VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ANONYMIZACE VIDEA

VIDEO ANONYMIZATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. MARTIN MOKRÝ**
AUTHOR

VEDOUCÍ PRÁCE **prof. Ing. ADAM HEROUT, Ph.D.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Mokrý Martin, Bc.**
Program: Informační technologie Obor: Počítačová grafika a multimédia
Název: **Anonymizace videa**
Video Anonymization
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s problematikou zpracování obrazu a videa.
2. Vyhledejte a popište existující nástroje a postupy pro anonymizaci obrazu a videa, charakterizujte jejich vlastnosti a výhody/nevýhody.
3. Pořiďte a podle potřeby anotujte datovou sadu pro hodnocení algoritmů anonymizace obrazu/videa.
4. Vyhledejte dostupné detektory objektů v obraze/videu a prostřednictvím vhodných experimentů vyhodnoťte jejich použitelnost pro anonymizaci videa.
5. Navrhněte a vytvořte systém pro anonymizaci videa.
6. Vyhodnoťte vytvořený systém na datové sadě a iterativně vylepšujte jak vytvořené řešení, tak datovou sadu.
7. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentaci výsledků projektu.

Literatura:

- Gary Bradski, Adrian Kaehler: Learning OpenCV; Computer Vision with the OpenCV Library, O'Reilly Media, 2008
- Richard Szeliski: Computer Vision: Algorithms and Applications, Springer, 2011

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3, značné rozpracování bodů 4 a 5.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 1. listopadu 2018

Abstrakt

Cieľom tejto práce je navrhnutie a vytvorenie automatického systému pre anonymizáciu videí. Tento systém na svoju činnosť využíva rôzne detektory objektov v obraze a taktiež aj aktívne sledovanie takto zdetekovaných objektov. Na zdetekované objekty je následne aplikovaná úprava, ktorá zabezpečí dostatočnú mieru anonymizácie. Hlavným prínosom takéhoto systému je urýchlenie anonymizácie videí, ktoré bude následne možné zverejniť.

Abstract

The goal of this thesis is to design and create an automatic system for video anonymization. This system makes use of various object detectors on an image to ensure functionality, as well as active tracking of objects detected in this manner. Adjustments are later applied to these detected objects which ensure sufficient level of anonymization. The main asset of this system is speeding up the anonymization process of videos that can be published after.

Kľúčové slová

anonymizácia, spracovanie videa, spracovanie obrazu, detekcia objektov, sledovanie objektov, neurónové siete

Keywords

anonymization, video processing, image processing, object detection, object tracking, neural networks

Citácia

MOKRÝ, Martin. *Anonymizace videa*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Anonymizace videa

Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pána profesora Adama Herouta. Uviedol som všetky literárne pramene, z ktorých som čerpal.

.....
Martin Mokrý
21. mája 2019

Poďakovanie

Rád by som poďakoval pánovi Prof. Ing. Adamovi Heroutovi, Ph.D., ktorý mi poskytoval cenné rady a dodal mi motiváciu k riešeniu tejto práce. Zároveň by som rád poďakoval mojej priateľke a rodine za to, že ma v práci po celý čas podporovali.

Obsah

Kapitola 1

Úvod

Táto práca sa zaoberá návrhom a vytvorením automatického anonymizačného systému, ktorý by dokázal anonymizovať všetky potrebné časti videa. V tejto práci sú rozobraté jednotlivé aspekty, ktoré je potrebné poznať, aby bolo možné takýto systém navrhnúť a vytvoriť.

Keďže som sa často stretával s nepochopením zmyslu takéhoto systému, najmä zo strany rodiny a priateľov, rozhodol som sa uviesť ako 2. kapitolu motiváciu, aby som jasne vymedzil dôvody a situácie, kedy môže byť takýto systém užitočný.

V ďalších častiach tejto práce, v kapitolách 3 a 4 sa zameriavam najmä na popis teoretických znalostí, ktoré sú potrebné pre návrh a vytvorenie takéhoto systému. Taktiež sú vhodné aj pre čitateľa tejto práce, aby chápal všetky súvislosti, techniky a technológie, ktoré sa skrývajú za týmto systémom. Tieto kapitoly sa venujú najmä teórii ohľadom anonymizácie - aké druhy anonymizácie existujú, na aké typy dát sa dajú aplikovať a ako sa aplikujú. Keďže vytváraný systém má byť automatický, sú potrebné aj detektory objektov a trackery objektov, ktoré sú v tejto časti taktiež popísané. Sú uvedené základné typy a základné princípy, na ktorých sú založené. Rovnako sú v niektorých prípadoch, keď to bolo vhodné, uvedené aj ich výhody alebo nevýhody.

Kapitola 5 a zameriava už na opis návrhu a implementácie samotného automatického systému pre anonymizáciu. Obsahuje podrobné popisy postupu pri návrhu a postupnej implementácie systému. Taktiež obsahuje vyhodnotenia testov a meraní nástrojov, ktoré boli použité alebo vytvorené pre tento systém.

V poslednej kapitole 6 je uvedené vyhodnotenie vytvoreného systému na základe užívateľských testov a taktiež sú tu navrhnuté možnosti pokračovania v projekte.

Kapitola 2

Motivácia

Anonymizácia videa môže byť veľmi užitočná v mnohých situáciách, kedy je potrebné zverejniť určité video, no už nie je žiadúce, aby toto video obsahovalo niektoré citlivé osobné informácie. Z tohto dôvodu je dobré mať vhodný program/software na anonymizáciu vloženého videa. V dnešnej dobe existuje veľa riešení ako anonymizovať video [kapitola 3.3]. Avšak napriek dnešným pokročilým detekčným technológiám väčšina existujúcich nástrojov používa stále manuálny prístup k anonymizácii, a to aj v prípade bežných objektov, ktoré je možné v obraze detegovať. Taktiež v prípade, ak daný nástroj poskytuje, či už manuálnu alebo automatickú možnosť anonymizácie videa, tak často je táto technika neestetická, a teda rušivá pre diváka.

V tejto kapitole budú rozobrané prípady toho, kedy je potrebné anonymizovať určité časti videa, tak aby video mohlo byť ďalej šírené, a to aj bez nežiadúcich následkov. Rovnako, aby video po aplikovaní anonymizácie bolo esteticky stále prijateľné, najlepšie je, aby divák anonymizáciu ani nepostrehol.

2.1 Identita osôb

Pravdepodobne najčastejší dôvod, prečo je potrebné anonymizovať video, je sťaženie identifikácie osôb vo videu, teda zakryť ich identitu. Dôvodov na odstránenie identity osoby z videa môže byť samozrejme viac a nebudú tu rozoberané všetky, spomenuté budú iba niektoré. Napríklad osoby, ktoré sa nachádzajú vo videu nedali súhlas na zverejnenie a teda sa vo videu nechcú objaviť, prípadne autor nemá súhlas žiadnej osoby z videa a teda je potrebné anonymizovať všetky osoby z videa [*Zákon č. 89/2012 Sb. ČR § 84*]. Identita osoby je v tomto prípade chápaná iba z pohľadu výzoru, vzhľadu, teda fyzickými dispozíciami. Identitou teda nie je myslená osobnosť človeka, pohľad na život a iné filozofické a psychologické vysvetlenia identity. Keďže sa táto práca zaoberá iba anonymizáciou videa, tak ako možnosť identifikácie nebude braná do úvahy ani analýza hlasu danej osoby. Identifikovať osobu na videu, prípadne obraze je možné na základe viacerých faktorov. Avšak pravdepodobne najspoľahlivejší spôsob ako identifikovať osobu je práve na základe jej tváre. Taktiež je možné identifikovať nám známu osobu aj na základe oblečenia, postoju tela, postavy a v prípade videa aj reči tela.

2.2 Identifikácia áut

Podľa poznávacej značky auta je možné identifikovať aj osobu, ktorá toto auto vlastní, teda odhaliť identitu tejto osoby. Zároveň v niektorých štátoch je zakázané zverejňovať videá, ktoré obsahujú aj poznávacie značky áut, prípadne iné osobné údaje [*Zákon č. 101/2000 Sb. ČR § 13*]. Tento fakt následne znemožňuje napríklad zverejniť video z palubnej kamery auta, či už za účelom zábavy na sociálnych sieťach alebo za účelom odhalenia páchateľa nehody. Autá a ich majiteľov je samozrejme možné identifikovať aj na základe iných poznávacích znakov, ako napríklad čísla VIN, čísla blokov motorov a karosérií, no tieto prvky nie sú vo videách tak veľmi zastúpené, a ani viditeľné ako poznávacie značky.

Kapitola 3

Anonymizácia

Anonymita vychádza z gréckeho slova „anonymia“ a znamená bezmenný, neidentifikovateľný, teda z určitej informácie (dát) odstránime tie prvky, ktoré vyjadrujú určitú vlastnosť objektu na základe, ktorej je daný objekt identifikovateľný, popísateľný. Existuje viacero techník [21, 17] ako toto dosiahnuť. Rovnako existuje aj viacero typov informácií alebo typov dát, ktoré sa dajú takto upraviť. Techniky, ktoré sa najčastejšie používajú sú:

- **Vynechanie informácie** – informácia, ktorá popisuje objekt je úplne vynechaná, a teda nie je možné zistiť žiadne bližšie vlastnosti objektu. Táto technika sa používa, ak je informácia nepotrebná pre ďalšiu prácu s upravenými dátami, prípadne je informácia prebytočná. V opačnom prípade môže použitie tejto techniky zapríčiniť nepoužiteľnosť takto upravených dát.

  Príklad: Určitá malá organizácia pozostáva z viacerých členov. O každom členovi sú známe informácie ako meno, dátum narodenia, adresa a telefónne číslo. V prípade, že by chcela organizácia kontrolovať účasť na svojich zasadnutiach pomocou prezenčnej listiny, tak by bolo postačujúce, ak by táto listina obsahovala iba mená členov a ostatné informácie by mohli byť vynechané za účelom zachovania osobných informácií.

  Príklad v obraze: Odstránenie jedného/viacerých farebných kanálov. Na obraze je stále možné identifikovať o aký objekt sa jedná, ale už nie jeho pôvodnú farbu.

- **Maskovanie** – informácia, prípadne iba časť informácie, sa prepíše tak, aby sa znehodnotila. Táto technika sa najčastejšie používa pri textových a číselných typoch dát, kedy je možné ich určitú časť zamaskovať a zvýšiť tak úroveň anonymizácie, no zároveň časť informácie zostane zachovaná.

  Príklad: Zamaskovanie poštového smerového čísla 616 00 (Brno Žabovřesky) na 61x xx. Teda nie je možné určiť o akú mestskú časť sa jedná, ale zostala zachovaná informácia, že sa jedná o Brno.

- **Pseudoanonymizácia** – táto technika sa tiež nazýva kódovanie alebo nahradzovanie. Môže byť vratná alebo nevratná. To záleží od toho, či sa dáta alebo ich časti nahradzujú za náhodné alebo vopred určené znaky/slová. Spočíva v nahradení informácie za inú informáciu, ktorá neskôr môže alebo nemusí byť dekódovaná. Používa sa najčastejšie pre uloženie citlivých informácií do databáz, prípadne iných pamäťových médií. Takéto kódovanie informácie môže byť tiež aplikované aj viac krát po sebe, čo sa nazýva dvojité prípadne viacnásobné kódovanie.

Príklad: Mená členov organizácie z príkladu 1, sa zakódujú do čísel. Teda iba organizácia bude vedieť, o ktorého člena sa jedná.

- **Generalizácia** – zhrnutie konkrétnej informácie alebo časti informácie do väčšieho všeobecného celku. Takto anonymizované dáta dokážu stále poskytovať určitú informáciu, avšak nie natoľko konkrétnu, aby mohla byť zneužitá.

  Príklad: Vek členov organizácie sa prevedie na intervaly. Teda ak má nejaký člen 25 rokov, bude pri jeho mene uvedená napríklad hodnota <20 – 30>.

- **Zamiešanie** – Prehodenie jednotlivých častí dát na iné miesta. Informácie sa tak v dátach stále nachádzajú, ale v inom poradí, čo má za následok zmenu vzťahov medzi jednotlivými časťami, a teda aj zmenu významu celku. Toto sa používa najmä v prípade, že je potrebné zachovať všetky pôvodné informácie nezmenené, napríklad pre potrebu agregácie. Túto techniku je vhodné použiť pri väčšom množstve dát, keďže pri použití na malom množstve dát môže byť vratná. Taktiež v niektorých prípadoch je vhodné zvoliť skupinové zamiešanie, kedy jednotlivé spolu súvisiace položky zostanú v rovnakom vzťahu aj po zamiešaní, aby tak navodili dojem, že zamiešané dáta sú stále relevantné.

  Príklad: prehodenie dátumov narodenia členov organizácie medzi sebou. To znamená, že pri každom členovi bude uvedený nesprávny dátum jeho narodenia a nebude sa teda dať zneužiť, no zároveň súčet rokov všetkých členov zostane nezmenený.

  Príklad skupinového zamiešania: Adresy jednotlivých členov zostanú rovnaké, teda Brno bude stále v Českej republike a so správnym poštovým smerovacím číslom, iba člen organizácie, ktorý bol pôvodne z Brna, bude mať inú adresu.

