lisada oma eesnime. Huvi korral võib küsida töö autorilt ligipääsu tagasisidevormile või selles kasutatud videotele.

Tagasiside Recap Assistant for DaVinci Resolve tarkvara töötulemustele

Aitäh, et jagate tagasisidet Lars Jesper Elleri lõputöö raames - Töö pealkirjaga "Pildi- ja helitöötlusteekidel põhinev tarkvara videote monteerimise automatiseerimiseks" tulemus on tarkvara nimega Recap Assistant for DaVinci Resolve.

Tarkvara palub kasutajalt ühte helifaili ning ühte või rohkemat videofaili ning seejärel kasutab masinõppelahendusi, et tuvastada heli tempo ja videotest leitud põnevad hetked. Seejärel valib kasutaja video monteerimisstiili ning tarkvara väljastab skripti, mille sisselugemine DaVinci Resolve monteerimistarkvaras video automaatselt kokku paneb.

Järgnev tagasisideküsitlus koosneb kolme selle tarkvara poolt kokku pandud video vaatamist, nende seast lemmiku monteerimisstiili valimist ning seejärel ühe video võrdlemist inimese kokku pandud versiooniga. Kõik videod on 31 sekundit pikad, seega kogu ajakulu videote vaatamisele on veidi üle 2 minuti. Suur tänu juba ette!

Tagasiside 1. osa - kolme eri stiilis videoga tutvumine ja eelistuse valimine

Palun vaadake läbi järgmised kolm videot.

Relaxed ehk rahulikumas stiilis video; *Standard* ehk standardstiilis video; *Fast-paced* ehk kiiremas tempos video.

1.1 Millises stiilis video meeldis Teile kõige rohkem?* (* - kohustuslik)

- Relaxed
- Standard
- Fast-paced
- Meeldisid mitu
- Ei meeldinud ükski

1.2 Kui valisite "Meeldisid mitu" või "Ei meeldinud ükski", palun põhjendada: ____

Tagasiside 2. osa - inimese tehtud videoga tutvumine, võrdlemine

Palun vaadake läbi allolev inimese valmistatud video.

Inimese tehtud video

2.1 Kui kvaliteetne on Teie arust tarkvara abil tehtud video võrreldes inimese tehtud videoga?*

- 1 - Tarkvara abil tehtud video on palju ebakvaliteetsem kui inimese tehtud video
- 2
- 3
- 4
- 5 - Tarkvara abil tehtud video on inimese tehtud videoga võrreldes sama kvaliteetne

2.2 Võrrelge inimese tehtud videot mõne tarkvara abil tehtud videoga. Kumb meeldis rohkem, mis olid ühe tugevused üle teise, milline oli kokkuvõttes meeleolukam, jne.* ____

Tagasiside 3. osa - üldised küsimused

3.1 Kokku pandud videod koosnesid neljateistkümnest originaalklipist, mille resolutsioon oli võrdlemisi madal (480p) ja kollektiivne pikkus 5 minutit. Kokku leidis tarkvara analüüsimise käigus ligi pooleteise minuti jagu materjali. Kaua oleksite valmis ootama tarkvara analüüsi lõppemist, et tarkvara kasutamine oleks Teile mugav ning väärtuslik?*

- Vähem kui 10 sekundit
- Vähem kui 1 minut
- Vähem kui 5 minutit
- Vähem kui 15 minutit
- Vähem kui 30 minutit
- 30+ minutit

3.2 Kas Teil oleks huvi seda programmi tulevikus kasutada?*

- Jah
- Ei
- Ei tea

3.3 Palun eelmist vastust lühidalt põhjendada: ____

3.4 Kui on lisakommentaare või mõtteid seoses tarkvara, selle tulemusel valminud videote või muu tööga seonduva osas, siis lisage nad siia: ____

VII. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Lars Jesper Eller,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Tarkvara videote automaatseks monteerimiseks programmiga DaVinci Resolve, mille juhendaja on Sven Aller, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Lars Jesper Eller

15.05.2025