

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

**VIDEO FAILU SATURA APSTRĀDE MEKLĒŠANAS
VAJADZĪBĀM**

BAKALaura DARBS

Autors: **Artjoms Latiševičš**

Studenta apliecības Nr.: al15057

Darba vadītājs: M. dat. Krišjānis Nesenbergs

RĪGA 2019

ANOTĀCIJA

Bakalaura darba mērķis ir izveidot algoritmu, kas dod iespēju meklēt video, nezinot to nosaukumus. Algoritms ir noderīgs situācijās, kad jāatrod līdzīgi video vai pilnu video (piemēram, filmu) pēc lietotāja rīcībā esošā fragmenta vai scēnas.

Bakalaura darbā tiek apskatītas un novērtētas eksistējošās video indeksācijas un meklēšanas metodes un iespējas. Tad tiek piedāvāts autora risinājums video meklēšanai pēc satura. Risinājums tiek īstenots prototipa veidā, kas sadala video attēlu virknē un, izmantojot datorredzes algoritmu, veido atzīmju indeksu priekš video. Meklēšana tiek veikta pēc šī indeksa. Programmas tiek rakstītas C# programmēšanas valodā. Tiek iegūts algoritms, kas spēj atrast līdzīgus video pēc padotā video. Video ar vislielāko atzīmju sakritību skaitu parādās kā pirmie rezultāti. No lietotāja skatupunkta tas darbojas līdzīgi tam, kā tagad darbojas attēlu meklēšana pēc padotā attēla, kas ir īstenots Google meklētājā, vai arī kā darbojas Shazam mūzikas meklēšana pēc ierakstītā fragmenta. Atšķirībā no eksistējošām metodēm, algoritms apvieno sevī visas šīs īpašības: strādā ar jebkura satura video, ir viegli ieviešams un pielāgojams.

Atslēgvārdi: video, meklēšana pēc satura, objektu atpazīšana, datorredze, C#, indekss

ABSTRACT

VIDEO FILE CONTENT PROCESSING FOR LOOKUP NEEDS

The aim of bachelor's paper is to create an algorithm, which allows one to search videos without knowing their titles. Algorithm is useful in situations, where it is necessary to find similar videos or full video (e.g. a movie) based on a video excerpt or a scene in possession of user.

The bachelor's paper examines and rates existing video indexing and lookup methods and options. Then author's solution to video lookup by contents is proposed. The solution is implemented in the form of a prototype, which segments a video to a sequence of images and creates tag index for the video using the computer vision algorithm. Lookup is done using this index. Programs are written using C# programming language. An algorithm, which can find similar videos to the given video, is obtained. Videos, that have the most tag matches, are shown as first results. From the point of view of a user, it works similarly to how image lookup by an image is implemented in Google search engine, or how Shazam looks up music by a recorded excerpt. Unlike existing methods, the algorithm combines all these features: working with video of any content, easy to implement and adaptive.

Keywords: video, lookup by contents, object recognition, computer vision, C#, index

SATURA RĀDĪTĀJS

DEFINĪCIJAS UN APZĪMĒJUMI	6
IEVADS	7
1. Esošā situācija	9
1.1. Problēma.....	9
1.2. Esošo metožu apraksts.....	9
1.3. Video meklēšana pēc autora pievienotas meta informācijas	9
1.4. Video meklēšana pēc satura	10
1.4.1. Indekss video saturam	10
1.4.2. Metodes, kas izmanto temporālo informāciju	11
1.4.2.1. Objektu atsekošana laikā	11
1.4.2.2. Kameras notikumu atsekošana	12
1.4.3. Metodes, kas izmanto kadru secību	12
1.4.3.1. Atslēgu kadri	12
1.4.3.2. Mozaīka	13
1.4.3.3. Scēnu pirkstu nospiedumi.....	16
1.4.4. Atsevišķu kadru apstrāde.....	17
1.4.4.1. Kadru krāsu vērtību salīdzināšana.....	17
1.4.4.2. Kontūru atpazīšana	19
1.4.4.3. Datorredze	20
2. Piedāvātā meklēšanas pēc satura metode	23
2.1. Video indeksācijas paņēmienu analīze	23
2.2. Video indeksācijas paņēmiena izvēle	23
2.3. Indeksa ģenerēšanas metodes definīcija	23
2.3.1. Indekss katram paņemtajam kadram	25
2.3.2. Indekss katrai scēnai.....	26
2.3.3. Indekss visam video	27
2.4. Meklēšanas metodes definīcija.....	28

2.4.1. Meklēšana pa indeksu katram paņemtajam kadram vai katrai scēnai	28
2.4.2. Meklēšana pa indeksu visam video	29
2.4.3. Atzīmju kopu salīdzināšana	29
3. Meklēšanas programmas prototipa izstrāde	31
3.1. Prasības prototipam	31
3.2. Prototipa realizācija	31
3.2.1. Indeksa ģenerēšanas metodes izvēle.....	31
3.2.2. Video fragmentu sagatavošana indeksa ģenerēšanai.....	31
3.2.3. Indeksa ģenerēšana	32
3.2.4. Meklēšana.....	33
REZULTĀTI	40
SECINĀJUMI	41
IZMANTOTĀ LITERATŪRA	42
Pielikumi.....	46
1. pielikums. Atzīmju priekš video kadriem iegūšanas programmas pirmkods.....	46
2. pielikums. Video meklēšanas pēc satura programmas pirmkods.....	50

DEFINĪCIJAS UN APZĪMĒJUMI

Indekss – atslēgvārdu saraksts un saistīti dati, kuri norāda uz vairāk aptverošu informāciju, piemēram, datnes un ieraksti uz diska vai ieraksta atslēgas datubāzē[1].

Meta informācija – informācija par informāciju, informācija, kas tiek lietota, lai atrastu citu informāciju[2].

Temporāls - attiecīgs pret laiku[2].

Pikselis - vismazākais attēla elements, kam rastrgrafikā var tikt patstāvīgi piešķirti tādi raksturojumi kā, piem., krāsa un spilgtums[1].

Orientēto gradientu histogramma (Histogram of Oriented Gradients) – īpašību deskriptors, kas lietots attēlu apstrādei objektu atrašanas mērķiem[3].

Datorredze – datoru un citu mašīnu apveltīšanas ar iespēju redzēt zinātne[4]. Šī darba ietvaros ar datorredzi tiek apzīmēts algoritms, kas spēj, saņemot grafisku materiālu, kā rezultātu atgriezt tajā attēloto objektu un vides raksturojumu teksta veidā. Šāds algoritms ir mašīnmācīšanas[5] attīstības rezultāts[6], kas pēdējā laikā tiek veiksmīgi ieviests[7, 8, 9, 10].

Scēna – spēles, filmas vai grāmatas daļa, kurā notikumu virkne notiek vienā vietā[17]. Šī darba ietvaros ar scēnu tiek apzīmēts nepārtraukti filmēts video fragments.

Atzīme – marķieris, ko var piemērot saturam vai vienumiem (piemēram, fotoattēliem vai tekstam), lai noteiktu identificētu informācijas veidus. Tas ļauj lietotājam viegli atrast, apskatīt un šķirot iezīmētos vienumus[1].

IEVADS

Mūsdienās paliek arvien vairāk un vairāk video satura – filmas, video blogi, informatīvie video un citi. Video satura īpatsvars internetā pārsniedz 75% un arvien pieaug[11]. YouTube vietnē katru minūti tiek augšupielādētas 300 stundas video[12], kopā YouTube ir augšupielādēti 1,300,000,000 video[13]. Atcerēties vai pat zināt nosaukumus vai paļauties uz video autoru pievienotajām atzīmēm paliek arvien grūtāk, taču vajadzība atrast vajadzīgos video tikai pieaug – nepieciešams ātri piemeklēt piemērotu video priekš projekta, atrast filmu pēc scēnas, kas iekļauta citā video, vai pārbaudīt video uz dublikātu vai plaģiātu. Autors uzskata, ka gala lietotājiem pieejamās iespējas meklēt video pašlaik ir nepietiekamas.

Bakalaura darbā tiek apskatītas eksistējošas meklēšanas pēc satura metodes un iespējas un, balstoties uz iegūtās informācijas, piedāvāts alternatīvs risinājums, kas, atšķirībā no eksistējošiem, ir vairāk universāls (nav specifisks priekš video tematikas, ir pielāgojams un efektīvāk lietojams). Autora piedāvātais risinājums dod iespēju meklēt video pēc satura, izmantojot datorredzes attīstības un pieejamības priekšrocības – padodot video fragmentu vai uzskaitot meklējamajā video sastopamos objektus cilvēku valodā. Šāds ieviešums var stāvēt līdzās analogiskiem algoritmiem attēlu meklēšanai pēc padotā attēla[14] un mūzikas meklēšanai pēc ierakstītā fragmenta[15].

Gala lietotājiem pieejamās video meklēšanas iespējas šobrīd ir pēc nosaukuma un video autora pievienotajām atzīmēm, kā tas ir realizēts vietnēs YouTube[16], Google vai Bing. Tas nozīmē, ka bez efektīvām meklēšanas metodēm ievērojama daļa no informācijas pazūd, jo nosaukums un meta dati dod tikai virspusēju informāciju par video. Ir pieejami četri meklēšanas pēc satura varianti – objektu atsekošana laikā[18], pēc atslēgu kadriem[19], lietojot scēnu mozaīkas[20] un izmantojot scēnu pirkstu nospiedumus[21]. Šie varianti, iespējams, tiek lietoti uzņēmumu ietvaros, taču nav pieejami gala lietotājam. Visi šie meklēšanas pēc satura varianti paļaujas uz to, ka scēnas nebūs modificētas; ne visi saturiski vienādi video ir identiski: video var būt modificēts (pagriezts, atspoguļots, nogriezts vai ar izmainītu kontrastu vai krāsām u.c.) vai arī nofilmēts tajā pašā vai līdzīgā vietā un laikā. Tādas izmaiņas radīs traucējumus meklēšanā. Eksistē arī attēlu indeksācijas metodes, kuras iespējams adaptēt video meklēšanai pēc satura.

Bakalaura darba turpmākās daļas ir problēmas identificēšana (Nodaļa 1.1), eksistējošo video indeksācijas un meklēšanas metožu un iespēju apskatīšana un novērtēšana (Nodaļas 1.2 – 1.4), autora piedāvātais risinājums (Nodaļas 2 un 3) un izstrādātā algoritma rezultātu analīze.

Tiek iegūts algoritms, kas spēj atrast līdzīgus video pēc padotā video vai pēc redzamā uzskaitījuma.

1. ESOŠĀ SITUĀCIJA

1.1. Problēma

Bez efektīvām video meklēšanas metodēm ievērojama daļa no informācijas pazūd. Eksistējošās meklēšanas pēc satura metodes netiek ieviestas gala lietotājiem pieejamajās sistēmās, kā arī tām piemīt trūkumi, kas turpmāk darbā tiek apskatīti.

Darba mērķis ir aplūkot eksistējošās video meklēšanas iespējas un piedāvāt efektīvu meklēšanas pēc satura pieeju, izvairoties no eksistējošo metožu trūkumiem.

1.2. Esošo metožu apraksts

Eksistējošās metodes tiek strukturētas šādi: vispirms tiek aplūkota metode, kas izmanto manuāli pievienoto meta informāciju, tad metodes, kas izmanto video satura informāciju – vispirms tās, kuras operē ar temporālo informāciju, tad tās, kuras apstrādā kadru secību, un beidzot atsevišķu kadru apstrādes iespējas. Tādā veidā metodes tiek apskatītas, sākot no tām, kuras abstrahējas no video kadriem, un beidzot ar tām, kuras strādā ar kadru kā ar vienumu.

1.3. Video meklēšana pēc autora pievienotas meta informācijas

Visvienkāršāk ieviešama meklēšanas metode ir pēc nosaukuma un video autora pievienotajām atzīmēm un apraksta, paļaujoties, ka video autors izvēlējās saturam piemērotu nosaukumu un atzīmes. Video saturs netiek ņemts vērā. Šādas metodes ieviešanai nav nepieciešams apstrādāt video. Meklēšana pēc autora pievienotas meta informācijas tiek izmantota populārās vietnēs, piemēram YouTube[16].

Ne nosaukums, ne meta informācija praksē neapraksta video pilnīgi, jo YouTube neļauj veidot nosaukumus garākus par 100 rakstzīmēm, apgriežot meklēšanas rezultātu nosaukumus līdz 70 rakstzīmēm, un aprakstus – par 5000 rakstzīmēm, apgriežot uzreiz redzamo daļu līdz 150[22,23], tāpēc daļa no vēlamo video var nekad netikt atrasti. Video meta informācija var būt arī aplama – video autors pilnīgi to nosaka (piemēram, nosaukumā var būt tas, kas neparādās video, vai arī video autora subjektīvais viedoklis).