  Príklad v obraze: Prehodenie pixelov časti obrazu. Časť obrazu stratí svoj pôvodný význam, avšak veľkosť snímky, jej histogram, aj počet kanálov zostane rovnaký.

- **Variácia hodnoty** – Zmena určitej číselnej hodnoty podľa spodného a horného ohraničenia intervalu. Na začiatku sa zvolí spodná a horná hranica intervalu. Ak sa anonymizovaná hodnota nachádza mimo intervalu, tak sa zmení na jednu z hraničných hodnôt intervalu. V prípade, že je hodnota v intervale, tak sa aplikuje vopred dané pravidlo, napríklad pripočítanie alebo odpočítanie určitého čísla od pôvodnej hodnoty. Aby sa pre hodnoty, ktoré sa nachádzajú vo vnútri intervalu neaplikovalo stále rovnaké pravidlo, tak aj toto môže byť obmieňané, napríklad pomocou náhodne generovaného čísla (0 – odčítanie, 1 – pričítanie). Podobne sa dá postupovať aj pri zmene dátumov.

  Príklad: Určenie intervalu veku pre členov organizácie, napríklad na <20-40>, s pravidlom +5: potom každý člen mladší ako 20 rokov bude v systéme vedený ako 20 ročný, každý člen starší ako 40 rokov bude v systéme vedený ako 40 ročný a členom medzi 20 a 40 bude k ich veku pripočítaná hodnota 5 *(maximálna hodnota ale bude stále iba 40 rokov, aj keby mal člen 39 rokov)*.

- **Kryptografické techniky** – používajú sa na tie typy dát, pri ktorých je potrebné zachovať ich integritu, no zároveň nemusia byť na prvý pohľad realistické. Rozdeľujú sa na tri hlavné skupiny: techniky symetrických kľúčov, techniky verejného kľúča a message digest algoritmy[1]. Zvyčajne sa používajú na šifrovanie komunikácie pri rôznych komunikačných protokoloch.

[1]RFC 1321 – https://tools.ietf.org/html/rfc1321

3.1 Typy dát

Samozrejme nie všetky techniky spomenuté v predchádzajúcej sekcii je možné aplikovať na ľubovoľné typy dát. Existuje veľké množstvo rôznych typov dát a ich reprezentácií. Záleží najmä od uhľa pohľadu, či sú typy dát chápané ako dátové typy používaný pri programovaní[2], štatistické typy dát[3] alebo multimediálne [13], ako je to aj v tomto prípade. V tejto sekcii budú spomenuté bežné multimediálne typy a možnosti ako sa dajú anonymizovať. To z toho dôvodu, aby bolo jasne vidieť postavenie videa a anonymizácie videa medzi ostatnými typmi multimédií a aj to, ako sa od nich odlišuje a prípadne, aké techniky sa bežne používajú.

3.1.1 Text

Textom je v tomto prípade možné chápať akékoľvek dokumenty a súbory, ktorých obsah zahŕňa informácie obsiahnuté v alfanumerických znakoch. Na takéto typy súborov je možné uplatniť všetky možnosti anonymizácie z predchádzajúcej sekcie presne takým spôsobom, akým je to pri nich uvedené.

3.1.2 Zvuk

Pravdepodobne najčastejším dôvodom, prečo je potrebné anonymizovať zvuk, je identifikácia osoby na základe hlasu. Jedná sa teda o potrebu znehodnotiť ľudský hlas na nahrávke takým spôsobom, aby sa stratila identita rečníka, no v ideálnom prípade, aby zostala zachovaná informácia z toho, čo rečník na nahrávke hovoril. Toto môže byť potrebné z mnohých dôvodov. Uvediem iba pár, pri ktorých môže byť hlas osoby nejakým spôsobom zneužitý: autentifikácia hlasom, zneužitie osobných informácii, prístup k účtom/budovám alebo v horšom prípade aj vydieranie rečníka.

Anonymizáciou môže byť v tomto prípade chápané aj vynechanie určitej informácie z nahrávky, napríklad ak sa jedná o nejakú citlivú informáciu, prípadne osobné informácie o nejakej osobe. Z toho teda vyplýva, že je možné zvoliť dva základné postupy ako anonymizovať zvuk. Znehodnotiť určitú časť nahrávky alebo upraviť určitú časť nahrávky.

Ak by išlo iba o prípad znehodnotenia časti nahrávky, tak toto je možné urobiť veľmi jednoducho, a to buď vymazaním danej časti alebo jej prepísaním inou informáciou.

V prípade, že je potrebné upraviť nahrávku alebo jej určitú časť, napríklad pre účel straty identity rečníka, je aj toto možné urobiť viacerými spôsobmi. Pravdepodobne najjednoduchším spôsobom je úprava frekvencie nahrávky [6], či už zvýšením alebo znížením základného tónu dôjde k zmene farby hlasu, a teda je horšie identifikovateľný. Táto zmena ale môže byť v niektorých prípadoch vratná. Za viac sofistikovanú metódu sa dá považovať prepis nahrávky na text a následné prečítanie textu niektorým z mnohých nástrojov na čítanie textu[4].

Podobne sofistikované môže byť aj použitie metódy pre spracovanie reči [4]. Jednoducho povedané ide o rozdelenie hlasu na model hlasového ústrojenstva a budiaceho tónu (zvuk z hlasiviek), pričom pre hlasové ústrojenstvo je možné odhadnúť autokorelačné koeficienty pre jednotlivé rámce (20 – 25ms úseky). Tie následne použiť pre filter, do ktorého pustíme tón, ktorým nahradíme budiaci tón hlasiviek. Potom zmenami, či už koeficientov filtra alebo tónu môžeme upravovať výsledný hlas.

[2] Integer, double, string, slovník, ukazatele atď

[3] https://towardsdatascience.com/data-types-in-statistics-347e152e8bee

[4] https://cloud.google.com/speech-to-text/, https://cloud.google.com/text-to-speech/

3.1.3 Obraz

Rovnako ako pri zvuku je aj pri obraze potrebné odstrániť určité informácie, ktoré môžu byť citlivého charakteru a môžu prezrádzať detaily o konkrétnej osobe alebo veci. Táto práca sa zameriava najmä na anonymizáciu spojenú s identitou osôb, a teda sa jej budem v tejto sekcii venovať. V prípade obrazu je možné použiť viacero techník, ktorými je možné obraz upraviť tak, aby osoby alebo predmety s nimi blízko súvisiace nebolo možné rozpoznať, prípadne sa ich rozpoznanie výrazne skomplikovalo.

Podobne ako pri zvuku je možné určitú časť obrazu (RoI[5]), buď upraviť alebo vynechať/znehodnotiť. Pod vynechaním časti obrazu je v tomto prípade možné rozumieť vymazanie pôvodnej informácie a jej nahradenie jednotnou farbou alebo náhodnými hodnotami. Týmto spôsobom sa úplne stratí pôvodná informácia a nie je ju možné nijako obnoviť. Avšak táto technika môže pôsobiť veľmi rušivým dojmom.

Práve z dôvodu rušivého dojmu je občas žiadúce zvoliť úpravu časti obrazu. Táto úprava ale musí byť dostatočná, aby jej aplikáciou prišlo k dostatočnému odstráneniu informácie o identite osoby. To znamená, že v prípade aplikovania iba jemného rozmazania na časť obrazu by mohlo byť stále rozpoznateľné, že osoba je mužského pohlavia, prípadne jej farba pleti je biela a podobné črty. Úpravy, ktoré je možné v tomto prípade použiť sú napríklad rozpixelovanie časti obrazu, Gaussovo rozostrenie, ponechanie iba hrán objektu, odstránenie farby alebo zmena farebných hodnôt. Samozrejme tieto úpravy je možné rôzne kombinovať.

Obr. 3.1: Samuel L. Jackson – pôvodný

Obr. 3.2: Cenzúra cez oči – nahradenie

Obr. 3.3: Cenzúra cez oči – úprava

Obr. 3.4: Cenzúra tváre – blur, farba

Obr. 3.5: Cenzúra tváre – hrany, blur, farba

Obr. 3.6: Cenzúra tváre – nahradenie šumom

[5]Region of Interest

Ako je možné vidieť na obrázkoch 3.1 až 3.6, tak jednotlivé techniky anonymizácie môžu byť použité na rôzne časti obrazu. Niektoré sú viac a iné menej rušivé. Zároveň pri tých menej rušivých, ako obrázku 3.3, nemusí byť miera anonymizácie natoľko dostatočná, aby odstránila črty danej osoby. Z tváre objektu je teda aj po aplikovaní úpravy stále možné určiť farbu pleti, črty tváre (okrem oblasti očí), mimiku tváre alebo pohlavie.

Z toho dôvodu je nutné pri anonymizácií obrazu zvoliť vhodný kompromis medzi mierou anonymizácie a estetickou stránkou po aplikovaní úpravy. Tento pomer musí byť zvolený vhodne tak, aby sa požadovaná informácia dostatočne zamaskovala, no zároveň, aby výsledok príliš nenarúšal estetickú stránku obrazu.

3.1.4 Ostatné

Podobne ako multimediálne dátové typy je možné anonymizovať aj iné typy dát alebo informácií, ktoré môžu poskytovať citlivé informácie. Môže sa jednať o internetovú komunikáciu, a to či už na lokálnej úrovni – používanie anonymného módu prehliadača alebo aj vzdialene použitím šifrovanej komunikácie[6] či VPN služieb[7].

3.2 Možnosti anonymizácie videa

Video kombinuje viacero obrazov do sekvencii idúcich za sebou, čím vzniká pohyblivý obraz, preto je pri videu možné použiť rovnaké techniky na anonymizáciu ako aj pri obraze[8]. Obvyklá obnovovacia frekvencia videa je 25 snímok za sekundu [2], to preto, aby sa video zdalo divákom plynulé. Práve z tohto dôvodu je teda potrebné aplikovať úpravy častí obrazu pre každú jednu snímku, čo je výpočtovo náročnejšie ako iba pre jeden obrázok. Taktiež nastáva problém, že objekty ktoré je potrebné vo videu upravovať môžu postupom času a zmenou snímok meniť v obraze svoju polohu. Nie je teda možné použiť rozmazanie alebo inú potrebnú úpravu na stále tých istých súradniciach obrazu.

Riešením môže byť aplikovanie úpravy po jednotlivých snímkach ručne a vždy na zvolené miesto. Toto ale nie je veľmi vhodné pri dlhých videách, ktoré môžu obsahovať až desať tisíce snímok a na každom z nich môže byť potrebné upraviť niekoľko objektov, preto je potrebné zvoliť iný prístup, ideálne nejaký automatizovaný.

Existuje hneď niekoľko prístupov, ktoré je možné použiť. V každom prípade je ako prvý krok potrebné zvoliť požadovanú časť obrazu, ktorú chceme upravovať. Táto voľba môže byť urobená buď manuálne užívateľom (kliknutím, zvolením súradníc) alebo automatizovane akýmsi detektorom objektov v obraze [kapitola 4].

Keďže video sa skladá z po sebe idúcich snímok, tak za predpokladu, že tieto snímky na seba nadväzujú, sú z veľkej časti aj podobné. Na obrázku 3.8 je možné vidieť znázornenie rozdielu medzi dvomi snímkami, ktoré išli po sebe vo videu. Túto podobnosť následne využívajú nástroje nazývané trackery [20], ktoré dokážu vybraný objekt sledovať medzi jednotlivými snímkami, a tak vždy presne určiť jeho novú pozíciu. K takémuto sledovaniu využívajú rôzne prístupy, záleží o aký tracker sa jedná[9].

Keď poznáme pozíciu objektu v obraze, tak na túto je následne možné aplikovať úpravy potrebné pre anonymizáciu. Toto je potrebné vykonať pre každú jednu snímku vo videu, na ktorej sa nachádza anonymizovaný objekt.

[6] HTTPS, Tor, šifrované správy pomocou WhatsApp a podobne

[7] http://ptgmedia.pearsoncmg.com/images/1587051796/samplechapter/1587051796content.pdf

[8] Podkapitola 3.1.3

[9] Point tracking, kernel tracking, silhouette tracking

Keďže snímky z jedného záznamu sú veľmi podobné, tak toto je možné využiť aj na získanie súradníc objektov v obraze na základe ich pohybu, teda zmeny polohy medzi jednotlivými snímkami. Na tomto princípe fungujú detektory pohybu [3]. Tie je napríklad možné využiť pri statických scénach, kedy jediné objekty, ktoré sa pohybujú sú napríklad práve ľudia, a tým zistiť ich pozíciu v obraze.

Existujú aj iné možnosti ako anonymizovať video a objekty, ktoré sa v ňom nachádzajú. Viac o týchto technikách bude uvedené v nasledujúcej podkapitole 3.3, ale všetky tieto techniky využívajú základný princíp detekcie objektu v obraze (automatickej/manuálnej), následné sledovanie objektu (automaticky/manuálne) a ako konečný krok úpravu danej časti obrazu.

Obr. 3.7: Dve po sebe nasledujúce snímky z videa

Obr. 3.8: Rozdiel *(červená farba)* medzi ľavou a pravou snímkou z obrázku 3.7

3.3 Existujúce riešenia pre anonymizáciu videa

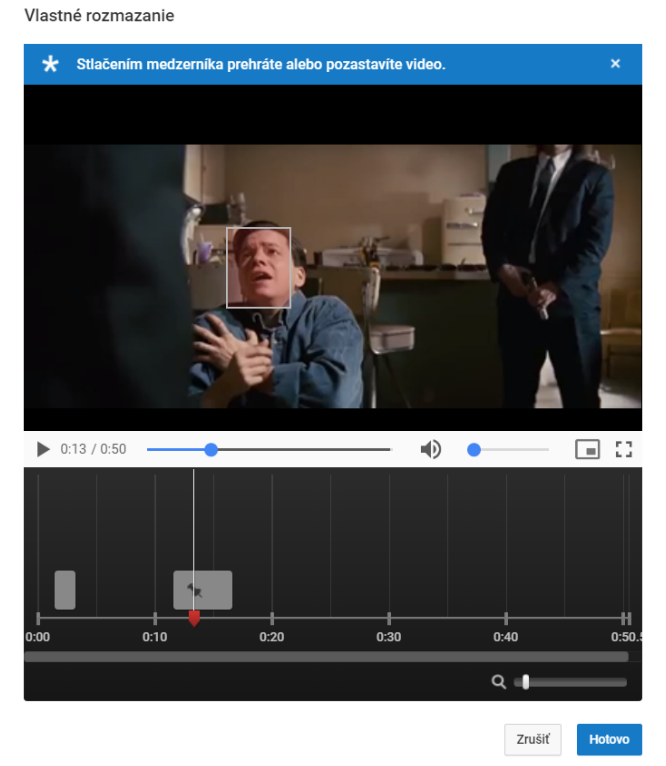
V tejto podkapitole rozoberiem, aké existujú riešenia a nástroje pre anonymizáciu videa. Nástroje, ktoré tu budú uvedené predstavujú zo zástupcov rôznych prístupov k problému anonymizácie, a teda sa určite nejedná o všetky existujúce nástroje pre tento typ úprav videí. Každý z nástrojov je nejakým aspektom odlišný a má svoje výhody ako aj nevýhody. S niektorými nástrojmi mám aj osobné skúsenosti a používal som ich v minulosti pre úpravu videí, prípadne priamo aj na anonymizáciu určitej časti videa.

3.3.1 YouTube

Samotná služba YouTube[10] vo svojom webovom rozhraní ponúka možnosť ako anonymizovať video. Užívateľ má na výber hneď dve varianty anonymizácie. Prvou je automatické rozmazanie tvárí, ktoré je dostupné po spracovaní videa, kedy služba YouTube deteguje vo videu všetky tváre. Následne si užívateľ z príslušného výberu tvárí môže vybrať tú, ktorú chce vo videu rozmazať. Toto je možné vidieť na obrázku 3.10.

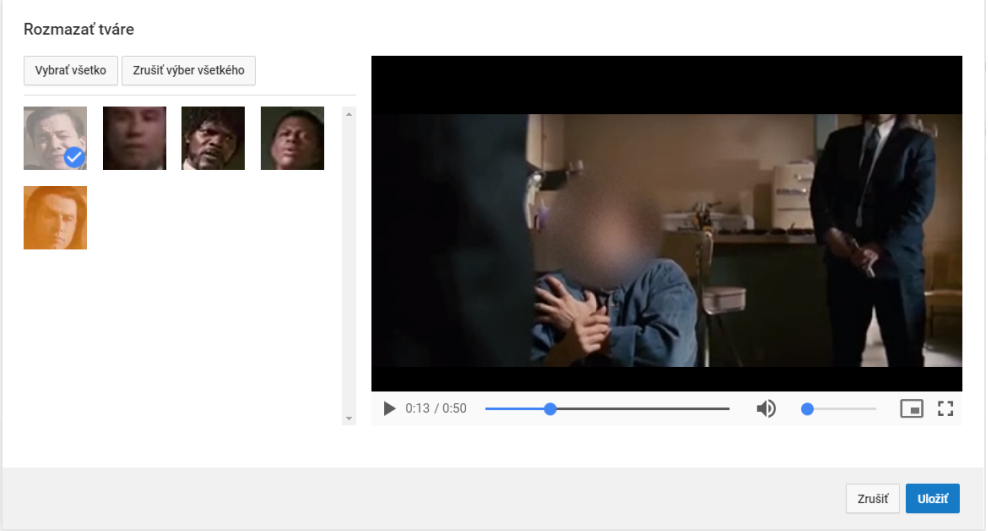
Druhou možnosťou je manuálny výber úseku videa, ktorý ma byť rozmazaný a kedy. K tomuto YouTube poskytuje jednoduché rozhranie, ktoré je možné vidieť na obrázku 3.9. Užívateľ si teda môže jasne definovať, ktorá časť videa má byť rozmazaná a v aký čas. Označená oblasť sa následne snaží sledovať pohyb objektu vo videu, aby užívateľ nemusel zmeny miesta robiť manuálne, avšak aj po zmene scény tam naďalej oblasť zostáva a je teda nutné ju manuálne od istého času vypnúť.

Oba spôsoby je možné aj kombinovať, to v prípade, ak automatické rozmazanie tvárí vynechá nejaký úsek a je potrebné ho opraviť. Veľkou výhodou tohto nástroja je jeho jednoduchosť a rýchlosť, akou sa dá použiť, prípadne upraviť. Naopak miernou nevýhodou by mohla byť nemožnosť zvoliť tvar a mieru rozmazania. Pri manuálnom výbere úseku je možnosť zvoliť iba obdĺžnikový tvar výberu a pri automatickom rozpoznaní tvárí sa implicitne použije kruhové rozmazanie, ktoré nie je možné neskôr meniť.



Obr. 3.9: Nástroj od YouTube na rozmazanie častí obrazu vo videu

[10] https://www.youtube.com/



Obr. 3.10: Nástroj od YouTube na automatické rozmazanie vybraných tvárí

3.3.2 Adobe Premiere Pro CC 2017

Tento nástroj na spracovanie videí síce neposkytuje automatickú detekciu tvárí tak ako YouTube, ale zato ponúka o veľa lepšie možnosti úpravy časti obrazu videa. Pre anonymizáciu sa hodí efekt s názvom Fast Blur, ktorým je možné rozmazať akúkoľvek časť videa. Funguje na veľmi jednoduchom princípe, kedy užívateľ zvolí objekt (oblasť) vo videu, ktorý chce rozmazať. Následne spustí vytváranie cesty objektu vo videu, teda spustí tracker, ktorý aktívne sleduje užívateľom zvolený objekt. Tým užívateľovi odpadáva úloha manuálne meniť pozíciu zvýraznenej časti obrazu. V momente, kedy sa objekt stratí z obrazu, tak sa aj ukončí jeho sledovanie. Potom je už jednoducho možné na vybranú oblasť aplikovať ľubovoľný filter zo širokého výberu možností, ktoré Adobe Premiere Pro ponúka. Taktiež pri každom efekte je aj možné meniť jeho intenzitu, a teda tým ovplyvniť výslednú estetickú stránku obrazu. Rovnako tvar oblasti, ktorú užívateľ vyberá, aby bola sledovaná, je možné meniť na užívateľom požadovaný. Je možné použiť rôzne mnohouholníky, elipsy, prípadne si naznačiť vlastný preferovaný tvar oblasti.

Na druhej strane nevýhoda tohto nástroja spočíva v nemožnosti automaticky detegovať určité vopred určené objekty a na tie použiť zvolený filter. Rovnako by za nevýhodu mohol byť považovaný aj fakt, že licencia tohto nástroja stojí mesačne 24,19€[11] , pričom nástroj od služby YouTube je zdarma.

3.3.3 Adobe After Effects CC 2017

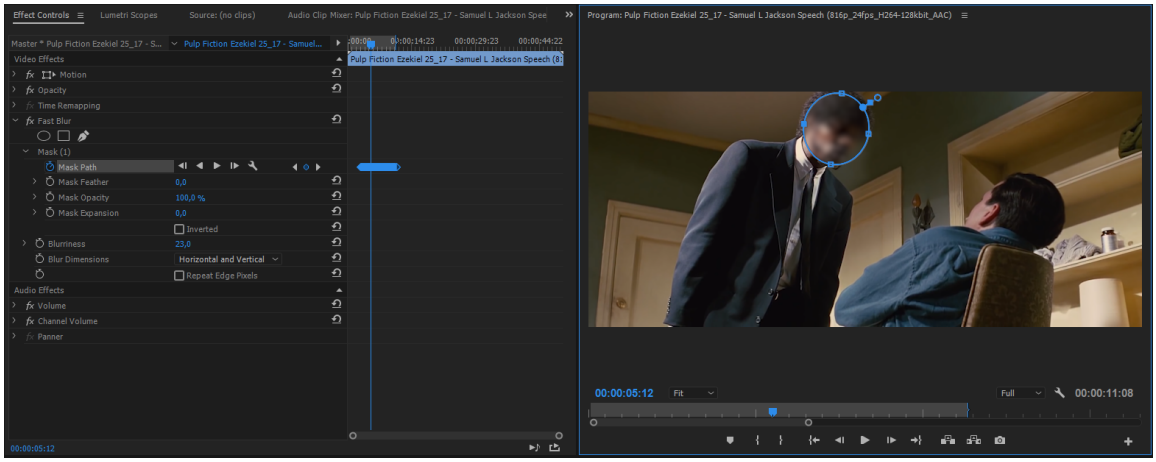
Tento nástroj funguje veľmi podobne ako Adobe Premiere Pro s tým rozdielom, že poskytuje pokročilejšie možnosti na sledovanie určitej časti obrazu. Na sledovanie je možné pridať do projektu novú vrstvu a do nej umiestniť tzv. Null Object, ktorému sa neskôr pridá trasa, ktorú sledoval tracker. Tracker je taktiež mierne pokročilejší oproti Adobe Premiere Pro a to tak, že spolu s vyznačením sledovaného objektu (oblasti) je možné vyznačiť aj oblasť, v ktorej sa má tento objekt nachádzať. To dáva užívateľovi viac možností, aby upravil túto

[11] https://www.adobe.com/cz/products/premiere.html

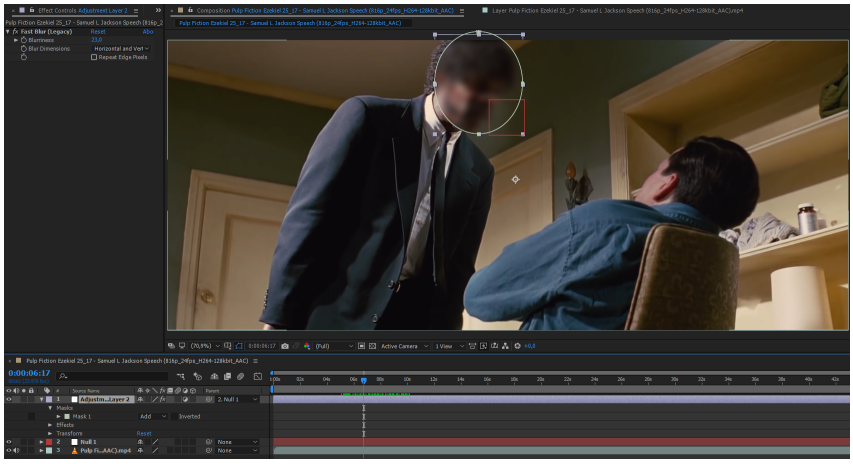
oblasť a zrýchlil, prípadne zefektívnil sledovanie požadovaného objektu. V prípade, kedy by došlo k výpadku trackeru je možné trasu upraviť manuálne.

Nakoniec užívateľ pridá novú vrstvu, na ktorú aplikuje požadovaný efekt anonymizácie. Túto vrstvu spojí následne s pohybom Null Objektu, ktorý má uloženú trasu z trackeru, takže sa vrstva s efektom pohybuje stále tam, kde sa nachádza aj objekt vo videu. Tento nástroj zvládne rovnaké efekty a aj ich použitie v rôznych tvaroch vybraných oblastí ako Premiere Pro. Jeho výhoda tkvie v tom, že je možné vytvoriť viacero takýchto vrstiev pre rôzne objekty vo videu a na každú vrstvu je možné skombinovať viacero rôznych efektov, teda v prípade, že užívateľ chce vytvoriť naozaj precíznu anonymizáciu, tak tento nástroj to veľmi dobre umožňuje.

Naopak nevýhodou môže byť mierne zložitejšie použitie, keďže sa jedná o komplikovanejší nástroj, ktorý ale poskytuje viac možností.



Obr. 3.11: Ukážka nástroja Adobe Premiere Pro CC 2017



Obr. 3.12: Ukážka nástroja Adobe After Effects CC 2017

Kapitola 4

Detekcia a sledovanie objektov

Ako už bolo spomenuté v kapitole 3.2, tak na to, aby bolo možné anonymizovať video, prípadne obraz, je potrebné poznať časti obrazu, ktoré je treba upraviť. Toto je možné urobiť manuálne, prípadne s následným využitím trackeru objektov, no často krát môže byť jednoduchšie použiť automatický detektor, najmä ak sa jedná o bežné objekty, pre ktoré existuje veľa druhov detektorov.