1.4. Video meklēšana pēc satura

Video meklēšanas pēc satura metodes ir piemērotas meklēšanai pēc video fragmenta un/vai manuālai meklēšanai. Meklējot pēc video fragmenta, tiek padots video fragments, un kā rezultāti tiek atgriezti atbilstoši video. Meklējot manuāli, tiek ievadīta informācija (atkarībā no metodes tā ir trajektorija, krāsa, atzīmes u.c.), un kā rezultāti tiek atgriezti atbilstoši video, vai arī lietotājs caurskata video indeksu un pats nosaka, vai tekošais video ir vēlamais. Autoru interesē meklēšana pēc video fragmenta, jo tas ir pilnīgi automatizēts process un spēj dot precīzākus rezultātus, jo padotais video fragments sniedz precīzus meklēšanas kritērijus. Metodes, kas spēj apvienot meklēšanu pēc video fragmenta un manuālu meklēšanu, tiek uzskatītas par labākiem kandidātiem autora risinājuma bāzēšanai, jo tās ir universālas. Metodes, kuras iespējams pielāgot dažādām vajadzībām, arī ir labi kandidāti, jo dod darbību brīvību.

1.4.1. Indekss video saturam

Lai būtu iespējams veikt video meklēšanu pēc satura, ir nepieciešams indekss šim saturam. Tiek piedāvāti trīs galvenie informācijas kanāli, pēc kuriem iespējams indeksēt: vizuālā, dzirdes un tekstuālā informācija[24]. Video satur vismaz vizuālo informāciju, bet ne katrā video ir audio (dzirdes informācija) vai uzraksti (tekstuālā informācija). Darbs turpmāk fokusēsies uz vizuālās informācijas indeksācijas.

Vizuālās informācijas indekss tiek veidots no atsevišķiem kadriem, kadru kopuma vai atslēgas kadriem. Veidojot indeksu no atsevišķiem kadriem, tiek ņemti vai nu visi kadri, vai nu katrs n-tais kadrs. Kadru kopuma indekss raksturo visiem kadriem kopīgās piemītošās īpašības. Atslēgas kadri šī darba ietvaros ir tie, kuri ir vistuvākie blakusesošo kadru īpašību vidējām vai biežāk sastopamajām vērtībām. Vizuālās informācijas indekss mēdz sastāvēt no kadriem, vai no informācijas, kas izrēķināma no video kadriem vai kadru apgabaliem.

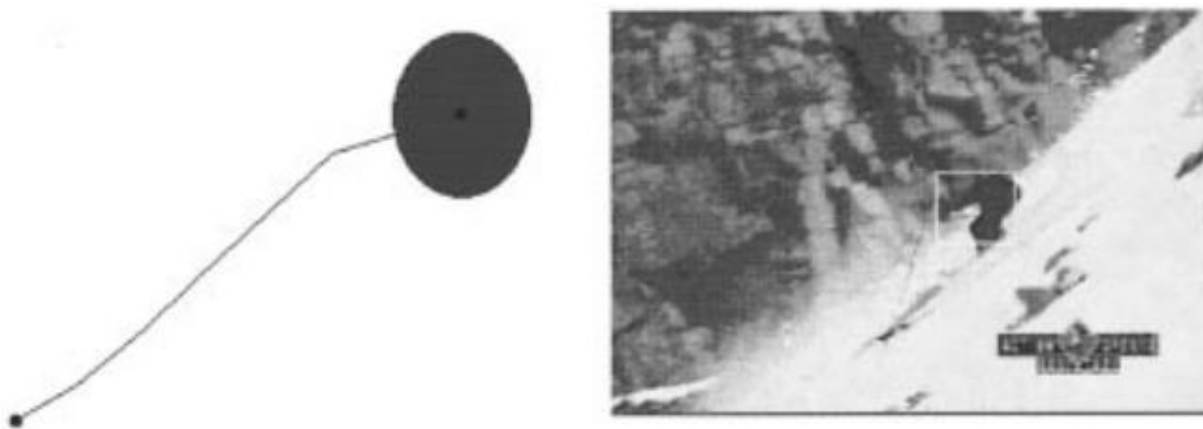
1.4.2. Metodes, kas izmanto temporālo informāciju

Temporālā informācija ir attiecināma uz kadru kopumu un ir saistīta ar laika plūdumu. Šī informācija ir abstrahēta no video veidojošiem kadriem. Šajā apakšnodaļā aprakstītās metodes kā indeksēšanas vienumu izmanto informāciju, kura ir kopīga laika intervāliem.

1.4.2.1. Objektu atsekošana laikā

Objekts šajā apakšnodaļā ir jebkurš vienums, kuru iespējams atsekot pa tā pārvietošanās trajektoriju. Indeksēšanas vienums ir objekta pārvietošanās trajektorija. Šis vienums nevar būt lielāks par scēnu, jo kustība var būt nepārtraukta tikai vienas scēnas ietvaros.

Meklēšana pēc scēnās attēlotajiem objektiem un to trajektorijām[18] aprakstīta sekojoši: Indeksa vienums ir objekta pozīcijas kadrā, kuras veido trajektoriju, pa kuru šis objekts pārvietojas. Katrai scēnai var būt vairāki objekti, ko atseko. Video meklēšana notiek pēc ievadītām objektu trajektorijām, ko lietotājs zīmē. Attēla 1.1. kreisajā pusē ir ievadīta objekta trajektorija, labajā pusē ir tai atbilstošais video.



1.1. att. Trajektorija un tai atbilstošais video[18]

Sistēma tika veidota sporta notikumu video indeksācijai. Ja video fragmenti tiek modificēti (griezti, nogriezti, pagriezti vai izstiepti), vēlamie video var netikt atrasti. Piemēram, ja scēnā objekts pārvietojas pa kvadrāta perimetru vienu reizi, un šai scēnai tiek nogriezta puse, objekta trajektorija vairs nebūs kvadrāts. Ja scēna tiek saīsināta, objekts nespēj veikt pilnu trajektoriju. Ja video tiek izstiepts, trajektorija ir taisnstūris, nevis kvadrāts. Ja video tiek pagriezts, trajektorija kļūst par rombu. Visos šajos gadījumos video netiks atrasti, ja meklēšanai tiek padota kvadrāta trajektorija. Metode der gan meklēšanai pēc padotā video fragmenta, gan pēc lietotāja ievadītajiem parametriem.

1.4.2.2. Kameras notikumu atsekošana

Kameras notikumi ir kameras kustība ap jebkuru savu asi, tuvināšana, tuvplāns, panorāma. Katra scēna iedalāma šajās kategorijās. Tādā veidā video sastāv no kameras notikumu virknes.

Uz kameras notikumiem bāzēta indeksēšanas metode[25] aprakstīta šādi: tiek pielietots seju atpazīšanas algoritms, kura rezultāti tiek izmantoti tuvplānu noteikšanai, balstoties uz sejas izmēru attiecībā uz visu kadru, kameras attālumu, balstoties uz seju daudzumu un pozīcijām, un kustību, balstoties uz seju pārvietojumiem. Papildus tiek izmantoti informatīvi logi un uzraksti, lai noteiktu brīžu svarīgumu.

Metode ir veidota futbola spēļu un ziņu video indeksēšanai. Tā ir noderīga gadījumos, kad nepieciešams atrast video ar konkrētiem kameras notikumiem. Taču ne visos video ir cilvēki, kuri šajā metodē spēlē nozīmīgu lomu kameras notikumu noteikšanā. Tādas video modifikācijas, kā krāsu un kontrasta izmaiņas un izstiepšana, neveidos traucējumus, jo tās neietekmē kameras notikumus, savukārt griešana, nogriešana, un pagriešana var novest pie citām kameras notikumu interpretācijām: sāsinot scēnu, daļa no notikumu neparādās; malu nogriešanas rezultātā scēnā var vairs neparādīties sejas; savukārt pagriešana maina interpretējamo kameras kustības virzienu. Metodi iespējams izmantot meklēšanai pēc padotā video fragmenta un, ievadot vēlamos kameras notikumus kā kritērijus.

1.4.3. Metodes, kas izmanto kadru secību

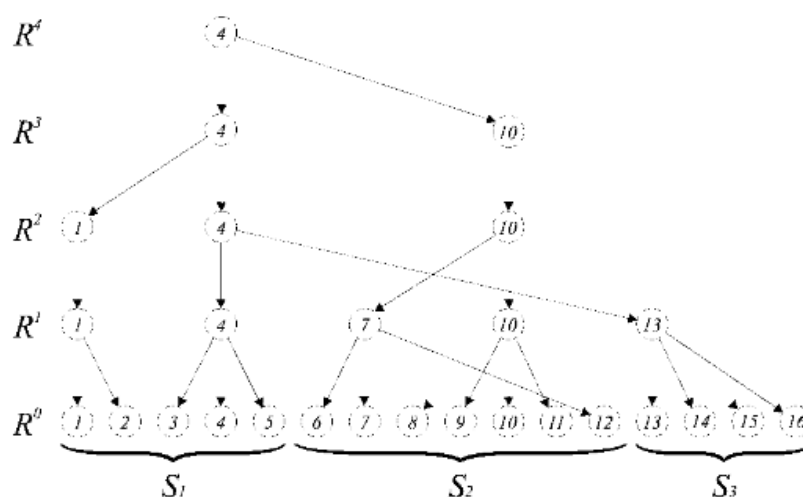
Šīs apakšnodaļas metožu indeksēšanas vienums ir kadru kopums vai atslēgas kadrs pār šo kopumu.

1.4.3.1. Atslēgu kadri

Atslēgu kadri ir kadru kopums, kas katrs reprezentē attiecīgu video apgabalu. Jo līdzīgāki ir apgabala kadri attiecīgajam atslēgas kadram, jo lietderīgāks atslēgas kadrs ir meklēšanā, jo tas dod labāku priekšstatu par apgabala attēlotu.

Atslēgu kadru izvilšanas metode[19] strādā šādi: Atslēgu kadri veido kokveida struktūru, kas kalpo par video satura raksturojumu; jo dziļāk, jo precīzākam momentam

atslēgas kadrs atbilst. Koks tiek veidots, vispirms paņemot vienu kadru, tad scēnas ietvaros atrod tādu kadru, kurš ir no tā visatsšķirīgākais. Process tiek atkārtots apgabalā starp šiem diviem kadriem. Šādai atslēgu kadru struktūrai pielieto meklēšanu dziļumā. Atslēgu kadru koka piemērs ir parādīts attēlā 1.2. Piemērā ir trīs scēnas; ejot pa koku, var nonākt līdz precīzam momentam. R^x apzīmē detalizācijas līmeni, kur atslēgas kadrs R^4 līmenī raksturo visu video fragmentu, bet R^0 – konkrētu kadru; S_x ir scēna; numuri atbilst kadru numuriem pēc kārtas.