V tejto kapitole bude uvedených niekoľko typov detektorov, ktoré je možné práve na tento účel použiť. Nejedná sa o všetky existujúce detektory, ale práve o tie, ktoré sú buď najčastejšie používané, najznámejšie alebo ich používam vo svojom automatickom anonymizačnom systéme na detekciu tvarí a poznávacích značiek áut. Pokúsim sa čo najlepšie vysvetliť a objasniť ako jednotlivé typy fungujú a na aké použitie sú najvhodnejšie. Keďže je táto téma pomerne obsiahla a bolo by možné ju rozviesť na značne rozsiahly úsek, ktorý by nebol úmerný tejto práci, tak sa zameriam iba na ich základný princíp a niektoré aspekty nebudú vysvetlené do presných detailov.

Taktiež budú v tejto kapitole uvedené trackery objektov, bude popísaný ich princíp funkčnosti a uvedené ich klady, prípadne zápory.

4.1 Klasifikátory

Jedná sa o programy, ktoré na základe dát, ktoré dostanú na vstupe určia, či sa jedná alebo nejdená o daný objekt, na ktorý sú určené. Takéto klasifikátory sú väčšinou trénované na tisícoch až sto tisícoch pozitívnych a negatívnych obrázkoch, teda obrázkoch obsahujúcich daný objekt a takých, ktorý daný objekt neobsahujú. Z hľadiska vývoja sa jedná o pomerne zastaraný prístup k detekcii, keďže prvý takýto klasifikátor (podkapitola 4.1.1) popísali v roku 2001 Paul Viola a Michael Jones vo svojej práci *Rapid Object Detection using a Boosted Cascade of Simple Features* [22].

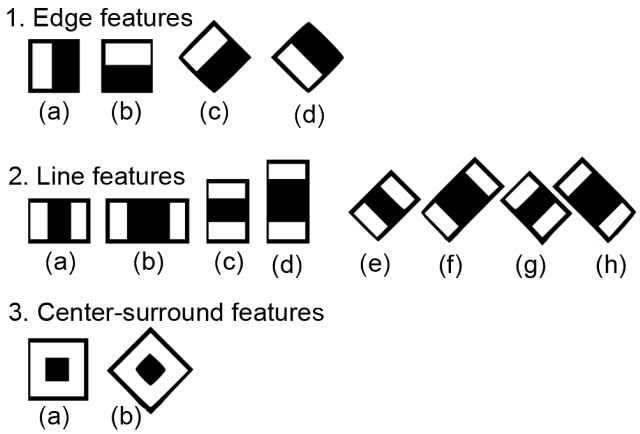
V tejto podkapitole sa zamieram najmä na dva typy kaskádových klasifikátorov, a to hlavne z dôvodu, že tieto dva typy obsahuje implicitne aj knižnica OpenCV[1], ktorú pri práci na systéme využívam, a tiež patrí k najpoužívanejším knižniciam na spracovanie obrazu.

4.1.1 HAAR

Jedná sa o klasifikátor[22], ktorý sa používa na detekciu objektov v obraze a k tomuto využíva vlastnosti obrazu, najmä hrán, ktoré sa v obraze nachádzajú. Pôvodne bol určený

[1] https://opencv.org/

na detekciu tvárí, ale je možné ho natrénovať aj na iné objekty. Tento systém pracuje iba s čierno bielymi obrázkami, pretože k detekcii hrán nie je potrebné poznať farby. Pri tomto type klasifikátora sa tieto vlastnosti alebo tiež funkcie nazývajú HAAR-like features a je ich niekoľko typov. Napríklad funkcie hrán či už zvislých alebo vodorovných, ohraničenia stredu a ďalších. Celkovo sa rozdeľujú na tri kategórie, a to dvoj-obdĺžnikové funkcie, troj-obdĺžnikové funkcie a štvor-obdĺžnikové funkcie. Rozdeľujú sa teda podľa toho, z koľkých obdĺžnikov vzniká konečný popis funkcie. Každá takáto funkcia je vyjadrená ako jediná hodnota, ktorá vznikne odpočítaním sumy pixelov pod bielou oblasťou od sumy pixelov pod čiernou oblasťou. Tieto oblasti je možné vidieť na nasledujúcom obrázku 4.1.



Obr. 4.1: Ukážka funkcii používaných v kaskádovom klasifikátore HAAR. Zdroj obrázku *https://docs.opencv.org*

Každá jedna hodnota funkcie pochádza z určitej časti trénovacieho obrazu s rôznou veľkosťou. Z toho teda vyplýva, že aj pre malé obrázky by bolo potrebné spočítať obrovské množstvo takýchto funkcií, a to v rôznych častiach obrazu, preto sa na urýchlenie používa tzv. integrálový obraz, ktorý už obsahuje predpočítané sumy rôznych častí trénovacieho obrazu pre rôzne pixeli, a tým sa urýchli jeho spracovanie.

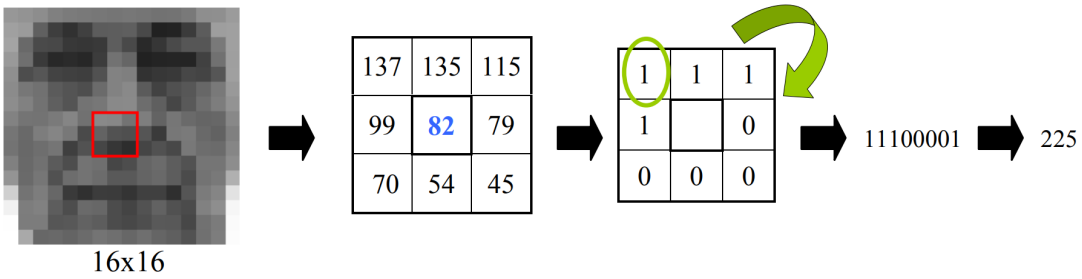
Avšak stále zostáva problém, že funkcií pre popis obrazu je príliš veľa a je potrebné ich zredukovať na menší počet. Na tento krok sa využíva algoritmus Adaptive Boosting, ktorý nájde najlepšie zodpovedajúce prahy hodnôt, ktoré popisujú obrazy s tvárami a bez nich. Samozrejme tieto funkcie a ich ideálne hodnoty sa trénujú na trénovacích sadách a následne sú uložené v tzv. váhových súboroch. Pre každú jednu funkciu následne vznikne tzv. slabý klasifikátor, ktorý dokáže určiť, či sa tá konkrétna funkcia nachádza v obraze alebo nie. Výsledný klasifikátor tvárí je tvorený až šiestimi tisícami týchto funkcií, ktoré spolu popisujú tvár človeka, jej rysy. Tento klasifikátor má základné rozlíšenie 24x24 pixelov, teda dokáže určiť, či na konkrétnom výreze obrazu s rozmermi 24x24 pixelov, sa nachádza alebo nenachádza tvár človeka.

Toto ale stále nestačí na detekciu tváre v obraze, keďže by bolo potrebné obraz, v ktorom sa snažíme nájsť tvár, rozdeliť na veľa oblastí s rozmermi 24x24 pixelov a na každej jednej z týchto oblastí spočítať a porovnať 6000 hodnôt popisujúcich ľudskú tvár. Takýto postup by bol veľmi pomalý a neefektívny, preto je klasifikácia rozdelená do stupňov alebo tiež kaskád (kaskádový klasifikátor), kedy každý stupeň obsahuje iba pár funkcií. V prípade, že sa na danom mieste v obraze tvár nenachádza, tak už v prvých stupňoch sa tento fakt

zistí. Daná oblasť sa teda vyhodnotí ako oblasť bez tváre a ďalej sa nespracováva. Tento kaskádový HAAR klasifikátor pozostáva až z 38 stupňov, pričom každý z nich obsahuje od jedného až po päťdesiat funkcií na popis vlastností tváre.

4.1.2 LBP – Local Binary Patterns

Tento klasifikátor [16] je založený na popise textúry obrazu. Základný princíp je podobný ako pri HAAR klasifikátore. Je rovnako trénovaný na veľkej sade pozitívnych a negatívnych obrázkov, teda tých čo obsahujú hľadaný objekt a tých čo nie. Základný variant sa trénuje tak, že trénovací obrázok rozdelí na bloky s rozmermi 9x9 pixelov (existuje viacero rozšírení). V týchto blokoch sa následne porovnáva prostredná hodnota bloku s jej okolitými hodnotami. Ak je okolitá hodnota väčšia, tak na jej miesto sa zapíše 1, inak 0. Tieto nové hodnoty, tvorené 1 a 0, sa prečítajú v smere hodinových ručičiek (niekedy proti smeru, záleží na variácii algoritmu) a vytvorí sa tak z nich jedno binárne číslo. Toto binárne číslo sa prepočíta na dekadickú hodnotu, ktorá je následne zapísaná do stredu bloku 9x9. Tento postup je znázornený na nasledujúcom obrázku 4.2.



Obr. 4.2: Znázornenie postupu používaného pre výpočet vektorov popisujúcich obrázky v algoritme LBP. Zdroj [16]

Takto sa postupuje pre každý pixel v obraze. Z nových hodnôt sa vytvorí histogram. Jeho hodnoty sa konvertujú a vzniká z nich viac rozmerný vektor, ktorý popisuje pôvodný obraz. Keďže ale pre trénovanie je použitých veľa obrázkov, tak na konci trénovania existuje veľa rôznych vektorov. Aby sa uľahčila následná klasifikácia, tak sa vektory popisujúce kladné obrázky, a rovnako aj vektory popisujúce záporné obrázky, spoja priemerovaním do jedného vektoru. Na toto môžu byť použité dve techniky, a to buď Chi Square klasifikácia alebo analýza hlavných komponent (PCA). Tieto dva výsledné vektory potom popisujú obrazy s tvárami, alebo hľadanými objektami a obrazy bez nich. Keďže vektory popisujúce obrazy s tvárami sú pomerne podobné (tváre sú podobné), tak pri ich zjednodušení nemusí prísť k veľkým chybám. Avšak vektory popisujúce okolie alebo obrazy bez tvárí môžu byť veľmi rozličné, a teda ich zjednodušením sa môže strácať informácia, preto sa môže používať aj postup klasifikácie, kedy sa berie do úvahy iba vektor popisujúci tváre a nastaví sa určitá hranica vzdialenosti (threshold), ktorá nesmie byť prekročená, aby objekt bol stále klasifikovaný napríklad ako tvár.

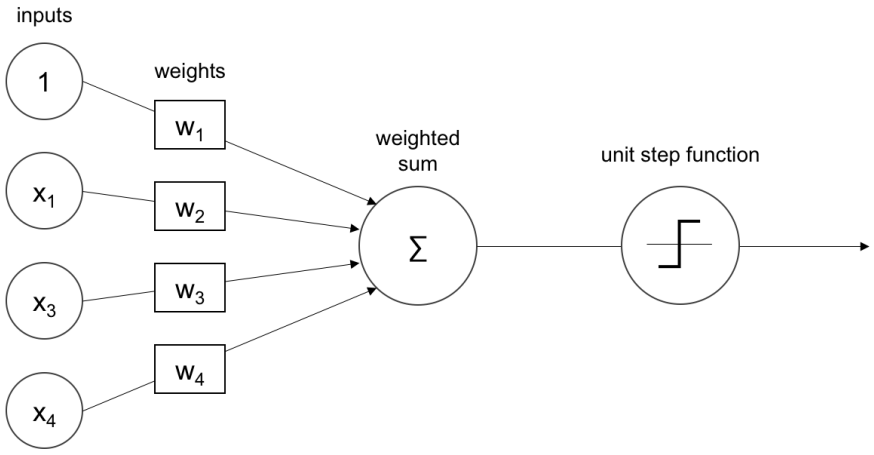
Pri detekcii objektov je časť vstupného obrázku najprv zanalyzovaná, podobne ako testovacie obrázky, a je popísaná viac rozmerným vektorom. Časť obrázku je možné vyberať pomocou viacerých metód, pričom každá z nich má určité výhody a nevýhody, či už pri úspešnosti detekcie alebo jej rýchlosti. Tento vektor následne môže alebo nemusí byť zjednodušený pomocou PCA. Vektor sa následne porovná s vektormi popisujúcimi kladné

a záporné obrázky, porovnanie prebieha pomocou SVM (Support Vector Machine) a toto porovnanie vráti odpoveď, či daná časť obrazu obsahuje tvár.

Existuje aj variant, kedy je táto vstupná analýza rozdelená na dve fázy, kedy sa v prvej iba určí, či obrázok môže byť kandidát na obrázok obsahujúci tvár a až v druhej fáze sa definitívne rozhodne. Na toto sa používa metóda nazývaná Skin segmentation. Použitie tohto postupu urýchľuje samotnú detekciu.

4.2 Neurónové siete

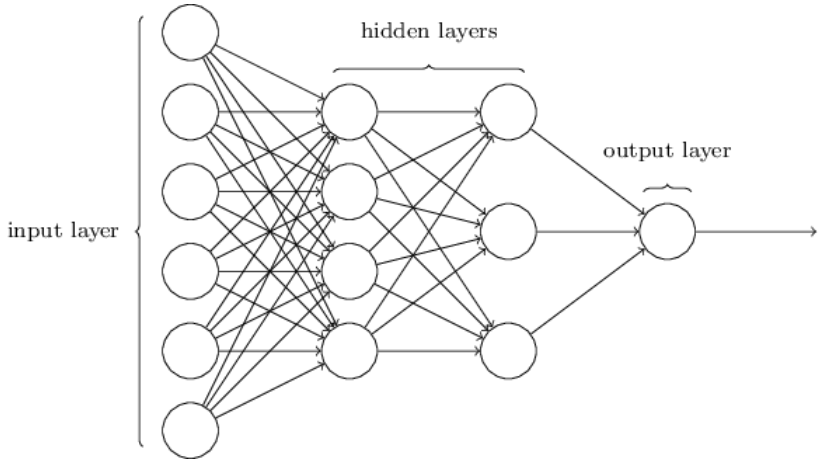
Nerónové siete [19] sú v podstate o orientovaný graf, ktorého jednotlivé uzly sú tzv. neuróny. Umelé neuróny sa v tomto prípade nazývajú aj perceptróny a podobne ako neuróny v ľudskom mozgu vykonávajú iba základné, jednoduché operácie. Dokážu klasifikovať lineárne separabilné priestory, teda napríklad logickú operáciu OR, ale XOR už nie. Každý takýto neurón má svoje vstupy, pričom každý vstup má určitú váhu a jeden výstup. Vstupy neurónu sú vstupom bázovej funkcie, ktorá produkuje iba jednu hodnotu, typicky sa používa lineárna bázová funkcia. Výstup bázovej funkcie je vstupom aktivačnej funkcie, ktorej výstup je aj výstup neurónu. Existuje viacero aktivačných funkcií, napríklad sigmoida, ReLU, Softmax a iné.



Obr. 4.3: Model perceptrónu. Zdroj obrázku: *https://towardsdatascience.com*

Existuje viacero architektúr neurónových sietí, no pre spracovanie obrazu sa využívajú najmä konvolučné neurónové siete. Tie budú detailnejšie popísané v ďalších častiach tejto kapitoly. Ich plne prepojená vrstva je plne prepojenou doprednou neurónovou sieťou. Takáto sieť môže mať až niekoľko vrstiev, pričom vrstva, ktorá je najbližšie vstupným hodnotám sa nazýva vstupná vrstva. Vrstva, ktorá je v sieti ako posledná a vystupujú z nej výstupné hodnoty sa nazýva výstupná vrstva. A medzi týmito vrstvami sa môže ešte nachádzať ľubovoľné množstvo skrytých vrstiev, viz. obrázok 4.4.

Trénovanie neurónovej siete prebieha ako postupná úprava váh jednotlivých vstupov neurónov v sieti. Tento proces prebieha nad nejakou dátovou sadou určenou na trénovanie a to tak, že ku každej cieľovej funkcii sa sieť snaží nájsť akúsi hypotézu, teda funkciu, ktorá sa čo najviac podobá tej pôvodnej. Jedná sa teda o učenie s učiteľom, kedy sú známe požadované hodnoty a na základe týchto sa určuje chyba, ktorú sieť pri určení hypotézy urobila. Tieto chyby sa následne berú do úvahy pri úprave váhových funkcií. Na toto sa najčastejšie



Obr. 4.4: Znázornenie neurónovej siete. Zdroj obrázku: *https://hackernoon.com.*

pri neurónových sieťach používa algoritmus spätného šírenia chyby. Tento algoritmus je založený na hľadaní minima stratovej funkcie[2], a to pomocou gradientného zostupu.

Výhody neurónových sietí spočívajú najmä v tom, že sú veľmi univerzálne a taktiež pri spracovávaní danej úlohy sú aj pomerne rýchle. Naopak nevýhodami môžu byť fatky, že neurónové siete sa dlho učia, pokiaľ sa jedná o zložitejší problém. Taktiež samotné učenie nie je práve jednoduchý proces a vyžaduje veľa skúšania, zmien parametrov, úprav trénovacej sady alebo správny odhad včasného ukončenia trénovania. V prípade, že je sieť príliš zložitá a nechá sa trénovať príliš dlho, tak dôjde k pretrénovaniu. To znamená, že bude reagovať iba na dáta z trénovacieho datasetu a na žiadne iné. Taktiež väčšie neurónové siete sú pomerne pamäťovo náročné, keďže môžu obsahovať veľmi veľké množstvo vrstiev a váh jednotlivých neurónov.

Konvolučné neurónové siete

Konvolučné neurónové siete sú zvláštny prípad neurónových sietí, ktoré spracovávajú najmä štruktúrované dáta, teda dáta, ktoré dávajú zmysel iba ak majú určitú štruktúru. Typicky sa môže jednať napríklad o obraz. Rozdielom je tiež, že je pri nich zachovaná dimenzionalita vstupu počas celého prechodu sieťou až po poslednú plne prepojenú vrstvu, teda každá vrstva má šírku, výšku a hĺbku. Jadrom konvolučných sietí sú práve konvolučné vrstvy, ktoré fungujú ako filtre. Tie postupne prechádzajú vstupné dáta po blokoch a z jednotlivých buniek robia pomocou konvolučného jadra jednu hodnotu. Nejedná sa ale o pravú konvolúciu, ale o tzv. cross-corelation, teda nedôjde k otočeniu filtru tak ako pri konvolúcii.

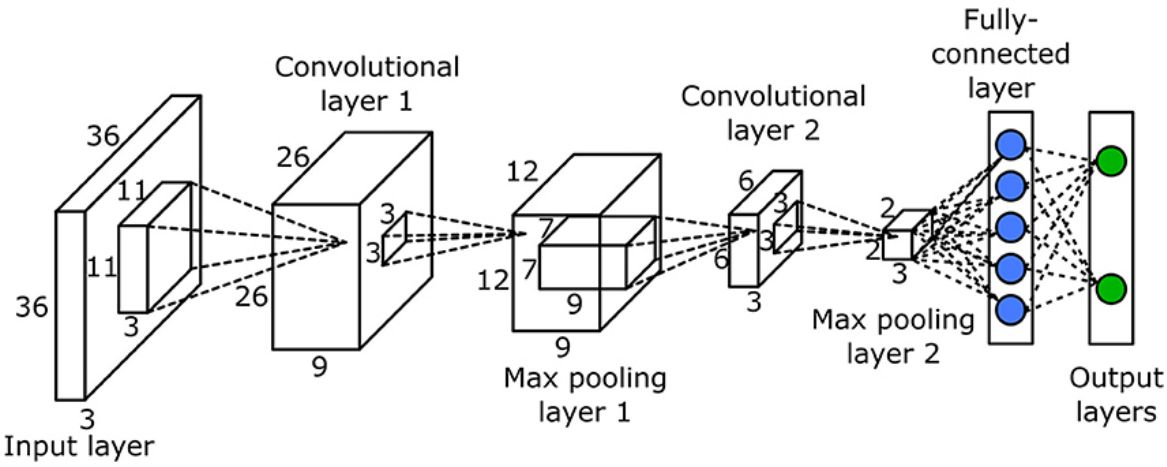
Trénovanie takýchto sietí funguje podobne ako pri obyčajných neurónových sieťach, taktiež je využívaný algoritmus spätného šírenia chyby, avšak medzi jednotlivými vrstvami je mierne zmenený.

Konvolučné siete sa skladajú typicky z viacerých vrstiev a tie môžu byť rôzne naskladané za seba. Konvolučná vrstva už bola spomenutá vyššie. Ďalšími vrstvami môžu byť napríklad poolingová vrstva, ktorá sa používa pre zmenšenie veľkosti vstupu do ďalšej vrstvy, typicky to býva napríklad funkcia na nájdenie maxima z viacerých hodnôt. Taktiež sa tu môže nachádzať aktivačná vrstva, drop-out vrstva alebo ako posledná väčšinou býva umiestňovaná

[2]Vyjadruje aktuálnu chybu siete behom trénovania, teda je to rozdiel medzi aktuálnym výstupom a očakávaným výstupom. Typicky sa používajú stredná kvadratická chyba alebo Cross-entropy.

plne prepojená vrstva. Práve táto posledná vrstva má typicky rozmery vstupného objektu, a to z dôvodu, že jej výstupy by mali zodpovedať rozmerom vstupných dát.

V nasledujúcich podkapitolách budú uvedené niektoré typy konvolučných neurónových sietí, ktoré je možné použiť práve na detekciu objektov v obraze. Tieto siete existujú vo veľkých množstvách modifikácii a každá modifikácia má inú presnosť, časovú a pamäťovú náročnosť. Avšak sú založené na princípe, ktorý je uvedený vyššie.



Obr. 4.5: Znázornenie konvolučnej neurónovej siete. Zdroj obrázku: *https://www.frontiersin.org.*

4.2.1 R-CNN

R-CNN [8] alebo tiež Region Convolution Neural Network dokáže detegovať rôzne objekty na vstupnom obrázku. K tomuto využíva celkovo 3 moduly. Prvý modul odhadne približne 2000 oblastí záujmu na danom obraze. To z dôvodu, aby sa nemusela sieť spúšťať nad všetkými regiónmi, ktorých môže byť pri rôznych veľkostiach a tvaroch príliš veľa. Tieto regióny sú nájdené pomocou selektívneho vyhľadávania nad daným obrázkom.

Druhý modul následne nad každým regiónom vypočíta 4096 dimenzionálny vektor, a to pomocou konvolučnej neurónovej siete, napísanej vo frameworku Caffe[3]. V treťom module sa pre každú triedu objektov spustí SVM[4], ktorý je natrénovaný na danú triedu. Ten ohodnotí každý vektor pre všetky triedy a výsledky vráti ako výstup. Vo výsledku teda dostaneme pre každý z 2000 regiónov ohodnotenie s akou pravdepodobnosťou sa v ňom nachádza objekt určitej triedy.

Tento typ detektoru teda dokáže s pomerne dobrou presnosťou detegovať objekty v obraze (presnosť 53.7% mAP[5] [8] na datasete VOC 2010[6]), ale taktiež má aj svoje nevýhody. Jednou z nevýhod je príliš dlhý čas potrebný na učenie siete, keďže nad každým trénovacím obrázkom je potrebné klasifikovať až 2000 regiónov a trénovacích obrázkov môžu byť až sto tisíce. Taktiež nefunguje v reálnom čase a na detekciu potrebuje rádovo až desiatky sekúnd, záleží od výpočtového výkonu. A algoritmus selektívneho vyhľadávania, ktorý zisťuje re-

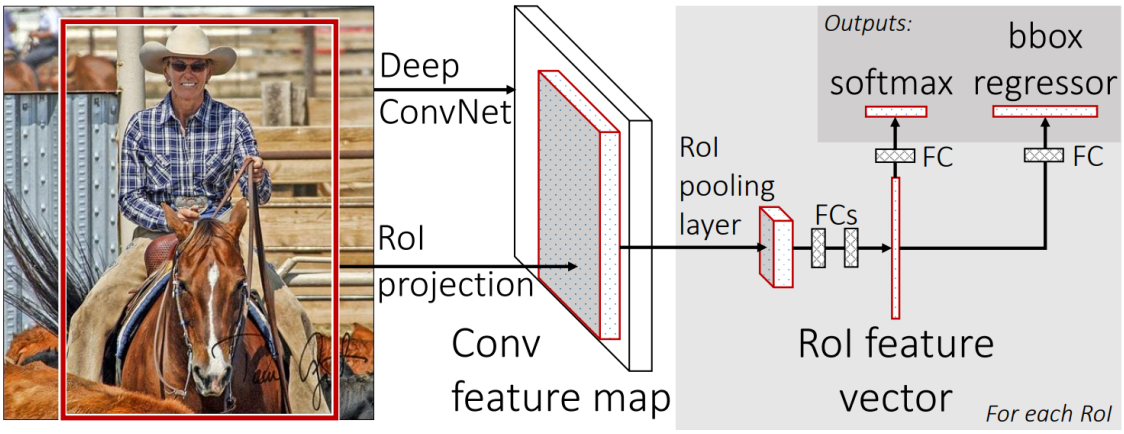
[3] http://caffe.berkeleyvision.org/
[4] Support-vector machine
[5] mean Average Precision
[6] http://host.robots.ox.ac.uk/pascal/VOC/voc2010/

gióny záujmu je pevný algoritmus, takže počas jeho behu nedochádza k žiadnemu učeniu, prípadne zlepšeniu hľadania.

4.2.2 Fast R-CNN

Tento detektor [7] je vylepšením R-CNN, ktorý bol príliš pomalý a potreboval veľké množstvo pamäte. Vylepšenie spočíva v tom, že sa nehľadajú regióny záujmov pomocou selektívneho vyhľadávania, ale celý vstupný obrázok sa pošle do prvej vrstvy neurónovej siete spolu s návrhmi pozícii objektov. Táto vrstva vytvorí konvolučnú mapu vlastností (convolutional feature map), ktorá obsahuje návrhy regiónov, v ktorých sa nachádzajú dané objekty. Následne sa každá z týchto oblastí pošle na druhú vrstvu (RoI[7] pooling layer), ktorá vyberie vektor z konvolučnej mapy vlastností, teda popis objektu, ktorý sa v oblasti nachádza. Každý takýto vektor je následne poslaný do dvoch súbežných vrstiev, pričom prvá z nich vráti pravdepodobnosť výskytu určitého objektu v danej oblasti a druhá vráti 4 číselné hodnoty, ktoré určujú presný región (bounding box) v ktorom sa objekt v obraze nachádza.