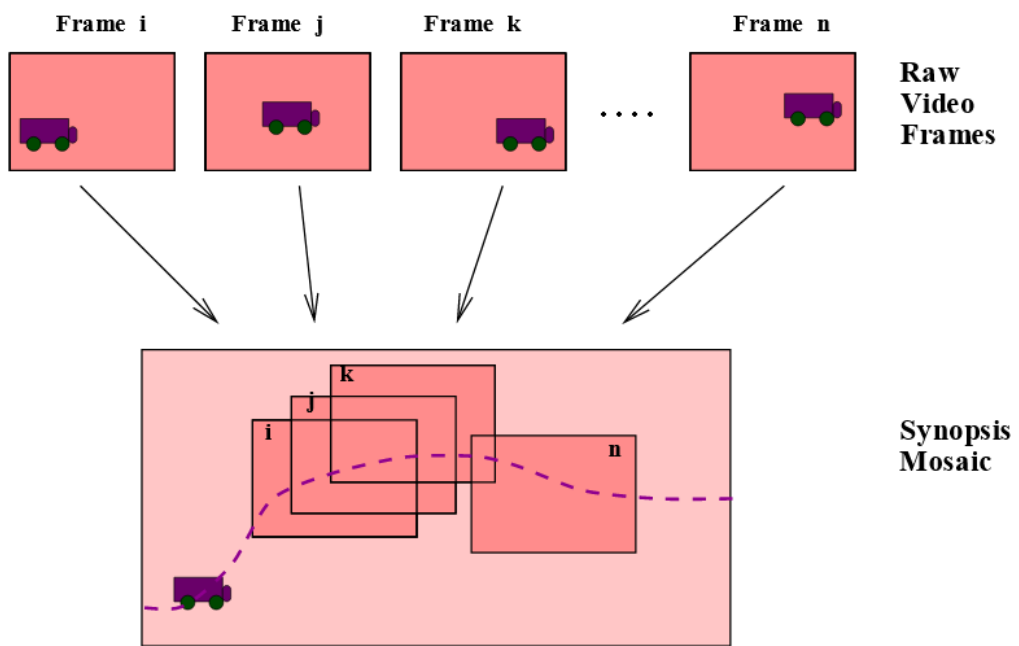


1.2. att. Atslēgu kadru koks[19]

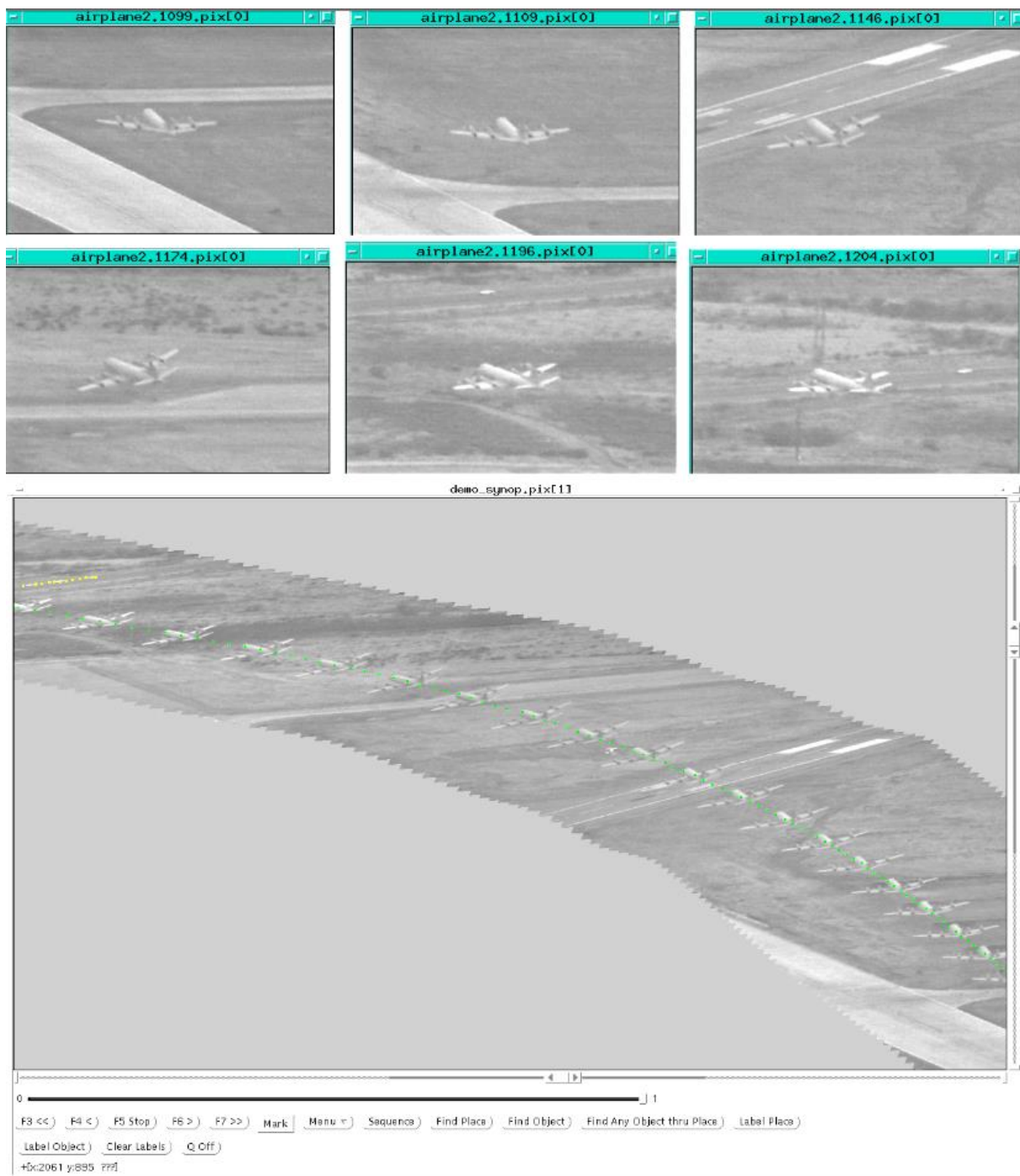
Šāda pieeja ļauj paātrināt meklēšanu, jo salīdzināmo kadru skaits ir neliels (pie nosacījuma, ka meklēšanas algoritms spēj noteikt, vai ir vērts meklēt tālāk konkrētajā koka zarā). Meklēt iespējams gan pēc padotā video fragmenta, gan cilvēkam staigājot pa atslēgu kadru koku.

1.4.3.2. Mozaīka

Mozaīka[20] ir attēls, kas sastāv no visas scēnas kadru salikuma. Tā parāda konkrētas scēnas ietvaros redzamo un notiekošo – statiski objekti paliek nemainīgi, bet kustīgi parādās vairākkārt, saskaņā ar trajektoriju, pa kuru tie kustas. Indekss no mozaīkām ir veidots tam, lai cilvēks varētu, ejot cauri mozaīkām, ātri atrast interesējošo scēnu.



1.3. att. Mozaikas veidošana[20]



1.4. att. Atsevišķi kadri un mozaīka[20]

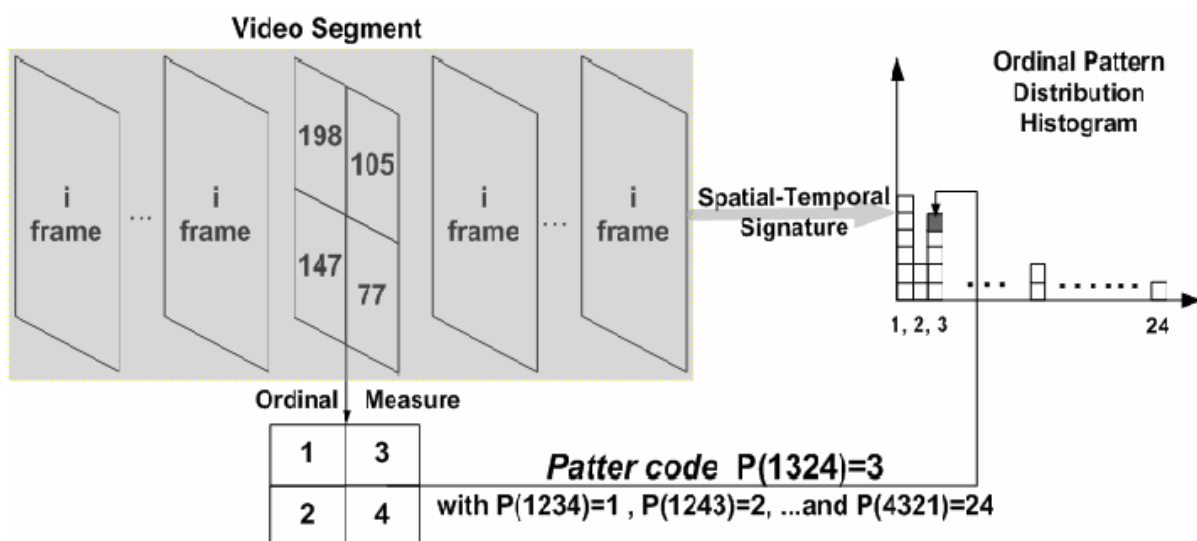
Scēnas kadru kombinēšana vienā mozaīkā tiek parādīta attēlā 1.3. (i, j, k, n ir kārtējie kadri) – nekustīgs fons un objekti tiek anulēti, atstājot kustīgos objektus, kas parādās tik daudz reižu, cik ir kadru. Rezultējošais attēls parāda visu, kas redzams scēnas ietvaros, kamerai griežoties ap jebkuru savu asi un/vai pārvietojoties telpā. Mozaīkas uzlabojums[20] ietver arī informāciju par kustīgajiem objektiem, parādot to trajektorijas. Mozaīkas piemērs ir parādīts attēlā 1.4. Lidmašīnas kustība ir marķēta ar zaļiem punktiem, mašīnas kustība – ar dzelteniem.

Metode der gadījumos, kad cilvēkam vajag ātri gūt sapratni par video notiekošo, taču nav piemērota automatizētai meklēšanai.

1.4.3.3. Scēnu pirkstu nospiedumi

Scēnas pirkstu nospiedums ir lielumu kopa, kura ir izrēķināma no scēnas kadriem. Tā tiek izmantota kā indeksēšanas vienums. Salīdzinot scēnu pirkstu nospiedumus, nosaka, vai scēnas ir vienādas.

Scēnu pirkstu nospiedumu metode[21] aprakstīta šādi: Visu scēnu laika un telpas īpašības (scēnas pirkstu nospiedumi) tiek ierakstītas kokveida struktūrā un saglabātas. Scēnas pirkstu nospiedumu rēķina sekojošā veidā: katrs kadrs scēnā tiek sadalīts četrās daļās; katrai daļai tiek izrēķinātas Y, Cb, Cr krāsu kanālu vidējās vērtības; šo vērtību sakārtojuma virknēm atbilst kodi; šos kodus grupējot kopā, sanāk scēnas pirkstu nospiedums (sk. 1.5. attēlu; “i frame” ir kārtējais scēnas kadrs). Meklējamajai scēnai arī tiek izrēķināts pirkstu nospiedums, tad pa iepriekš izveidoto koku meklē atbilstošu pirkstu nospiedumu.



1.5. att. Scēnas pirkstu nospieduma izrēķināšana[21]

Metode ir piemērota atsevišķu scēnu meklēšanai video. Meklēšana tiek paātrināta, jo salīdzināmais vienums ir scēna, nevis atsevišķi kadri. Ja meklēšanai tiek padots video fragments ar vairākām scēnām un scēnu kārtība video tiek mainīta, tad, lai garantēti varētu atrast atbilstošus video, algoritmam jāmeklē katra scēna atsevišķi. Meklēt var pēc padotā video, bet ne manuāli.

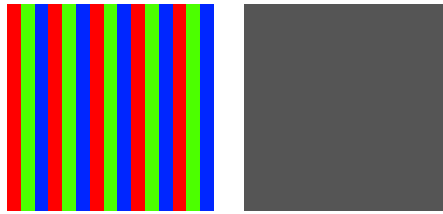
1.4.4. Atsevišķu kadru apstrāde

Metodes šajā apakšnodaļā strādā ar atsevišķiem kadriem kā ar attēliem. Metožu rezultāts ir vērtība vai vērtību kopums vienam attēlam. Indeksa vienums var būt atsevišķs kadrs, kadru kopa, scēna vai viss video, atkarībā no tā, kā šīs metodes adaptē video indeksācijai.

1.4.4.1. Kadru krāsu vērtību salīdzināšana

Eksistē attēlu indeksēšanas pieeja, kas balstīta uz krāsu līdzīguma, kā arī šīs pieejas modifikācijas[26, 27], taču pamatā ideja ir tā pati. Vienkāršības pēc tiek pieņemts, ka tiek izrēķināta krāsu vidējā vērtība visam attēlam. Tad šo pieeju iespējams adaptēt video meklēšanai sekojoši: video sadala attēlu virknē; katram attēlam izrēķina tā visu pikseļu krāsu vidējo vērtību. Meklēšanu veic, padoto video fragmentu tāpat sadalot attēlos un izrēķinot pikseļu krāsu vidējo vērtību; pirmais tiek atgriezts video, kura kadru vidējās vērtības ir tuvākās padotā video fragmenta kadru vidējām vērtībām; nākamie video tiek atgriezti krāsu vidējo vērtību atšķirīguma pieaugšanas secībā.

Šāda pieeja Autora redzējumā ir vienkārši realizējama, taču rezultējošās krāsu vērtību virknes nebūs unikālas priekš video – iespējami gadījumi, kad summu vērtības ir tuvas vai pat vienādas, kaut gan to veidojošo pikseļu vērtības ir ļoti atšķirīgas: pilnīgi zils kadrs dos to pašu rezultātu, kas būs pilnīgi sarkanam kadram. Iespējamais uzlabojums ir nesummēt visu trīs krāsu vidējās vērtības, bet glabāt tās atsevišķi. Bet joprojām divi atšķirīgie kadri, kas redzami attēlos nr. 1.6 un 1.7, dos to pašu rezultātu, jo abos gadījumos katras krāsas vidējā vērtība attēlā būs trešdaļa no maksimālās vērtības. Līdzīgi, attēli ar atšķirīgu krāsu izvietojumu, bet vienādām krāsu proporcijām, dos to pašu rezultātu (attēls nr. 1.8.). Attēlā nr. 1.9. ir redzami reāli iegūtie rezultāti[26], meklējot pēc līdzīgām krāsām – daļa attēlu patiešām saturiski ir līdzīgi, taču ir arī tie, kuri ir līdzīgi tikai krāsu nozīmē. Vēl viens šīs pieejas trūkums ir tas, ka lietotājam vidējās krāsu vērtības nebūs noderīgas – viņš nevarēs ievadīt vidējo krāsu vērtību virkni meklētājā; atliks tikai iespēja meklēt, padodot video fragmentu.