Oproti pôvodnej verzii R-CNN sa Fast R-CNN zrýchlilo veľmi výrazne, a to ako pri detekcii objektov, tak aj pri trénovaní siete. Toto zrýchlenie nastalo najmä kvôli faktu, že sa nemusí pre 2000 oblastí spúšťať celá sieť, ale spustí sa iba raz pre všetky oblasti záujmu a nájde presné návrhy regiónov, ktoré je treba klasifikovať. Jej presnosť sa tiež oproti R-CNN zvýšila na 66% mAP [7] na datasete VOC 2012[8].



Obr. 4.6: Znázornenie architektúry detektora Fast R-CNN. Zdroj [7]

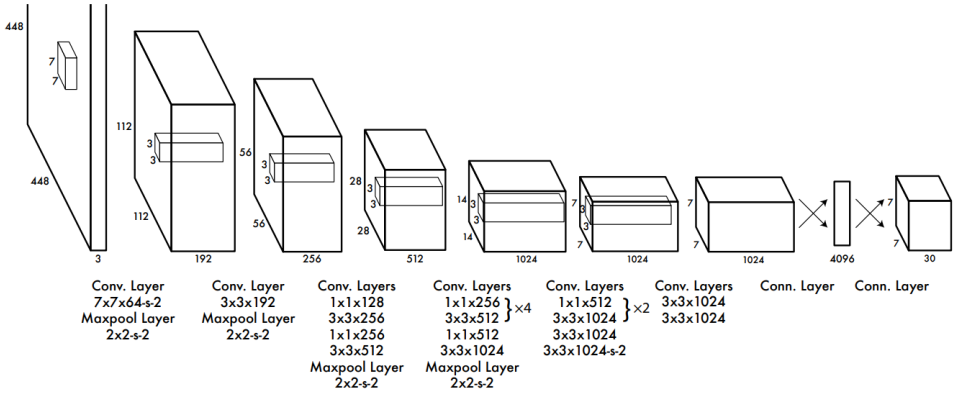
4.2.3 YOLO

You Only Look Once je detektor [18], ktorý používa úplne odlišný prístup oproti R-CNN. Odlišný je v tom, že nepoužíva pri detekcii iba oblasti záujmu, ale celý obraz. Na vstupe teda dostane celý obraz, ktorý zmení rozmer 448x448 pixelov. Na takto zmenenom obraze spustí jednu konvolučnú neurónovú sieť, ktorá predikuje bounding boxy pre objekty, to aj spolu s klasifikáciou a pravdepodobnosťou o aký objekt sa jedná. Potom sa vyberú tie

[7] Region of Interest
[8] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

oblasti, ktoré majú najvyššiu pravdepodobnosť výskytu daného objektu a tie sa vrátia ako výstup siete vo výstupnom obrázku. Takýmto spôsobom prebieha detekcia objektov a aj trénovanie siete.

Vďaka tomuto prístupu má YOLO k dispozícií kontext celého obrázku, a teda sa tak zmenší počet falošných detekcií. Taktiež je možné detegovať objekty aj na obrázkoch väčších rozmerov, a to v reálnom čase. V prípade použitia varianty tiny-YOLO je možné robiť detekciu aj na menej výkonných zariadeniach s menším množstvom operačnej pamäte a výpočtového výkonu. Na druhej strane YOLO zle deteguje malé objekty v obraze, to najmä z dôvodu, že rozlíšenie siete je iba 448x448 pixelov, čo pri menších objektoch nemusí stačiť. Rovnako je problém, keď sa niektoré objekty nachádzajú príliš blízko pri sebe, tak nemusia byť správne detegované.



Obr. 4.7: Znázornenie architektúry detektora YOLO. Zdroj [18]

4.3 Trackery

Vzhľadom k tomu, že videá obsahujú zväčša veľké množstvo snímok, tak by bolo potrebné pre každý nový snímok robiť detekciu objektov znovu. Toto riešenie ale nie je žiaduce, keďže detekcia objektov v obraze je časovo náročná a spomalila by tak spracovanie videa. V niektorých prípadoch až veľmi výrazne[9]. Z tohto dôvodu je teda potrebné robiť detekciu čo možno najmenej a využívať iné techniky na zistenie polohy objektu medzi detekciami. Na to slúžia práve trackery. Ako bolo už spomenuté v kapitole 3.2, tak existuje niekoľko druhov trackerov. V tejto kapitole sa zamieram na tie, ktoré sú dostupné v knižnici OpenCV a je ich teda možné využiť pri vytváraní automatického anonymizačného systému. Pre každý z týchto trackerov bude popísaný jeho princíp činnosti a následne v kapitole 5.2 bude vyhodnotené, ktorý z nich je ideálny pre automatický anonymizačný systém. V tejto kapitole budú uvedené trackery pod názvami, pod akými sú implementované v knižnici OpenCV. Presné názvy týchto trackerov je možné nájsť v použitej literatúre.

[9]Veľa objektov na každom snímku, malé objekty

4.3.1 Boosting

Tento tracker [9] je založený na algoritme Adaptive Boosting, ktorý je použitý aj v klasifikátore HAAR, kapitola 4.1.1. Princíp jeho funkčnosti spočíva v tom, že je trénovaný priamo za behu, a to pomocou pozitívnych a negatívnych príkladov. Na začiatku dostane na vstupe región, ktorý má sledovať, teda pozitívny príklad. Ako negatívne príklady berie množstvo regiónov z pozadia. Pri nasledujúcom snímku sa overí okolie pôvodného regiónu z predchádzajúceho obrázku a spočíta sa skóre v jednotlivých častiach. Tam kde je skóre najväčšie, sa nachádza sledovaný objekt. Nová pozícia objektu sa následne berie ako ďalší pozitívny príklad pre ďalšie iterácie. Tento algoritmus je ale pomerne zastaraný a dnes sa príliš nepoužíva. Taktiež tento tracker nedokáže spoľahlivo určiť, kedy sa stratí sledovaný objekt, a teda nie je možné presne určiť čas, kedy je potrebná opätovná inicializácia.

4.3.2 MIL

Je podobný ako Boosting tracker, avšak s niekoľkými podstatnými rozdielmi. MIL [1] tracker neberie ako pozitívny príklad iba jeden obrázok. V tomto prípade sa berie ako pozitívny príklad hneď niekoľko regiónov, ktoré susedia s označeným regiónom pre sledovanie. Tieto jednotlivé regióny sa následne berú ako akýsi zhluk, ktorý obsahuje práve jednu presnú pozíciu objektu. Tým sa zaistí, že sledovaný objekt sa bude nachádzať v centre tohto zhluku a zvýši sa tým presnosť sledovania. Oproti trackeru Boosting dosahuje lepšiu presnosť a pozícia sledovaného objektu príliš „neskáče“.

4.3.3 KCF

Tento tracker [10] využíva podobné princípy ako predchádzajúci tracker MIL. Kým jeho princíp využíval zhluky regiónov, ktoré sa navzájom prekrývali, tak KCF tracker práve tieto prekrývajúce sa časti zjednodušuje, tým že ich považuje za rovnaké. Takýmto prístupom urýchľuje spracovanie regiónov a zároveň šetrí aj pamäťové nároky pri behu algoritmu. Oproti trackerom MIL a Boosting dosahuje lepšie výsledky v presnosti sledovania objektov a aj v rýchlosti. Lepšie tiež dokáže rozpoznať výpadok pri sledovaní a umožní tak, aby bolo sledovanie znovu inicializované. Tento tracker je dostupný v OpenCV od verzie 3.2.

4.3.4 TLD

TLD [12] tracker je založený z troch častí: trackeru, detektoru a učenia. Tracker sleduje objekt z jednej snímky na druhú. Detektor lokalizuje všetky predchádzajúce výskyty objektu v minulých snímkach a následne v prípade potreby, upraví výslednú pozíciu objektu na aktuálnom snímku. Učenie odhaduje priebežné chyby detektora a upravuje jeho parametre pre zvýšenie presnosti v budúcich detekciách. Tieto tri časti teda spolu vytvárajú celok, ktorý je schopný sledovať požadovaný objekt a to aj v prípade, že objekt na chvíľu zmizne z obrazu. Nevýhodou tohto trackeru je ale množstvo falošne pozitívnych regiónov, ktoré vznikajú, ak sa v obraze nachádzajú podobné objekty ako objekt, ktorý je sledovaný. Tento tracker sa tiež dokáže dobre vysporiadať aj so zmenou veľkosti objektu počas jeho sledovania.

4.3.5 MedianFlow

Tracker [11] sleduje objekty v dvoch časových smeroch, dopredu v čase a aj dozadu v čase. Výsledné dve hodnoty následne porovná a na základe rozdielov medzi nimi určí chybu sle-

dovania. Takýmto spôsobom dokáže táto metóda veľmi presne určiť, kedy príde k výpadku sledovania. Taktiež dokáže predpovedať budúci pohyb objektu na základe jeho minulého pohybu. Tento tracker sa hodí na sledovanie objektov, ktoré sa nehýbu príliš rýchlo a tiež v prípadoch, kedy je potrebné presne určiť výpadok trackeru.

4.3.6 MOSSE

Skratka pre Minimum Output Sum of Squared Error [5]. Na sledovanie objektov používa adaptívnu koreláciu, ktorá dokáže vytvoriť stabilné korelačné filtre pri inicializácii iba z jednej snímky. Sledovanie založené na takýchto filtroch dokáže poskytnúť kvalitné sledovanie objektov nezávislé na zmenách osvetlenia, veľkosti alebo polohy. Tento tracker dokáže rozpoznať, ak objekt zmizne zo scény a v takomto prípade sa pozastaví až do doby, kedy sa objekt znovu v scéne objaví. Je schopný vyhodnotiť viac ako 450 snímok za sekundu, a teda je výrazne rýchlejší ako ostatné trackery. Na druhú stranu, presnosť sledovania v mnohých prípadoch nie je príliš vysoká.

4.3.7 CSRT

Tento tracker [15] využíva diskriminatívne korelačné filtre (DCF), ktoré sa osvedčili ako výkonná metóda na sledovanie objektov. Pridáva k nim však kanálovú a priestorovú spoľahlivosť v podobe mapy - pre upravovanie týchto filtrov. To zabezpečí, že v sledovanom regióne sa bude sledovať len tá časť, ktorá je na sledovanie najvhodnejšia. Vďaka tomuto prístupu je možné zväčšovať sledovaný región bez zníženia rýchlosti a zároveň to umožňuje lepšie sledovanie neobdĺžnikových tvarov. Táto metóda by mala byť schopná poskytnúť real-time výkon na CPU, avšak veľmi záleží na výkone daného CPU. Spomedzi ostatných trackerov poskytuje najvyššiu presnosť sledovania, ale za cenu nižšej rýchlosti.

Kapitola 5

Automatická anonymizácia videí

V prechádzajúcich kapitolách, ktoré pojednávali najmä o teoretických základoch potrebných pre návrh a vývoj automatického anonymizačného systému, boli vysvetlené a popísané všetky potrebné znalosti a techniky na vytvorenie takéhoto systému. V tejto kapitole sa preto zameriam na opis návrhu a vývoja už s konkrétnymi použitými nástrojmi. Rozoberiem všetky potrebné aspekty, ktoré bolo treba počas implementácie zistiť, či už experimentálne, meraniami alebo mnohými pokusmi na testovacej sade videí. Táto kapitola bude obsahovať vyhodnotenie použitých nástrojov, metód a knižníc. Taktiež tu bude uvedený návrh systému a aj opis algoritmu na ktorom funguje, aké časti obsahuje a tie dôležitejšie budú popísané podrobnejšie.

Vzhľadom k faktu, že tento systém má byť vo výsledku plne automatický, boli použité nástroje na detekciu (kapitoly 4.1 a 4.2) a sledovanie (kapitola 4.3) objektov v obraze/videu. Niektoré detekčné nástroje bolo potrebné aj rôzne skombinovať a následne otestovať, toto bude popísané v nasledujúcich podkapitolách.

Systém bude implementovaný pomocou knižnice OpenCV, ktorá poskytuje takmer všetky potrebné nástroje pre jeho vytvorenie. Samotná implementácia bude napísaná v programovacom jazyku Python vo verzii 3.

5.1 Experimenty a merania s detektormi objektov

Pred samotnou implementáciou systému bolo potrebné experimentálne zistiť, ktoré nástroje či už detekčné alebo sledovacie poskytujú najlepšie výsledky. V tejto podkapitole bude rozobraný postup experimentu, v ktorom boli porovnané a vyhodnotené rôzne detekčné nástroje.

Pred samotným vyhodnocovaním bolo potrebné ustanoviť jednotlivú metriku, na ktorej sa bude vyhodnocovať. Teda použiť rovnakú sadu obrázkov, ktoré sa budú detegovať a následne aj rovnaký nástroj na vyhodnotenie tejto detekcie. Výsledný systém by mal byť schopný detegovať dva druhy objektov, a to tváre ľudí a štátne poznávacie značky vozidiel. V prvej časti sa zameriam na experiment s detektormi tvárí.