1.6. un 1.7. att. 2 attēli ar vienādu krāsu vidējo vērtību



1.8. att. Attēli ar vienādām krāsu proporcijām[26]



1.9. att. Meklēšanas pēc līdzīgām krāsām rezultāti[26]

Šīs pieejas priekšrocība ir gadījumos, kad video tiek pagriezts, atspoguļots vai izstiepts, jo, neskatoties uz šīm transformācijām, rezultējošais indekss būs tāds pats, kā attēlam pirms transformācijām. Taču, ja no video tiek nogrieztas malas vai pamainītas krāsas un kontrasts, indeksi, kas veidoti no šiem attēliem, var atšķirties pietiekami, lai cits video kļūtu par labāku sakritību, nekā tas pats, bet modificēts video.

1.4.4.2. Kontūru atpazīšana

Orientēto gradientu histogramma[3, 28, 29], tās modifikācijas[30] un krāsu-tekstūru reģionu segmentācija[31] tiek izmantotas formu (kontūru) atpazīšanai attēlos. Orientēto gradientu histogrammas metodes pamatā ir gradienta vektori, kuri raksturo krāsu izmaiņas pikseļu apgabalos gan horizontālā, gan vertikālā virzienā[32]. Orientēto gradientu histogrammas rezultāta vizualizācija redzama attēlā nr. 1.10. Video kadriem iespējams pielietot orientēto gradientu histogrammas algoritmu, tādā veidā veidojot indeksu visam video. Meklēšana veicama, padoto video fragmentu apstrādājot tādā pašā veidā un atgriežot video orientēto gradientu histogrammas vektoru virkņu atšķirības pieaugšanas secībā.



1.10. att. Orientēto gradientu histogrammas vizualizācija[29]

Kontūru atpazīšana katram kadram dotu labāku indeksu, nekā krāsu vidējās vērtības, jo tas jau ir tuvāks attēla saturam. Taču joprojām nav iespēju meklēt manuāli; ir jāpadod video fragments. Nozīmīgs ierobežojums ir arī tas, ka manipulācijas ar video ievērojami apgrūtina meklēšanu, tāpēc ka kontūras atšķirsies, neskatoties uz to, ka saturiski video ir tādi paši: rezultējošais indekss būs atšķirīgs, ja video tiek apgriezts, pagriezts, izstiepts vai atspoguļots, jo tas ir atkarīgs no pikseļu izvietojuma. Toties šī metode ir izturīga pret manipulācijām ar krāsām un kontrastu, jo tās neietekmē kontūrus, uz kuru pamata tiek veidotas kontūru atpazīšanas metodes.

1.4.4.3. Datorredze

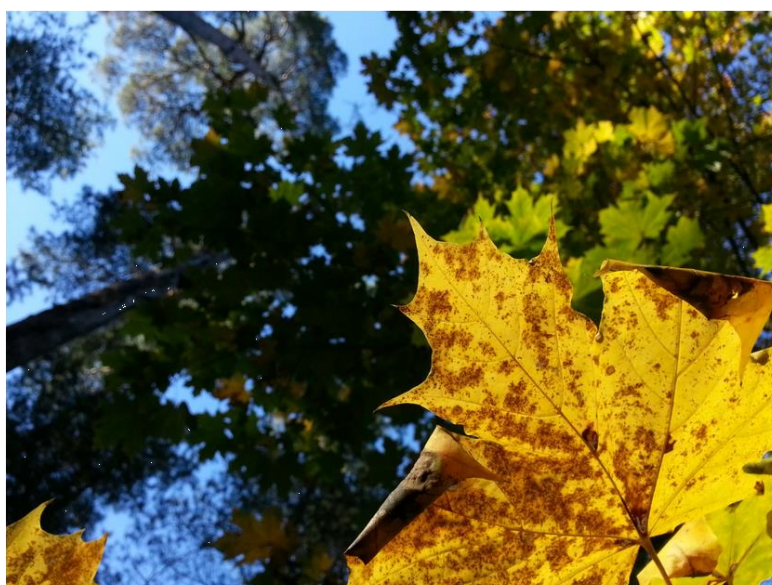
Datorredze dod iespēju veidot attēlā vai video redzamā aprakstu vārdu (atzīmju) vai pat teikumu formā, piemēram Microsoft Azure Computer Vision API[7]. Atzīmēm mēdz būt svāri. Atzīmes svārs ir pārliecība, ka atzīme ir pareiza un ir robežās no 0 līdz 1. Datorredzei nepieciešams daudz resursu un parauga attēlu, uz kuriem to apmācīt. Eksistē tiešsaistes datorredzes servisi[7, 8, 9, 10], kuri apstrādā padotos attēlus; daži no servisiem spēj ģenerēt atzīmes priekš video[9]. Priekšrocība to izmantošanā ir tajā, ka algoritmi tiek apmācīti uz lieliem datu apjomiem, kurus sniedz servisa lietotāji[8], kas ļauj gūt precīzākus rezultātus, kā arī servisu lietotājiem nav nepieciešama jaudīga aparatūra, jo apstrāde notiek uz servisa sniedzēja servera.

Iespējams izmantot kādu no pieejamajiem datorredzes servisiem. Vienkāršībai tiek pieņemts, ka serviss spēj ģenerēt atzīmes priekš attēliem bez atzīmju svāriem. Šie servisi ļauj iegūt informāciju par attēlu atzīmju formā. Tāpat, kā citos variantos, video sadala attēlu virknē un katru padod kā ieeju datorredzes servisam. Šādā veidā katram paņemtajam video kadram būs savu atzīmju kopa, kas īsi apraksta attēlā redzamo. Indeksa vienums var būt kadrs, kadru kopa vai viss video. Ņemot kadru par indeksa vienumu, palielinās meklēšanas precizitāte, jo iespējams salīdzināt kadru pēc kadra, savukārt indekss visam video ir kompaktāks un meklēšana ir ātrāka, jo atzīmes neatkārtojas, taču zūd precizitāte – ilgāks video ar identisku fragmentu meklētajam var tikt uzskatīts par sliktāku sakrītību, nekā cits video, kur nav identiska fragmenta ar meklēto, bet kopumā atzīmes vairāk sakrīt ar meklētā video fragmenta atzīmēm.

Ja ir pieejami atzīmju svāri, meklēšanas precizitāti iespējams uzlabot, jo atzīmes ar svaru zem noteiktas robežas var atņemt, lai nebūtiskas vai aplamas sakrītības neradītu traucējumus (troksni nevēlamu video rezultātu veidā). Traucējumi rodas datorredzes atpazīšanas nepilnību rezultātā – algoritms tiek apmācīts uz attēliem un var izdarīt nepareizus secinājumus par citiem attēliem, jo abos attēlos kopīga ir kāda cita īpašība (piem. apmācīšanai tika doti attēli ar lapsu mežā; tad tiek padots meža attēls – šajā gadījumā var kļūdaini tikt izdota atzīme “lapsa”, jo kopīga īpašība ir mežs).

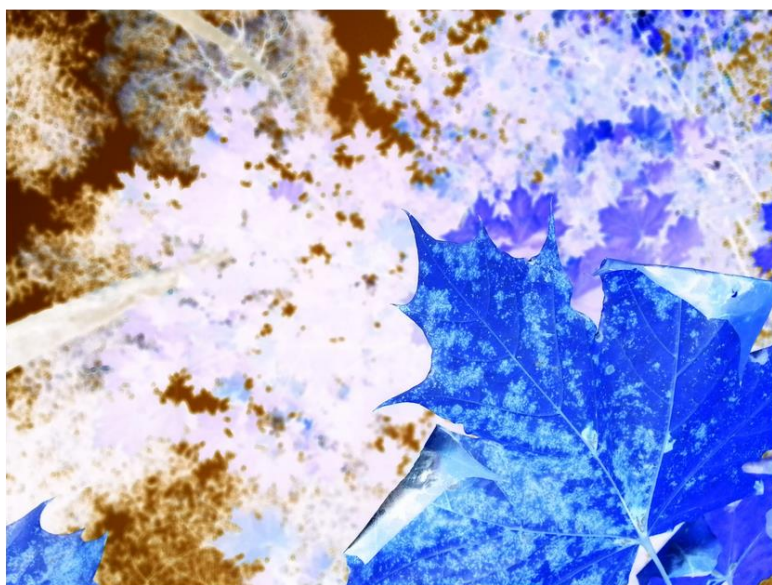
Datorredze ļauj ne tikai izveidot kompakto indeksu, bet arī dod iespēju meklēt manuāli (pārskaitot objektus, kuriem jābūt vēlamajā video). Tādas video transformācijas, kā nogriešana, pagriešana, atspoguļošana, izstiepšana, ātruma izmaiņas, atskaņošana atpakaļvirzienā un krāsu vai kontrasta korekcija, neveidos traucējumus, jo kadri tiek ņemti kā atsevišķi attēli, un datorredzes apmācīšanai tiek izmantoti attēli no dažādām kamerām ar

dažādu kvalitāti, leņķiem un krāsu reprezentāciju. Savukārt krāsu nobīdes vai inversija var traucēt, jo datorredzes algoritmi visticamāk tika apmācīti uz piemēriem bez šādām manipulācijām (Clarifai datorredzes servisa gadījumā apmācīšanai tiek izmantoti lietotāju padotie attēli[8]). Attēlos 1.11., 1.12. un 1.13. ir viena un tā pati bilde (1.11. ir oriģināls, 1.12. ir ar krāsu inversiju un 1.13. ir ar krāsu nobīdi) un atbilstošās atzīmes, ko atgriezta Clarifai serviss. Atzīmes atšķiras, it īpaši tas novērojams krāsu inversijas gadījumā.



PREDICTED CONCEPT	PROBABILITY
fall	0.999
leaf	0.998
no person	0.994
maple	0.990
trees	0.981
nature	0.971
wood	0.970
bright	0.970
fair weather	0.967
outdoors	0.954
gold	0.938
sun	0.937
medialy	0.930
season	0.924
lush	0.908
descending	0.901
color	0.890

1.11. att. Apstrāde ar Clarifai: oriģināls attēls



PREDICTED CONCEPT	PROBABILITY
no person	0.993
winter	0.981
fall	0.978
Christmas	0.976
leaf	0.973
frost	0.966
nature	0.953
bright	0.952
abstract	0.940
artistic	0.939
desktop	0.934
snow	0.933
season	0.902
cold	0.899
outdoors	0.892
whining	0.891
paper	0.888

1.12. att. Apstrāde ar Clarifai: krāsu inversija



PREDICTED CONCEPT	PROBABILITY
leaf	0.990
fall	0.984
no person	0.984
trees	0.951
nature	0.948
color	0.943
texture	0.939
desktop	0.930
bright	0.916
wood	0.913
art	0.912
outdoors	0.910
abstract	0.907
maple	0.906
flora	0.890
season	0.858
flower	0.848

1.13. att. Apstrāde ar Clarifai: Krāsu nobīde

Būtisks ierobežojums datorredzes izmantošanai ir servisu pieejamība – jo lielāks attēlu skaits, jo dārgāk, kā arī ir piemērojami ātruma ierobežojumi[33]. Bakalaura darba rakstīšanas brīdī eksistē daži tiešsaistes servisi, kuri sniedz ierobežotu piekļuvi datorredzes pakalpojumiem. Kā arī datorredzes sniedzamās atzīmes ir visai nenoteiktas – iespējami gadījumi, kad lietotājam būs nepieciešams atrast noteiktu vietu, zīmolu vai personu, taču algoritms spēs tikai noteikt, ka, piemēram, ir redzama upe, bet nepateiks, ka tā ir Lielupe, ja vien algoritms nav speciāli apmācīts atpazīt specifiskas lietas[34]. Lai datorredze spētu atpazīt specifiskas lietas, tā ir jāapmāca uz konkrētiem piemēriem. Daži servisi ļauj to darīt[7, 8, 10], taču pieejamie vispārīgi apmācīti algoritmi sniedz tikai vispārīgas atzīmes (Clarifai algoritms ir apmācīts atpazīt 11 000 konceptu[35]).

2. PIEDĀVĀTĀ MEKLĒŠANAS PĒC SATURA METODEDE

2.1. Video indeksācijas paņēmieni analīze

. Video indeksācijas metodes tika sadalītas grupās; tabulā nr. 2.1 tiek apkopotas metožu grupu īpašības.

2.1. tabula: **Metožu grupu īpašības**

Metožu grupa	Īpašības
Metodes, kas izmanto temporālo informāciju	Izturība pret krāsu un kontrasta izmaiņām. Meklēšana gan automātiski, gan manuāli. Informācija par darbībām.
Metodes, kas izmanto kadru secību	Kadru secība kā indeksa viensums – ātrāka meklēšana un/vai kompakts indekss. Optimizētas meklēšanai automātiski vai manuāli.
Atsevišķu kadru apstrāde	Pielāgojamība – jebkurš indeksa viensums.

Metožu, kas izmanto kadru secību un atsevišķu kadru apstrādes grupas ir labi kandidāti, lai uz to bāzes veidotu risinājumu, jo iespējams sasniegt ātrāku meklēšanu un veidot kompaktāku indeksu. Savukārt metodes, kas izmanto temporālo informāciju, sniedz iespēju meklēt gan automātiski, gan manuāli, kas atbilst autora prasībām. Šī priekšrocība piemīt arī atsevišķu kadru apstrādes grupas datorredzes metodei (Apakšnodaļa 1.4.4.3.).

2.2. Video indeksācijas paņēmiena izvēle

Autors izvēlējās izmantot datorredzes metodi (Apakšnodaļa 1.4.4.3.) indeksa ģenerēšanai. Datorredze ir izturīga pret daudzām video veidojošo kadru modifikācijām, izmantojot datorredzes rezultējošo atzīmju indeksu, iespējams meklēt gan padodot video fragmentu, gan manuāli, ievadot vēlamu objektu nosaukumus, kā arī šī metode ir pielāgojama.

2.3. Indeksa ģenerēšanas metodes definīcija

Video scēnas kadri ir saistīti savā starpā ar tajos attēlotajiem objektiem un nepārtrauktām darbībām. Divi blakusesošie kadri ir līdzīgi viens otram. Lai datorredzes ģenerētās atzīmes kadriem atkārtotos pēc iespējas mazāk (veidojot kompaktāku indeksu), no video ir jāņem pēc iespējas mazākais kadru skaits, nezaudējot informāciju par video redzamo. Tika veikts pētījums[36], saskaņā ar kuru cilvēks spēj atpazīt gandrīz visus objektus, kuri

parādās video, kas tika paātrināts, parādot katru 64-to kadru, tātad aptuveni viens kadrs divās sekundēs (video ir 30 kadri sekundē). Tas nozīmē, ka datorredze sasniegs tādu pašu vai labāku rezultātu, jo viens kadrs divās sekundēs dod nepieciešamo informāciju. Pētījumā tika izmantotas mākslas filmas. Tā kā video saturs var būt dažāds, autors piedāvā ņemt indeksam vienu kadru vienā sekundē. Tas ļauj samazināt apstrādājamo kadru skaitu no 24 līdz 60 reizēm (standarta video ir 24, 29.97, 30, 50, 59.94 vai 60 kadri sekundē[37]). Cits intervāls vai indeksācija, kad saturiski kadrs atšķiras par noteiktu koeficientu, tiek uzskatīti par iespējamajiem uzlabojumiem.

Tiek piedāvātas trīs indeksa detalizācijas pakāpes - katrs paņemtais kadrs, scēna vai viss video kā indeksa viens. Visi trīs varianti indeksē visu video, bet ar dažādu temporālo detalizāciju - katrs paņemtais kadrs ļauj pārbaudīt visu video uz identiskumu, viss video ļauj ātrāk atrast līdzīgus video, scēnas ir kompromiss. Jebkurā gadījumā tiek ņemts viens kadrs vienā sekundē, atšķirības ir tālākajās darbībās.

Indekss turpmāk sastāv no atzīmju kopām. Sākotnēji šī kopa reprezentē paņemto kadru. Atkarībā no detalizācijas pakāpes (kuras aprakstītas zemāk), atzīmju kopa indeksā reprezentē katru paņemto kadru, scēnu vai visu video.

2.3.1. Indeks katram paņemtajam kadram

Indeksa katram paņemtajam kadram veidošanas algoritms ir pieejams tabulā 2.2.

2.2. tabula

Indeksa katram paņemtajam kadram veidošana

Apstrādājama video v
Apstrādājamā video kadru direktorijs d
Tekošais kadrs i = pirmais video kadrs
Atzīmju kopa t
Datnes nosaukums f
Iegūt no v kadru virkni un saglabāt tos d <i>piemēram, ar programmu ffmpeg[38]</i>
Atkārtot
$t =$ Padot i datorredzes servisam un iegūt tam atbilstošas atzīmes
$f = d + i + ".txt"$
Saglabāt t teksta datnē ar nosaukumu f
$i =$ Iegūt nākamo video kadru
Kamēr tekošajā video ir kadri

Veidojot indeksu katram paņemtajam kadram, tiek sasniegta lielākā detalizācijas pakāpe, kas ļauj, salīdzinot kadru pēc kadra, noteikt, vai ir ne tikai attēlotu objektu, bet arī hronoloģiska sakritība. Taču scēnas, kas ir modificētas, atskaņojot atpakaļvirzienā, bet citādi identiskas, var vairs netikt uzskatītas par vienu un to pašu scēnu (piemēram scēnas sākumā garām brauc mašīna; otrādi atskaņotajā scēnā tas notiek beigās). Līdzīgi, ja meklējama video fragments satur scēnas, kuras ir atšķirīgā secībā, tas vairs nesakrītīs ar oriģinālo video. Kā arī meklēšanas laiks ir visilgākais starp piedāvātajām indeksa veidošanas metodēm, jo meklēšana notiek, secīgi salīdzinot visu paņemto kadru atzīmju kopu indeksus.

2.3.2. Indekss katrai scēnai

Panākt kompromisu starp detalizētu un kompaktu indeksu iespējams, pielietojot scēnu atpazīšanas algoritmu. Par scēnu maiņas pazīmi var kalpot pašas atzīmes. Algoritms ir pieejams tabulā 2.3.

2.3. tabula

Indeksa katrai scēnai veidošana

```
Apstrādājamais video  $v$ 
Apstrādājamā video kadru direktorijs  $d$ 
Tekošais kadrs  $i$  = pirmais video kadrs
Skaitītājs  $j = 1$ 
Atzīmju kopa  $t$ 
Atzīmju kopa  $t\_iepriekšējā$ 
Atzīmju kopa  $t\_scēnai$ 
Datnes nosaukums  $f$ 
Iegūt no  $v$  kadru virkni un saglabāt tos  $d$    piemēram, ar programmu ffmpeg[38]
Atkārtot
     $t =$  Padot  $i$  datorredzes servisam un iegūt tam atbilstošas atzīmes
    Ja neviena no pirmajām 3 atzīmēm no  $t$  nav sastopama pirmajās 3 atzīmēs no
     $t\_iepriekšējā$    atšķirības kritērijs var būt arī cits
         $f = d + j + \text{“.txt”}$ 
        Saglabāt  $t\_scēnai$  teksta datnē ar nosaukumu  $f$ 
         $j++$ 
         $t\_scēnai = t$ 
    Citādi
         $t\_scēnai += t$    dublikāti tiek apvienoti
     $t\_iepriekšējā = t$ 
     $i =$  Iegūt nākamo video kadru
Kamēr tekošajā video ir kadri
 $f = d + j + \text{“.txt”}$ 
Saglabāt  $t\_scēnai$  teksta datnē ar nosaukumu  $f$ 
```

Atšķirības pakāpe varētu būt pirmās atzīmes, kuru svars ir lielāks par noteikto vai arī pirmās n atzīmes, ja informācija par svaru nav pieejama, bet ir zināms, ka atzīmes sakārtotas svara dilšanas secībā (konkrēti lielumi atšķirsies dažādiem datorredzes servisiem). Salīdzināt labāk ir iepriekšējā un nākamā kadra atzīmes, nevis visas uzkrātās scēnas atzīmes ar nākamā kadra atzīmēm, jo scēnas ietvaros attēlotais var mainīties pietiekami, lai nevarētu atšķirt nākamās scēnas sākumu (piemēram, vienas scēnas ietvaros kamera rāda pļavu, tad skats tiek pacelts, un kadrā ir tikai debesis; nākamās scēnas pirmais kadrs rāda pļavu). Šāda pieeja negarantēs to, ka video tiks pareizi sadalīts pa scēnām, jo divas blakusesošās scēnas var būt tik līdzīgas, ka to kadru atzīmju kopas neatšķirsies par uzstādīto pakāpi (piemēram, viena un

tā pati vieta filmēta no dažādiem rakursiem). Toties, neskatoties uz scēnu sadalījuma korektumu, tas būs tāds pats gan padotajam video, gan iepriekš indeksētajam.

Lai uzkrātās atzīmes neatkārtotos, dublikāti jāapvieno. Ja ir pieejama informācija par atzīmju svaru, autors piedāvā visiem atkārtoto atzīmju svāriem rēķināt aritmētisko vidējo. Savukārt tās atzīmes, kurām svārs ir mazāks par noteiktu lielumu, un kuras reti parādās kadros (mazāk par noteiktu procentu no visiem kadriem), neiekļaut scēnas atzīmju kopā, tādā veidā mazinot troksni un indeksa lielumu.

Indekss katrai scēnai ir kompaktāks par indeksu katram paņemtajam kadram. Scēnas ietvaros attēlotais tiek grupēts, kas sniedz labāku izturību pret modifikācijām, nekā indekss katram paņemtajam kadram – scēnas, kas ir modificētas, atskaņojot atpakaļvirzienā, šajā gadījumā tiks uzskatītas par identiskām. Uzlabot video fragmentu ar atšķirīgu scēnu sakārtojumu meklēšanas precizitāti iespējams, meklējot pa atsevišķām scēnām, taču tas negatīvi ietekmēs meklēšanas laiku, jo katra padotā video scēna tiek atsevišķi pārbaudīta uz sakritību ar kārtējā video visām scēnām.

2.3.3. Indekss visam video

Visas atzīmes tiek uzkrātas līdz viss video tiks iziets cauri, tad saglabātas. Indeksa visam video veidošanas algoritms ir ieejams tabulā 2.4.

2.4. tabula

Indeksa visam video veidošana

Apstrādājama video v	
Apstrādājamā video kadru direktorijs d	
Tekošais kadrs $i =$ pirmais video kadrs	
Atzīmju kopa t	
Datnes nosaukums $f = d + v + \text{“.txt”}$	
Iegūt no v kadru virkni un saglabāt tos d	<i>piemēram, ar programmu ffmpeg[38]</i>
Atkārtot	
$t +=$ Padot i datorredzes servisam un iegūt tam atbilstošas atzīmes	<i>dublikāti tiek apvienoti</i>
$i =$ Iegūt nākamo video kadru	
Kamēr tekošajā video ir kadri	
Saglabāt t teksta datnē ar nosaukumu f	

Dublikāti tiek apvienoti tāpat, kā aprakstītajā indeksēšanā katrai scēnai.

Šai pieejai ir imunitāte pret scēnu kārtības maiņu, jo viena atzīmju kopa ietver visas scēnas. Indekss visam video drīzāk derēs līdzīgu video meklēšanai, nekā precīzu atbilstību meklēšanai: ja meklējamais fragments ir ietverts kādā video, ir īss salīdzinājumā ar to, un

fragmentā attēlotais nebūs līdzīgs pārējam video saturam, tas video skaitīsies par vāju atbilstību meklējamajam fragmentam.

2.4. Meklēšanas metodes definīcija

Meklēšanas algoritmi atšķirsies atkarībā no tā, kāda ir indeksa detalizācija. Zemāk aprakstīti meklēšanas algoritmi visām augstāk definētajām indeksa detalizācijas pakāpēm.

2.4.1. Meklēšana pa indeksu katram paņemtajam kadram vai katrai scēnai

Katra video paņemtā kadra indeksa vienums ir kadra atzīmju kopa; katras video scēnas indeksa vienums ir scēnas atzīmju kopa. Meklēšana veicama, salīdzinot atzīmju kopas padotā un kārtējā video vienumam, kur vienums atbilstoši ir vai nu kadrs, vai nu scēna. Meklēšanas katra vienuma indeksā algoritms ir pieejams tabulā 2.5.