5.1.1 Detektory tvárí

Existuje viacero voľne dostupných detektorov na ľudské tváre. Z tohto dôvodu som sa rozhodol vo vyvíjanom systéme použiť už tie existujúce a nevytvárať vlastné. Otestoval som tri dostupné detektory, jeden založený na princípe klasifikátoru a ďalšie dva založené na princípe neurónových sietí. Konkrétne sa jednalo o `Haar Feature-based Cascade Classifier`

(kapitola 4.1.1), `GoogLeNet`[1] `neural network` s vopred natrénovaným modelom založeným na `Cafee Zoo`[2] a detektorom `MobileNetSSD`[3] s modelom natrénovaným na datasete `WIDER FACE`[4], implementovaným pomocou `TensorFlow API`[5]. Detektory sú použité s nasledujúcimi detekčnými jadrami *(modelmi)* a pomocou nasledujúcich metód:

- `Haar Feature-based Cascade Classifier`:
  - Detekčné jadro: `haarcascade_frontalface_default.xml`[6]
  - Použitie: pomocou modulu `Cascade Classifier`[7] v knižnici OpenCV.
- `GoogLeNet`:
  - Detekčné jadro: `res10_300x300_ssd_iter_140000_fp16.caffemodel`[8]
  - Použitie: pomocou modulu `dnn`[9] v knižnici OpenCV.
- `MobileNetSSD`:
  - Detekčné jadro: `frozen_inference_graph_face.pb`[10]
  - Použitie: `TensorFlow API`

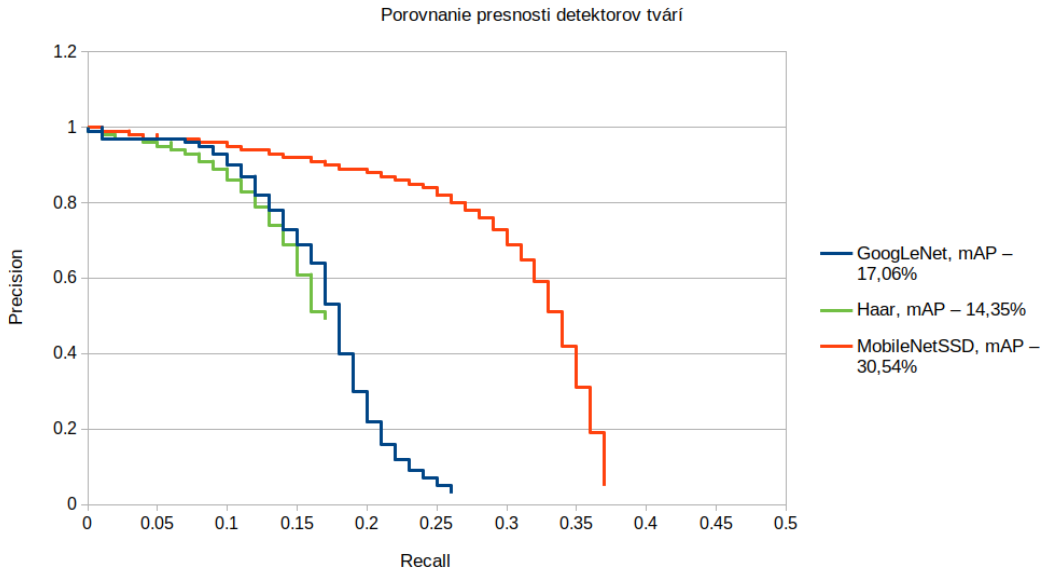
Všetky tieto detektory som testoval na rovnakej dátovej sade získanej z datasetu *WIDER FACE*[11], avšak na jej validačnej a nie trénovacej časti. Jednalo sa konkrétne o 3 226 obrázkov, na ktorých boli označené ľudské tváre. Tento dataset sa dá považovať za jeden z ťažších, keďže na mnohých obrázkoch sa nachádzali niekedy až stovky tvárí, pričom niektoré mali rozlíšenie iba pár jednotiek až desiatok pixlov.

Tento dataset bolo potrebné pred použitím upraviť na požadovaný formát. Skripty, pomocou ktorých som testoval detektory som vytvoril tak, že vyžadovali v jednom priečinku všetky testovacie obrázky a v druhom priečinku všetky anotácie k týmto obrázkom. Teda napríklad obrázok `1.jpg` má anotácie uložené v súbore `1.txt`. Pomocou jednoduchého skriptu som dataset konvertoval na mnou požadovaný formát.

Na vyhodnotenie presnosti detektorov som používal nástroj[12], ktorý vypočítal presnosť podľa metriky definovanej počas výzvy PASCAL VOC 2012[13]. Jedná sa o metriku, ktorá vracia výsledok ako hodnotu mAP[14] v percentách. Keďže hodnota mAP sa počíta ako priemer strednej presnosti pre viacero tried, tak v mojom prípade zodpovedala presnosť mAP aj presnosti AP tvárí, keďže tváre boli jediná testovaná trieda.

Pre lepšiu predstavu sa dá skóre mAP aj vizualizovať, a to pomocou Precision Recall krivky. Plocha pod touto krivkou je skóre mAP. V grafe 5.1 je možné vidieť porovnanie presností jednotlivých detektorov na testovacej sade.

[1] https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf
[2] http://caffe.berkeleyvision.org/model_zoo.html
[3] Neurónová sieť MobileNet s detekčnou vrstvou SSD (Single Shot multibox Detector)
[4] http://shuoyang1213.me/WIDERFACE/
[5] https://www.tensorflow.org/
[6] https://github.com/opencv/opencv/tree/master/data/haarcascades
[7] https://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
[8] https://github.com/spmallick/learnopencv/tree/master/FaceDetectionComparison/models
[9] https://docs.opencv.org/3.4/d6/d0f/group__dnn.html
[10] https://github.com/yeephycho/tensorflow-face-detection/tree/master/model
[11] http://shuoyang1213.me/WIDERFACE/
[12] https://github.com/Cartucho/mAP
[13] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
[14] mean Average Precision



Obr. 5.1: Porovnanie detektorov tvárí na datasete WIDER FACE. Nastavené prahy istoty pri zobrazených presnostiach na grafe boli: **Haar** – minimálny počet susediacich detekcii: 1. **GoogLeNet** – prah istoty: 0,08. **MobileNetSSD** – prah istoty: 0,01.

Ako je možné z grafu vyčítať, tak najmenej presný bol detektor Haar a naopak najlepšie výsledky dosiahol detektor MobileNetSSD. Krivka detektoru Haar nedosahuje precision hodnôt menších ako 0,5 z dôvodu, že nebolo možné nastaviť menší minimálny počet susediacich detekcii ako 1.

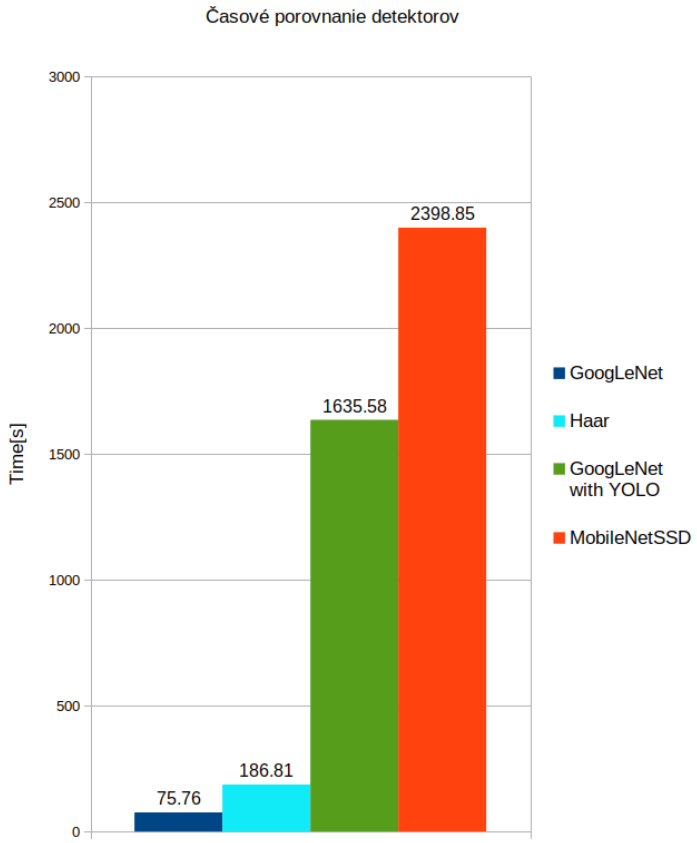
Ak chceme, aby porovnanie bolo úplné, tak je potrebné porovnať aj časovú náročnosť jednotlivých detektorov. Keďže v navrhovanom systéme je optimálne riešenie - čo možno najrýchlejšie a zároveň dostatočne presné. Z toho dôvodu som porovnal aj priemerné dosiahnuté časy detektorov, pričom každý z nich detegoval objekty na rovnakej sade 20× a zakaždým s iným prahom istoty detekcie[15]. V grafe 5.2 je možné vidieť porovnanie týchto priemerných časov.

Najpresnejší detektor MobileNetSSD je až 31× pomalší ako najrýchlejší GoogLeNet. Detektor Haar bol najmenej presný a nebol ani najrýchlejší, z toho dôvodu som ho z ďalšieho experimentovania odstránil a nepovažoval ho za vhodný pre vytváraný systém.

Detektor GoogLeNet sa ukázal ako veľmi rýchly, no málo presný - najmä, čo sa týkalo menších tvárí. Rozhodol som sa teda, že ho skúsim spojiť s pred-detekciou iným detektorom, a tak jeho presnosť zvýšiť. Vzhľadom k faktu, že sa jedná o detekciu tvárí, tak logickou pred-detekciou je detekcia osôb. Na takúto detekciu som si zvolil detektor `YOLO` [18], ktorý už obsahuje natrénovaný model pre 80 tried objektov. Jednou zo spomínaných tried sú aj osoby. Spojenie detekcií fungovalo v praxi tak, že najprv prebehla detekcia tvárí iba pomocou GoogLeNet a následne prebehla detekcia osôb pomocou YOLO. Pre každú detekovanú osobu sa v jej regióne detekovala aj tvár pomocou GoogLeNet. Tváre, ktoré boli detekované samotným GoogLeNet bez pred-detekcie mali väčšiu váhu ako tie detekované s pred-detekciou. Takže, ak sa jedna tvár detekovala dvakrát, pomocou GoogLeNet a aj s pred-detekciou YOLO, tak boli vrátené iba súradnice tváre zo samostatnej detekcie

[15] Confidence threshold

GoogLeNet. Takéto fungovanie bolo implementované z dôvodu, že pri detekcii osoby môže dôjsť k chybám a región tváre teda môže byť poškodený *(zrezaný z rôznych strán)*. Na druhú stranu by mal takýto princíp zachytiť aj menšie tváre, ktoré samotným detektorom GoogLeNet detekované neboli a zvýšiť tak presnosť a mieru detekcie. Pre potreby písania tejto správy nazvime tento detektor ako `GoogLeNet with YOLO`.

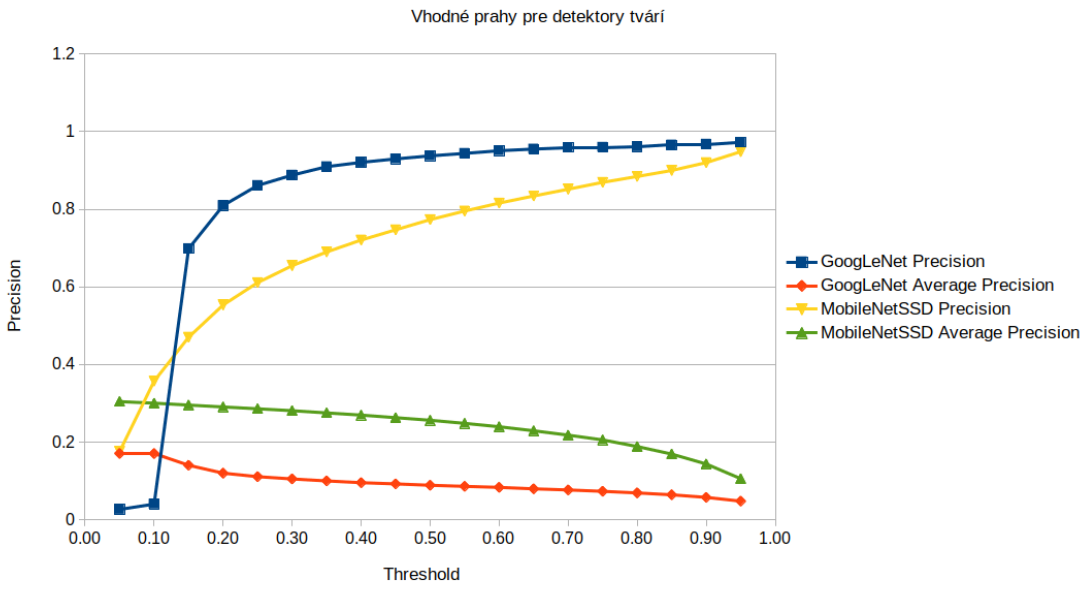


Obr. 5.2: Porovnanie časov detekcie rôznych detektorov tvárí. Zobrazená hodnota je čas v sekundách, po ktorý trvalo detektoru spracovať všetkých 3 226 obrázkov v dátovej sade. Tieto časy boli dosiahnuté na CPU Intel 6600K s frekvenciou 4,25 Ghz.