2.5. tabula

Meklēšana katra vienuma indeksā

Apstrādājama video v
Apstrādājamā video indeksa direktorijs d
Video bibliotēka b
Atzīmju kopa t_1
Atzīmju kopa t_2
Izveidot indeksu priekš v ar Metodi 2.3.1. vai 2.3.2. un saglabāt to d
Atkārtot
$t_1 =$ Padotā video pirmā vienuma atzīmju kopa
$t_2 =$ Kārtējā video pirmā vienuma atzīmju kopa
Atkārtot
Ja kopas t_1 un t_2 sakrīt <i>salīdzināšana aprakstīta apakšsadaļā 2.4.3.</i>
$t_1 =$ Padotā video nākamā vienuma atzīmju kopa
$t_2 =$ Kārtējā video nākamā vienuma atzīmju kopa
Kamēr tekošā video indeksā ir atzīmju kopas un kārtējā video indeksā ir atzīmju kopas
Kamēr b ir video

Lai paātrinātu meklēšanu, iespējams pārtraukt salīdzināšanu, ja atzīmju kopas ir pārāk atšķirīgas (piemēram, padotā video kārtējā vienuma atzīmju kopā neviena atzīme nesakrīt ar salīdzināmā video vienuma atzīmju kopas pirmajām trīs atzīmēm).

2.4.2. Meklēšana pa indeksu visam video

Visa video indeksa vienums ir video atzīmju kopa. Meklēšana veicama, salīdzinot atzīmju kopas padotajam un kārtējam video. Meklēšanas visa video indeksā algoritms ir pieejams tabulā 2.6.

2.6. tabula

Meklēšana visa video indeksā

Apstrādājama video v
Apstrādājamā video indeksa direktorijs d
Video bibliotēka b
Atzīmju kopa t_1 = Padotā video atzīmju kopa
Atzīmju kopa t_2
Izveidot indeksu priekš v ar Metodi 2.3.3. un saglabāt to d
Atkārtot
t_2 = Kārtējā video atzīmju kopa
Salīdzināt t_1 un t_2 <i>salīdzināšana aprakstīta apakšsadaļā 2.4.3.</i>
Kamēr b ir video

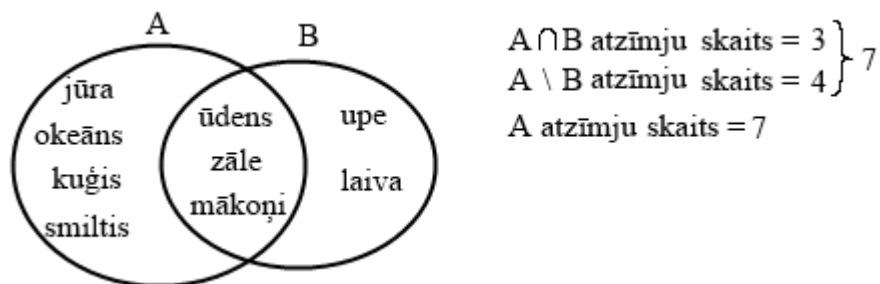
Meklēšana ir ātrāka, nekā meklēšana pa indeksu katram paņemtajam kadram vai katrai scēnai, jo, lai salīdzinātu padoto video ar kārtējo video, tiek salīdzināts tikai viens atzīmju kopu pāris.

2.4.3. Atzīmju kopu salīdzināšana

Meklēšana veicama, salīdzinot padotā un kārtējā video, scēnas vai kadra (tālāk saukts par vienumu) atzīmju kopas. Vienumu līdzīgumu iespējams, veicot atzīmju kopu starpību vai šķēlumu.

Veicot kopu starpību, tiek pierakstītas klāt visas atzīmes, kas ir padotā vienuma atzīmju kopā, bet nav kārtējā vienuma atzīmju kopā. Jo lielāks uzkrāto atzīmju skaits padotajam video fragmentam, jo mazāka ir atbilstība kārtējam video. Atstājot atzīmes, kas ir padotā vienuma atzīmju kopā, nevis kārtējā vienuma atzīmju kopā, tiks sasniegti labāki rezultāti, jo atzīmju kopas satur dažādu atzīmju skaitu, kas saistīts ar to, ka vienumā var būt dažāds skaits aprakstāmu objektu; atstājot atzīmes, kas ir padotā vienuma atzīmju kopā, visās salīdzināšanās atzīmes tiek dzēstas no tās pašas kopas. Veidojot kopu starpību, paliekošo atzīmju skaits nosaka video atšķirīguma pakāpi – mazāks uzkrāto atzīmju skaits nozīmē lielāku atbilstību starp padoto video fragmentu un kārtējo video; kārtējā video nesakrītošu atzīmju skaits netiek ņemts vērā.

Veicot kopu šķēlumu, tiek pierakstītas klāt visas atzīmes, kas ir gan kārtējā vienuma atzīmju kopā, gan padotā vienuma atzīmju kopā. Jo lielāks uzkrāto atzīmju skaits kārtējam video, jo lielāka ir atbilstība padotajam video fragmentam.



2.1. att. Atzīmju kopu operācijas

Gan kopu starpība, gan šķēlums dos to analogisku rezultātu, jo par cik vairāk atzīmju paliek pēc kopu starpības, par tik mazāk paliek pēc kopu šķēluma, kas ir parādīts attēlā nr. 2.1. (A ir padotā video atzīmju kopa; B ir kārtējā video atzīmju kopa).

3. MEKLĒŠANAS PROGRAMMAS PROTOTIPA IZSTRĀDE

3.1. Prasības prototipam

Video meklēšanas pēc satura programmas prototipam jāprot kā pirmo rezultātu atdot nosaukumu identiskam video; visi pārējie rezultāti būs līdzīgi video līdzīguma dilšanas secībā.

Kā ievads prototipam jāspēj saņemt video fragmenta indekss. Prototips to izmantos video meklēšanai un atdos identiskus vai līdzīgus video.

3.2. Prototipa realizācija

3.2.1. Indeksa ģenerēšanas metodes izvēle

Autors izvēlējās izmantot indeksu visam video. Tas ir visātrāk realizējams no visiem trīs variantiem un tajā pašā laikā ļauj gūt priekšstatu par metodes efektivitāti.

3.2.2. Video fragmentu sagatavošana indeksa ģenerēšanai

Tika izvēlēti 19 autora rīcībā esošie video, kas sadalīti pa tematikas grupām. Sadalījums redzams tabulā 3.1. Video tika izvēlēti tā, lai katram video būtu vismaz viens līdzīgs video, kuram vajadzētu parādīties pirmajos rezultātos, meklējot. Video skaits no grupas “Skats pa braucošā autobusa logu” ir vislielākais, lai pārbaudītu, kā vienas tematikas grupas video sadalās meklēšanas rezultātos. Tā kā indekss tiek veidots no attēliem, video tika sadalīti kadru secībā ar ffmpeg[38] rīka palīdzību (lietojums parādīts attēlā 3.1). Autors ņēma kadrus ar vienas sekundes intervālu (parametrs “-r 1” 3.1. attēlā redzamajā komandā).

Video sadalījums pa tematikas grupām

Video fragmenta nosaukums	Tematika
DSC_4199	Koka ziedi tuvumā
DSC_4204	
DSC_4226	
DSCN0789	Skats pa braucošā autobusa logu
DSCN0792	
DSCN0794	
DSCN0826	
DSCN0844	
DSCN0849	
DSCN0866	
DSCN0843	
DSCN4875	Upe
Forest1	Mežs
Liepaja1	Piekraste
Scenery1	Dzelzceļa makets
Train1	Garāmbraucošs vilciens
Train2	
Train3	
Train4	

```
ffmpeg -ss 00:00 -i D:\temp\video_search_res\DSCN0789.MOV -r 1 D:\temp\video_search_res\DSCN0789_frames\%04d.jpeg
```

3.1. att. **ffmpeg komanda kadru virknes ģenerēšanai no video**

Tādā veidā tika iegūtas attēlu virknes priekš visiem video fragmentiem.

3.2.3. Indeksa ģenerēšana

Lai izveidotu indeksu, autors izmantoja Microsoft Computer Vision API[39], modificējot sniegto piemēru indeksa ģenerēšanas vajadzībām. Kods ir pieejams 1. pielikumā. Microsoft piedāvātais serviss apmierinoši tiek galā ar uzdevumu; uzģenerētās atzīmes der šī darba mērķiem. Atzīmes ir bez svariem; spriežot pēc to konsistences un atbilstības padotajiem kadriem, tie ir sakārtoti ticamības dilšanas secībā. Attēlos 3.2. un 3.3. ir parādīts kadrs un tam atbilstošais atzīmju komplekts.



3.2. att. Viens no video kadriem

outdoor
tree
plant
grass
forest
park
large
standing
sitting
bench
field
group
tall
grazing
red
giraffe
man
street
white

3.3. att. Attēlam 3.2. atbilstošais atzīmju komplekts

Katrs atzīmju komplekts tika saglabāts atsevišķā teksta datnē ar tādu pašu nosaukumu, kā atbilstošajam kadram.

3.2.4. Meklēšana

Meklēšana tika realizēta valodā C#. Atzīmju kopas tiek glabātas HashSet[40] tipa mainīgajos. Meklēšanai pēc kārtas tika izmantoti video fragmenti, no kuriem tika ģenerēts indekss. Programma salīdzina padotā video indeksu ar visu pārējo video indeksiem.

Salīdzināšanas tika veiktas ar nesakrītību uzskaitīšanas pieeju (pierakstot tās atzīmes, kas ir padotā video atzīmju kopā). Rezultāti tika izvadīti kā video nosaukumi un nesakrītošu atzīmju skaits. Kods ir pieejams 2. pielikumā. Tabulās 3.2 – 3.20 ir programmas darbības rezultāti.

3.2. tabula

Rezultāti priekš DSC_4199

Video	Starpība
DSC_4199	0
DSC_4226	14
DSCN4875	14
Liepaja1	14
DSC_4204	15
DSCN0826	18
DSCN0844	18
Train4	19
Forest1	20
Scenery1	20
Train1	20
Train2	20
DSCN0789	21
DSCN0849	21
DSCN0792	22
DSCN0794	22
DSCN0866	22
Train3	22
DSCN0843	23

3.3. tabula

Rezultāti priekš DSC_4204

Video	Starpība
DSC_4204	0
Liepaja1	10
DSC_4226	11
DSC_4199	12
DSCN0844	14
DSCN4875	14
DSCN0789	15
DSCN0792	15
Forest1	15
Scenery1	15
Train1	15
DSCN0826	16
Train2	16
DSCN0849	17
Train4	17
DSCN0794	18
DSCN0866	18
DSCN0843	19
Train3	19

3.4. tabula

Rezultāti priekš DSC_4226

Video	Starpība
DSC_4226	0
Liepaja1	22
DSC_4199	26
DSC_4204	26
DSCN4875	26
Forest1	26
DSCN0826	28
Train4	28
DSCN0844	29
Scenery1	29
DSCN0792	30
DSCN0794	30
Train1	30
Train2	30
DSCN0849	31
DSCN0866	31
DSCN0789	32
Train3	32
DSCN0843	34

3.6. tabula

Rezultāti priekš DSCN0789

Video	Starpība
DSCN0789	0
Train4	17
Train1	20
Train3	21
DSCN0792	22
Liepaja1	22
DSCN0826	25
Train2	25
Scenery1	27
DSCN4875	29
DSCN0849	30
DSCN0844	31
DSCN0794	33
Forest1	38
DSCN0866	40
DSC_4204	50
DSCN0843	51
DSC_4226	52
DSC_4199	53

3.5. tabula

Rezultāti priekš DSCN0792

Video	Starpība
DSCN0792	0
Liepaja1	14
DSCN0826	17
Train1	17
Train4	17
DSCN0789	19
DSCN0849	21
DSCN0794	22
DSCN0844	25
Train2	25
Train3	25
DSCN4875	27
DSCN0866	29
Scenery1	29
Forest1	31
DSCN0843	46
DSC_4204	47
DSC_4226	47
DSC_4199	51

3.7. tabula

Rezultāti priekš DSCN0794

Video	Starpība
DSCN0794	0
Liepaja1	13
DSCN0826	19
DSCN0866	20
DSCN0792	23
Train4	24
DSCN0849	25
Train1	26
DSCN4875	29
Forest1	29
Train2	30
Train3	30
DSCN0789	31
DSCN0844	36
Scenery1	37
DSCN0843	46
DSC_4226	48
DSC_4204	51
DSC_4199	52

3.8. tabula

Rezultāti priekš DSCN0826

Video	Starpība
DSCN0826	0
Liepaja1	30
DSCN0849	32
DSCN4875	45
Train1	46
DSCN0792	47
DSCN0794	48
DSCN0844	49
Train2	49
Train4	50
DSCN0789	52
Forest1	53
DSCN0866	57
Scenery1	58
Train3	60
DSCN0843	71
DSC_4226	75
DSC_4199	77
DSC_4204	78

3.10. tabula

Rezultāti priekš DSCN0844

Video	Starpība
DSCN0844	0
Liepaja1	24
DSCN0826	25
DSCN0849	30
DSCN0792	31
Scenery1	31
Train2	32
DSCN0789	34
DSCN4875	36
Train1	36
Train4	38
Train3	39
DSCN0794	41
Forest1	42
DSCN0866	44
DSCN0843	49
DSC_4204	52
DSC_4226	52
DSC_4199	53

3.9. tabula

Rezultāti priekš DSCN0849

Video	Starpība
DSCN0849	0
DSCN0826	5
Liepaja1	15
DSCN0792	24
DSCN4875	26
DSCN0794	27
DSCN0844	27
Train1	27
Forest1	29
Train4	29
DSCN0789	30
Train2	30
DSCN0866	34
Train3	38
Scenery1	40
DSCN0843	46
DSC_4226	51
DSC_4204	52
DSC_4199	53

3.11. tabula

Rezultāti priekš DSCN0866

Video	Starpība
DSCN0866	0
DSCN0794	4
Liepaja1	6
DSCN0826	12
DSCN0792	14
Train1	15
DSCN0849	16
Train4	16
DSCN4875	20
Train2	20
Forest1	21
DSCN0789	22
Train3	22
DSCN0844	23
Scenery1	25
DSC_4226	33
DSCN0843	34
DSC_4204	35
DSC_4199	36

3.12. tabula

Rezultāti priekš DSCN0843

Video	Starpība
DSCN0843	0
Liepaja1	2
DSCN0826	16
DSCN0844	18
DSCN0849	18
DSCN4875	19
DSCN0794	20
Forest1	20
Train2	20
Train4	20
DSCN0792	21
Train3	21
Scenery1	22
Train1	22
DSCN0789	23
DSCN0866	24
DSC_4204	26
DSC_4226	26
DSC_4199	27

3.14. tabula

Rezultāti priekš DSCN4875

Video	Starpība
DSCN4875	0
Liepaja1	19
DSCN0826	23
Forest1	27
Train2	28
DSCN0849	31
Train4	31
Train3	33
DSCN0789	34
DSCN0792	35
DSCN0794	36
Train1	36
DSCN0844	38
Scenery1	40
DSCN0866	43
DSC_4199	51
DSC_4226	51
DSCN0843	52
DSC_4204	54

3.13. tabula

Rezultāti priekš Forest1

Video	Starpība
Forest1	0
DSCN4875	12
Liepaja1	12
DSCN0826	16
DSCN0849	19
DSCN0794	21
Train1	21
Train4	22
DSCN0792	24
Train3	24
Train2	25
DSCN0789	28
Scenery1	28
DSCN0844	29
DSCN0866	29
DSC_4226	36
DSCN0843	38
DSC_4204	40
DSC_4199	42

3.15. tabula

Rezultāti priekš Liepaja1

Video	Starpība
Liepaja1	0
DSCN0826	56
DSCN4875	67
Train2	67
DSCN0794	68
DSCN0849	68
Train4	69
DSCN0792	70
Train1	71
DSCN0844	74
Train3	74
DSCN0789	75
Forest1	75
Scenery1	75
DSCN0866	77
DSCN0843	83
DSC_4226	95
DSC_4204	98
DSC_4199	99

3.16. tabula

Rezultāti priekš Scenery1

Video	Starpība
Scenery1	0
Liepaja1	32
Train2	34
Train4	34
DSCN0789	37
DSCN0844	38
Train3	38
Train1	39
DSCN0826	41
DSCN0792	42
DSCN4875	45
Forest1	48
DSCN0794	49
DSCN0849	50
DSCN0866	53
DSC_4226	59
DSC_4204	60
DSCN0843	60
DSC_4199	62

3.18. tabula

Rezultāti priekš Train1

Video	Starpība
Train1	0
Train4	15
Liepaja1	18
DSCN0826	19
DSCN0789	20
DSCN0792	20
Train3	23
DSCN0849	27
DSCN0794	28
Scenery1	29
Train2	30
DSCN4875	31
Forest1	31
DSCN0844	33
DSCN0866	33
DSC_4204	50
DSC_4226	50
DSCN0843	50
DSC_4199	52

3.17. tabula

Rezultāti priekš Train2

Video	Starpība
Train2	0
Liepaja1	20
DSCN0826	28
DSCN4875	29
Scenery1	30
Train3	30
DSCN0789	31
Train4	32
DSCN0792	34
DSCN0844	35
DSCN0849	36
Train1	36
DSCN0794	38
Forest1	41
DSCN0866	44
DSCN0843	54
DSC_4226	56
DSC_4204	57
DSC_4199	58

3.19. tabula

Rezultāti priekš Train3

Video	Starpība
Train3	0
Train4	16
DSCN0789	24
Liepaja1	24
Train1	26
Train2	27
DSCN0792	31
DSCN4875	31
Scenery1	31
DSCN0794	35
DSCN0826	36
Forest1	37
DSCN0844	39
DSCN0849	41
DSCN0866	43
DSCN0843	52
DSC_4226	55
DSC_4199	57
DSC_4204	57

3.20. tabula

Rezultāti priekš Train4

Video	Starpība
Train4	0
Train3	22
Train1	24
Liepaja1	25
DSCN0789	26
DSCN0792	29
DSCN0826	32
Scenery1	33
DSCN0794	35
DSCN4875	35
Train2	35
DSCN0849	38
Forest1	41
DSCN0866	43
DSCN0844	44
DSC_4226	57
DSCN0843	57
DSC_4199	60
DSC_4204	61

Programmas rezultāti tika sakārtoti pēc atzīmju skaita. Ar iekrāsotām šūnām ir apzīmēti video, kuri ir no tās pašas tematikas grupas vai tai līdzīgas grupas: grupa “Upe” ir līdzīga grupai “Piekaste”; “Dzelzceļa makets” – “Garāmbraucošs vilciens”; “Skats pa braucošā autobusa logu” – “Mežs” – “Piekaste”.

REZULTĀTI

Darba ietvaros tika identificēta problēma, tad tika apskatītas un analizētas video meklēšanas metodes. Balstoties uz identificētajām video meklēšanas pēc satura metožu īpašībām un trūkumiem, tika izvēlēta datorredzes metode, uz kuras bāzes tika izveidots indeksēšanas un meklēšanas pēc satura risinājums ar trīs indeksa detalizācijas pakāpēm un divām dažādām meklēšanā lietojamām salīdzināšanas pieejām.

Darba praktiskajā daļā tika uzģenerēts indekss priekš autora filmētajiem video, kā arī izveidots prototips, lai pārbaudītu definētā risinājuma darbību. Programmas darbības rezultāti parāda, ka algoritms strādā labāk vieniem video, bet citiem sliktāk: DSCN0843 (kadrā ir tikai upe) kā pirmo rezultātu pēc sevis atgriezta Liepaja1 (piekraste; parādās gan jūra, gan mežs); savukārt, video no grupām “Koka ziedi tuvumā”, “Skats pa braucošā autobusa logu”, “Mežs” un “Garāmbraucošs vilciens” deva jauktus rezultātus meklēšanā, taču kopumā sagaidāmie video ir koncentrēti tuvāk pirmajiem rezultātiem.

Visi 19 apstrādājamie video ir HD kvalitātē; kopējais garums ir 7 minūtes un 24 sekundes; paņemto kadru skaits ir 473. Programmas tika darbinātas uz datora ar i5-3570K procesoru (3.8GHz) un atmiņas ātrumu 1600MHz. 50 sekunžu video sadalīšana attēlu virknē aizņēma 6.471 sekundes. Atzīmju saņemšanas priekš video kadriem ātrums tika ierobežots no servisa sniedzēja puses līdz 20 apstrādājamiem attēliem minūtē (maksas opcija atļauj sūtīt 10 attēlus sekundē[33]). Atzīmju apvienošana visiem video aizņēma 2.643 sekundes un visu video meklēšana pēc kārtas video bibliotēkā aizņēma 0.503 sekundes. Ātrdarbības mērījumi tika veikti ar PowerShell[41] komandas “Measure-Command” un Visual Studio Profiler[42] palīdzību.

Video sadalīšanas attēlu virknē ātrums ir atkarīgs no video kvalitātes un garuma: jo lielāka kvalitāte un/vai garums, jo lielāks datu apjoms, ar kuru jāstrādā. Atzīmju apvienošanas visiem video un video meklēšanas ātrums nav atkarīgs no video kvalitātes, jo tiek apstrādātas atzīmes, nevis pats video vai tā kadri, bet ir atkarīgs no uzģenerēto atzīmju skaita – jo vairāk atzīmju, jo lielāks datu apjoms, ar kuru jāstrādā. Uzģenerēto atzīmju skaits ir atkarīgs no video garuma un satura – jo vairāk objektu datorredze atpazīs, jo vairāk atzīmju uzģenerēs.

SECINĀJUMI

Video meklēšanas pēc satura metodes definīcija bija veiksmīga – tā ļāva darba ietvaros izveidot vienkāršu ieviešanā un strādājošu prototipu, ko iespējams uzlabot un uz tā pamata veidot izmantojamu risinājumu.

Programmas darbības rezultāti parāda, ka algoritms strādā labāk, ja padotajam video ir unikāla tematika, par ko liecina meklēšanas rezultāti: kad padotā video kadrā ir tikai upe, kā pirmo rezultātu pēc sevis tika atgriezts video ar piekrasti, kurā parādās gan jūra, gan mežs, savukārt, meklējot otrādi, video ar upi parādās starp pēdējiem rezultātiem, kamēr par līdzīgākiem tika uzskatīti video ar mežu un ainām, kuras redzamas pa autobusa logu. Padodot video no kategorijas “Skats pa braucošā autobusa logu”, starp sagaidāmajiem rezultātiem parādās arī video no kategorijas ” Garāmbraucošs vilciens”, jo, tāpat, kā video no autobusa, video ar vilcieniem arī ir redzama daba. Analogisks efekts novērojams, meklējot otrādi (padodot video no kategorijas “Garāmbraucošs vilciens”). Autors neuzskata, ka tā ir algoritma nepilnība, bet gan informācijas par atzīmju nozīmīgumu neesamības ierobežojums – datorredze nepasaka, kuri objekti ir nozīmīgi, tā tikai pārskaita visu, ko atpazīst. Šis ierobežojums nav aktuāls, ja meklēšana notiek, ierakstot atzīmes, pēc kurām meklēt, jo lietotājs, rakstot atzīmes, pats nosaka, ko tieši viņš vēlas redzēt. Kā arī eksperimentā netika lietoti atzīmju svāri, kuri varētu palīdzēt sasniegt precīzāku meklēšanu. Autora rīcībā esošie video ir tuvi pēc tematikas, kas arī radīja traucējumus video nošķiršanā; no otras puses, reālo video starpā ir daudz līdzīgu.

Lai metode būtu piemērota izmantošanai, algoritmam jāievēro atzīmju svaru informācija. Objektu svarīguma noteikšanas iespēja arī palīdzētu sasniegt labākus rezultātus. Iespējams, izvēlētais datorredzes serviss nav labākais starp pieejamajiem, kas arī ietekmē meklēšanas kvalitāti. Datorredze turpina attīstīties, kas palīdzēs metodei sasniegt labākus rezultātus.

Datorredzes algoritma pārmācīšana uz citiem datiem, vai cita servisa izmantošana, kad indekss jau ir uzģenerēts ar iepriekšējo algoritmu, var radīt traucējumus meklēšanā, ja kādi objekti vairs netiks atpazīti vai ja tie paši objekti tiks apzīmēti ar citām atzīmēm.

Autors guva priekšstatu par eksistējošām video meklēšanas pēc satura metodēm, kā arī datorredzes iespējām un trūkumiem. Turpmāk autors izmantos atzīmju svaru informāciju, lai uzlabotu algoritmu, kā arī daudzveidīgāku un lielāku video kolekciju. Indeksa kompaktnumu iespējams uzlabot, glabājot nevis pašas atzīmes, bet to identifikatorus, kamēr atzīmes glabāsies vārdnīcā ar tiem piešķirtiem identifikatoriem, kas arī tiks izmēģināts.

IZMANTOTĀ LITERATŪRA

1. LZA terminu vārdnīca. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<http://termini.lza.lv>
2. Oxford vārdnīca. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<https://en.oxforddictionaries.com>
3. Orientēto gradientu histogrammas definīcija pēc Wikipedia. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
4. Learned-Miller, E. G. (2011). Introduction to computer vision. University of Massachusetts, Amherst. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
http://www.cs.umass.edu/~elm/Teaching/Docs/IntroCV_9_4_13.pdf
5. Algorithmia. 2018. Introduction to Machine Learning. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://blog.algorithmia.com/introduction-to-machine-learning/>
6. Algorithmia. 2018. Introduction to Computer Vision. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://blog.algorithmia.com/introduction-to-computer-vision/>
7. Microsoft. 2019. Computer Vision API Documentation. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://docs.microsoft.com/lv-lv/azure/cognitive-services/computer-vision/>
8. Clarifai datorredzes serviss. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<https://clarifai.com/>
9. Amazon Rekognition datorredzes serviss. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://aws.amazon.com/ru/rekognition/>
10. Google Vision AI. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<https://cloud.google.com/vision/>
11. Cisco. 2019. Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>
12. Salman Aslam. 2019. YouTube by the Numbers: Stats, Demographics & Fun Facts. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<https://www.omnicoreagency.com/youtube-statistics/>
13. Artyom Dogtiev. 2019. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<http://www.businessofapps.com/data/youtube-statistics/>

14. Google attēlu meklēšana pēc padotā attēla. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://www.google.com/imghp?hl=en>
15. Shazam mūzikas meklēšana pēc ierakstītā fragmenta. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://www.shazam.com/>
16. YouTube. 2019. Lesson: Search and discovery on YouTube. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:
<https://creatoracademy.youtube.com/page/lesson/discovery#strategies-zippy-link-2>
17. Collins vārdnīcas definīcija scēnas jēdzienam. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://www.collinsdictionary.com/dictionary/english/scene>
18. Chang, S. F., Chen, W., Meng, H. J., Sundaram, H., & Zhong, D. (1998). A fully automated content-based video search engine supporting spatiotemporal queries. IEEE transactions on circuits and systems for video technology, 8(5), 602-615. [tiešsaiste][atsauce 28.09.2019]. Pieejams internetā:
<https://pdfs.semanticscholar.org/ad44/ffbad45b2de2147585016e5272a7d745cb57.pdf>
19. Chang, H. S., Sull, S., & Lee, S. U. (1999). Efficient video indexing scheme for content-based retrieval. IEEE Transactions on Circuits and Systems for Video Technology, 9(8), 1269-1279. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<https://ieeexplore.ieee.org/abstract/document/809161>
20. Irani, M., & Anandan, P. (1998). Video indexing based on mosaic representations. Proceedings of the IEEE, 86(5), 905-921. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<http://www.weizmann.ac.il/math/irani/sites/math.irani/files/publications/video-indexing.pdf>
21. Yuan, J., Duan, L. Y., Tian, Q., & Xu, C. (2004, October). Fast and robust short video clip search using an index structure. In Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval (pp. 61-68). ACM. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
http://www.ece.northwestern.edu/~jyu410/index_files/papers/p61-Yuan_Duan_Tian_MIR04.pdf
22. Elizabeth Hansen. 2014. YouTube SEO part 2: How to write titles and descriptions that help you rank. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://www.flint-group.com/blog/youtube-seo-titles-descriptions/>
23. Brendan Cournoyer. 2014. YouTube SEO Best Practices: Titles and Descriptions [eBook Excerpt]. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā:

<https://www.brainshark.com/ideas-blog/2014/March/youtube-seo-best-practices-titles-descriptions>

24. Snoek, C. G., & Worring, M. (2005). Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25(1), 5-35. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.8258&rep=rep1&type=pdf>
25. Snoek, C. G., & Worring, M. (2005). Multimedia event-based video indexing using time intervals. *IEEE Transactions on Multimedia*, 7(4), 638-647. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
https://www.researchgate.net/profile/Marcel_Worring/publication/3424333_Multimedia_event_based_video_indexing_using_time_intervals/links/0912f50893291919eb000000.pdf
26. SCHETTINI, R., CIOCCA, G., ZUFFI, S., & delle Ricerche, C. N. A SURVEY OF METHODS FOR COLOUR IMAGE INDEXING AND RETRIEVAL IN IMAGE DATABASES. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<https://pdfs.semanticscholar.org/c66b/2a6fcd195b0a37031644c705b509fa02ac46.pdf>
27. Pass, G., Zabih, R., & Miller, J. (1996, November). Comparing Images Using Color Coherence Vectors. In *ACM multimedia* (Vol. 96, pp. 65-73). [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9596&rep=rep1&type=pdf>
28. McConnell, Robert K. "Method of and apparatus for pattern recognition." U.S. Patent No. 4,567,610. 28 Jan. 1986. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<https://patents.google.com/patent/US4567610A/en>
29. Satya Mallick. 2016. Histogram of Oriented Gradients. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://www.learnopencv.com/histogram-of-oriented-gradients/>
30. Li, J., Wu, W., Wang, T., & Zhang, Y. (2008, June). One step beyond histograms: Image representation using Markov stationary features. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). IEEE. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.4570&rep=rep1&type=pdf>
31. Deng, Y., & Manjunath, B. S. (2001). Unsupervised segmentation of color-texture regions in images and video. *IEEE transactions on pattern analysis and machine intelligence*, 23(8), 800-810. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā:
<http://192.41.170.42/~mdailey/cvreadings/Deng-ColorTexture.pdf>

32. Lilian Weng. 2017. Object Detection for Dummies Part 1: Gradient Vector, HOG, and SS. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://lilianweng.github.io/blog/2017/10/29/object-recognition-for-dummies-part-1.html>
33. Microsoft. 2019. Cognitive Services Pricing—Computer Vision API. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/computer-vision/>
34. Charles Ollion. 2018. Why computer vision APIs won't do the trick for verticalized applications. Heuritech's take in Fashion. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://blog.serenacapital.com/why-computer-vision-apis-wont-do-the-trick-for-verticalized-applications-93abe59fb021>
35. Clarifai. 2019. Custom Training Overview. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <http://help.clarifai.com/custom-training/guides/custom-training-overview>
36. Wildemuth, B. M., Marchionini, G., Yang, M., Geisler, G., Wilkens, T., Hughes, A., & Gruss, R. (2003, May). How fast is too fast?: evaluating fast forward surrogates for digital video. In Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries (pp. 221-230). IEEE Computer Society. [tiešsaiste][atsauce 26.09.2019]. Pieejams internetā: <https://video.ils.unc.edu/papers/p221-wildemuth.pdf>
37. Cherish Ortiz. 2018. Intro to Video Frame Rates and Frames Per Second Shooting Speeds. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://www.borrowlenses.com/blog/intro-to-video-frame-rates-and-frames-per-second-shooting-speeds/>
38. FFmpeg video un audio ierakstīšanas, konvertēšanas un straumēšanas rīks. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://ffmpeg.org/>
39. Microsoft. 2019. Quickstart: Analyze an image using the Computer Vision SDK and C#. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/quickstarts-sdk/csharp-analyze-sdk>
40. C# HashSet klases apraksts. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.hashset-1?view=netframework-4.8>
41. Microsoft. 2018. PowerShell. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6>
42. Microsoft. 2019. Measure app performance in Visual Studio. [tiešsaiste][atsauce 25.09.2019]. Pieejams internetā: <https://docs.microsoft.com/en-us/visualstudio/profiling/?view=vs-2019>

PIELIKUMI

1. *pielikums* Atzīmju priekš video kadriem iegūšanas programmas pirmkods

```
using Microsoft.Azure.CognitiveServices.Vision.ComputerVision;
using Microsoft.Azure.CognitiveServices.Vision.ComputerVision.Models;

using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;

namespace ImageAnalyze
{
    class Program
    {
        private const string subscriptionKey = "xxxxxxxxxxxxxxxxxxxx";

        private const string path = "D:\\temp\\video_search_res\\DSCN0866_frames\\";

        // Specify the features to return
        private static readonly List<VisualFeatureTypes> features =
            new List<VisualFeatureTypes>()
            {
                VisualFeatureTypes.Categories,
                VisualFeatureTypes.Description,
                VisualFeatureTypes.ImageType,
                VisualFeatureTypes.Tags
            };

        static void Main(string[] args)
        {
            ComputerVisionClient computerVision = new ComputerVisionClient(
                new ApiKeyServiceClientCredentials(subscriptionKey),
                new System.Net.Http.DelegatingHandler[] { });

            // Specify the Azure region
            computerVision.Endpoint =
                "https://westcentralus.api.cognitive.microsoft.com";

            String i4;
            for (int i = 1; i < 17; i++)
            {
                i4 = i.ToString().PadLeft(4, '0');
                var t2 = AnalyzeLocalAsync(computerVision, path + i4 + ".jpeg", i4);

                Task.WhenAll(t2).Wait(10000);
                Task.Delay(4000).Wait();
            }

            // Analyze a local image
            private static async Task AnalyzeLocalAsync(
                ComputerVisionClient computerVision, string imagePath, string i4)
            {
                if (!File.Exists(imagePath))
                {
                    Console.WriteLine(
                        "\nUnable to open or read localImagePath:\n{0} \n", imagePath);
                    return;
                }
            }
        }
    }
}
```

```

        using (Stream imageStream = File.OpenRead(imagePath))
        {
            ImageAnalysis analysis = await
computerVision.AnalyzeImageInStreamAsync(
                imageStream, features);
            DisplayResults(analysis, imagePath, i4);
        }
    }

    private static void DisplayResults(ImageAnalysis analysis, string imageUri,
string i4)
    {
        String tags = "";

        foreach (String tag in analysis.Description.Tags)
            tags += tag+Environment.NewLine;

        File.WriteAllText(path + i4 + ".txt", tags);
    }
}
}
}

```

2. pielikums Video meklēšanas pēc satura programmas pirmkods

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Video_lookup_by_contents
{
    class Program
    {
        //Apvieno visas atzīmes priekš viena video
        static HashSet<String> getTagsForVideo(string videoPath)
        {
            string tag; //tekošā atzīme
            string i4 = "0001"; //iterators četrциparu formātā
            HashSet<String> tagsUnique = new HashSet<String>();

            //Kamēr video ir kadri
            for (int i = 1; File.Exists(videoPath + i4 + ".txt"); i++)
            {
                StreamReader tags = new System.IO.StreamReader(videoPath + i4 +
                ".txt");

                //Nolasa kadra atzīmju kopu
                while ((tag = tags.ReadLine()) != null)
                {
                    tagsUnique.Add(tag);
                }

                tags.Close();

                i4 = (i+1).ToString().PadLeft(4, '0');
            }

            return tagsUnique;
        }

        static void Main(string[] args)
        {
            string[] videos = { "DSC_4199",
                                "DSC_4204",
                                "DSC_4226",
                                "DSCN0789",
                                "DSCN0792",
                                "DSCN0794",
                                "DSCN0826",
                                "DSCN0844",
                                "DSCN0849",
                                "DSCN0866",
                                "DSCN0843",
                                "DSCN4875",
                                "Forest1",
                                "Liepaja1",
                                "Scenery1",
                                "Train1",
                                "Train2",
                                "Train3",
                                "Train4"
                                }; //visi video
        }
    }
}
```

```

int[] matches = new int[19];
String inputVideo, results = "";

for (int j = 0; j < 19; j++)
{
    inputVideo = videos[j];           //paņem katru video pēc kārtas
    results += inputVideo + Environment.NewLine;

    HashSet <String> tagsSearched =
getTagsForVideo("D:\\temp\\video_search_res\\" + inputVideo + "_frames\\"),
tagsSearchedRemainder;

    //Salīdzina ar visiem video
    for (int i = 0; i < 19; i++)
    {
        HashSet<String> tags =
getTagsForVideo("D:\\temp\\video_search_res\\" + videos[i] + "_frames\\");
        tagsSearchedRemainder = new HashSet<String>(tagsSearched);
        tagsSearchedRemainder.ExceptWith(tags);           //atzīmju kopu

        matches[i] = tagsSearchedRemainder.Count;
    }

    for (int i = 0; i < 19; i++)
        results += videos[i] + ";" + matches[i] + Environment.NewLine;
}
File.WriteAllText("L:\\results_except.csv", results);
}
}
}

```

Bakalaura darbs „Video failu satura apstrāde meklēšanas vajadzībām” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Artjoms Latiševičš _____

Rekomendēju/nerekomendēju darbu aizstāvēšanai (nederīgo svītro vadītājs)

Vadītājs: pasniedzējs Mg. dat. Krišjānis Nesenbergs _____ __.05.2019.

Recenzents: Mg.sc.comp. Ilvars Mizniks

Darbs iesniegts Datorikas fakultātē __.05.2019.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

____.06.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____