Ďalším problémom, ktorý bolo treba experimentálne vyriešiť, bolo zistenie ideálnej hodnoty pre prah istoty detekcie. Keďže merania presnosti zobrazené v grafe 5.1 mali nastavené tieto hodnoty veľmi nízko z dôvodu presnosti grafu, tak ich výstupy obsahovali viac falošne pozitívnych *(FP)*[16] detekcií ako tých pravdivo pozitívnych *(TP)*[17]. Bolo teda potrebné nájsť také hodnoty prahov, ktoré by zabezpečili, že TP bude väčšie ako FP a zároveň presnosť *(average precision)* detektora nebude príliš nízka *(so zvyšujúcim sa prahom sa znižuje presnosť detektora)*. V rámci tohto experimentu som testoval iba dva detektory, a to GoogLeNet a MobileNetSSD. V grafe 5.3 je možné vidieť výstup experimentu.

[16]Falošne pozitívna detekcia – detektor vráti súradnice objektu aj keď sa na týchto súradniciach žiadny hľadaný objekt nenachádza.

[17]Pravdivo pozitívna detekcia – detektor vráti súradnice objektu, ktorý sa na daných súradniciach skutočne nachádza.

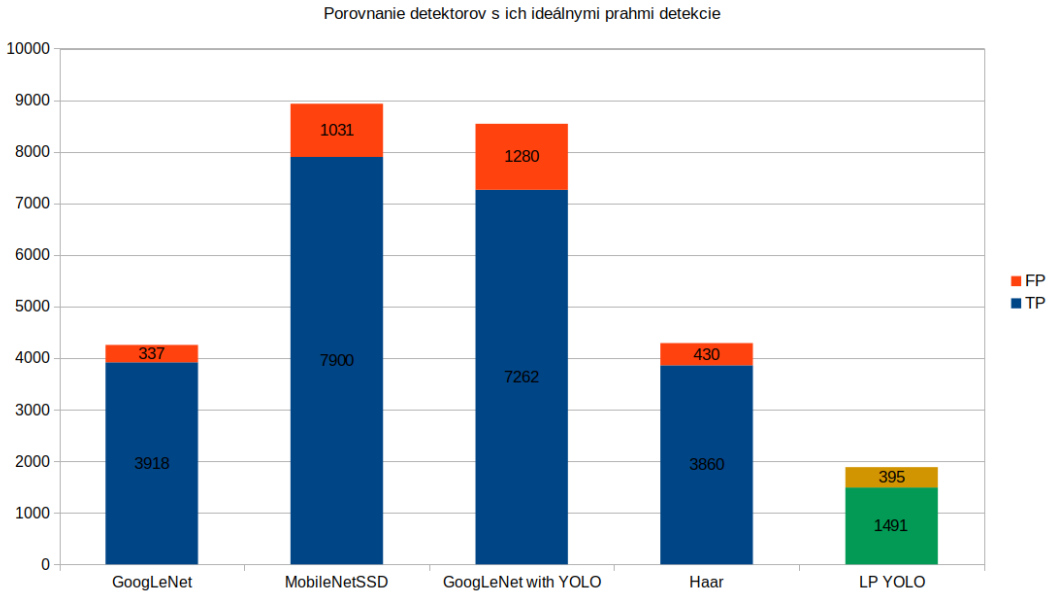


Obr. 5.3: Zistenie ideálneho prahu istoty pre detekciu tvárí. Pričom precision je v tomto grafe hodnota TP / Všetky detekované objekty. A average precision je hodnota mAP.

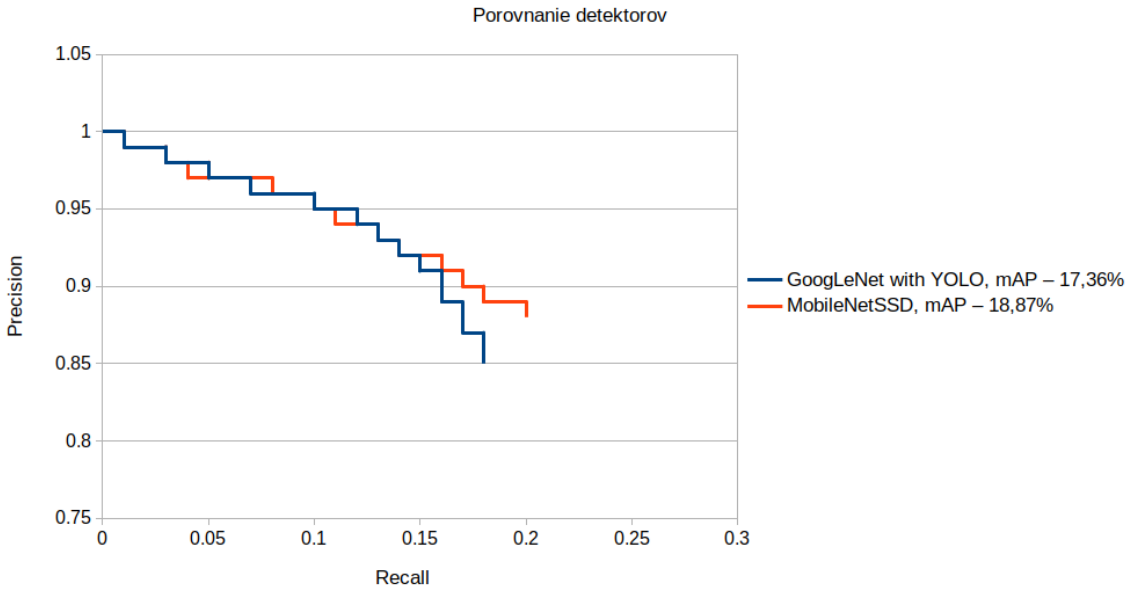
Pre detektor GoogLeNet z grafu vychádza ideálna hodnota prahu 0,4 a pre detektor MobileNetSSD je ideálna hodnota 0,8. Vďaka takto získaným prahom pre každý detektor bolo možné porovnať aj presnosť a mieru detekcie medzi detektormi MobileNetSSD a navrhnutým detektorom GoogLeNet with YOLO. Pre navrhnutý detektor som ideálne hodnoty prahov zistil experimentálne, avšak nie skúšaním všetkých možností *(dva detektory – dva prahy, jeden pre GoogLeNet a druhý pre YOLO)*, ale postupným skúšaním niektorých vopred odhadnutých hodnôt. Zistil som, že ideálne hodnoty prahov sú 0,7 a 0,7 pre oba detektory, z ktorých sa GoogLeNet with YOLO skladá.

V grafe 5.4 je možné vidieť porovnanie všetkých detektorov na základe TP/FP pomeru, s ich ideálnymi nastavenými hodnotami prahov pre detekcie. Je jasne vidieť, že detektor MobileNetSSD je stále mierne lepší ako detektor GoogLeNet with YOLO, ale tento rozdiel je zanedbateľný v porovnaní s výsledkami časového porovnania z grafu 5.2, kde GoogLeNet with YOLO je rýchlejší takmer 1,5× ako MobileNetSSD.

Ako posledná časť tohto experimentu bolo porovnanie presností detektorov MobileNetSSD a GoogLeNet with YOLO. Toto porovnanie je možné vidieť v grafe 5.5. Ako jasne vyplýva z tohto grafu, tak detektory sú takmer rovnako presné, líšia sa iba o 1,51 %. Z toho vyplýva, že mnou navrhnutý detektor je presnosťou porovnateľný s detektorom MobileNetSSD a zároveň je rýchlejší. Z tohto dôvodu som sa rozhodol v navrhovanom systéme na automatickú anonymizáciu videí použiť navrhnutý detektor GoogLeNet with YOLO.



Obr. 5.4: Porovnanie detektorov tvárí s ich ideálnymi hodnotami prahov. Ako posledný je uvedený graf pre detektor štátnych poznávacích značiek, ktorý je zámerne zobrazený inými farbami, aby bolo jasne vyznačené, že nepatrí k zvyšným štyrom. Tento detektor bude popísaný v nasledujúcich častiach práce.



Obr. 5.5: Porovnanie detektorov tvárí. Oba detektory v tomto porovnaní už boli spustené so svojimi ideálnymi prahovými hodnotami, teda detektor MobileNetSSD s hodnotou 0,8 a detektor GoogLeNet with YOLO s hodnotami 0,7 a 0,7.

5.1.2 Detektory štátnych poznávacích značiek

Narozdiel od detektorov tvárí, ktorých bolo voľne dostupných viacero, tak detektor pre české a slovenské špz som nenašiel žiaden. Voľbe dostupné boli iba detektory schopné detegovať indické, čínske, prípadne americké špz. Avšak tieto značky majú iný rozmer a aj iné farebné kombinácie ako značky české a slovenské, pre ktoré som chcel anonymizačný systém zacieliť. Z toho dôvodu tieto detektory nefungovali správne na požadované špz.

Vytvorenie datasetu a natrénovanie modelu pre detektor

Jediným riešením bolo teda vytvoriť vlastný detektor pre české a slovenské značky. Rozhodol som sa zvoliť detektor YOLO, s ktorým som už pracoval pri detekcii tvárí. Tento detektor mal taktiež podporu aj v knižnici OpenCV, a tiež som mal už skúsenosti s trénovaním modelu pre tento typ detektoru z doby, kedy som sa snažil natrénovať model pre detekciu osôb, no neúspešne.

Na natrénovanie modelu bolo potrebné mať anotovaný dataset s dostatočným množstvom obrázkov áut, ktoré obsahujú české alebo slovenské špz. Takýto dataset by bol následne použitý na trénovanie modelu pre detekčnú sieť.

Dataset som si pre tieto účely vytvoril sám, a to nasledovným spôsobom. Stiahol som 199 fotiek áut z internetu pomocou vyhľadávača Google. Jednalo sa o autá so slovenskými a českými značkami. Takto stiahnuté fotky boli pomerne kvalitné, málo rozmazané a autá na nich boli dobre osvetlené. V reálnych situáciach, ktoré sa môžu vyskytovať vo videu, sú ale autá často rozmazané, zle viditeľné, v zlých svetelných podmienkach, prípadne v noci alebo sú inak obrazovo deformované. Preto som zvyšné obrázky áut získal zo záznamov palubných kamier, ktoré som stiahol z YouTube. Jednalo sa o videá natočené na českých, prípadne slovenských cestách. Na týchto videách som následne pomocou pred-trénovaného detektora YOLO detegoval autá, dodávky a autobusy. Ponechal som iba tie, ktoré mali rozmer aspoň 100×100 pixlov, obsahovali čitateľnú alebo rozpoznateľnú špz a neboli príliš rozmazané alebo inak deformované. Z približne 2 h záznamov som dokázal získať 488 obrázkov áut. Následne som všetky anotoval pomocou nástroja LabelImg[18] a pridal ich k 199 stiahnutým, vopred anonotavaným fotkám. Takto vznikol dataset[19], ktorý obsahoval 687 fotiek/obrázkov áut.

Vytvorený dataset som použil na trénovanie modelu pre detektor YOLO. Vzhľadom k potrebe detegovať iba jednu triedu objektov, som sa rozhodol pre detektor založený na detekčnom jadre YOLO-tiny. Toto jadro obsahuje menej vrstiev, a teda nepotrebuje pre detekciu ani trénovanie také veľké množstvo pamäte ako celé jadro YOLO. Taktiež detekcia s ním je rýchlejšia, no nedosahuje takú presnosť ako plné jadro.

Dataset som náhodne rozdelil na trénovaciu a validačnú časť. Validačná časť bola používaná iba počas trénovania na hrubý odhad jeho správneho priebehu. Preto obsahovala iba 50 obrázkov a trénovacia obsahovala zvyšných 637. Samotné trénovanie som realizoval pomocou frameworku darknet[20], no mierne upraveného pre ľahšiu prácu a rýchlejší tréning. Na dosiahnutie ešte väčšej rýchlosti trénovania som tréning modelu robil pomocou Google Colaboratory[21], ktoré poskytuje výkonné grafické karty ideálne na takéto účely.

[18] https://github.com/tzutalin/labelImg

[19] Dataset je možné nájsť v priečinku *anonymization/experiments/dataset/LICENSE_PLATES/xmokry09-dataset/license_plate_dataset* na odovzdanom disku.

[20] https://github.com/AlexeyAB/darknet

[21] https://colab.research.google.com/

Na natrénovanie bolo potrebných 4 000 iterácií. Pri každej z nich sa model postupne upravoval, až kým hodnota loss funkcie neprestala klesať. Po ukončení trénovania som otestoval všetky modely, ktoré sa ukladali každých 300 iterácií a vybral som z nich ten najlepší. Týmto spôsobom som zabezpečil, že v konečnom detektore nebude použitý model, ktorý by mohol byť pretrénovaný.

Vytvorenie detektoru

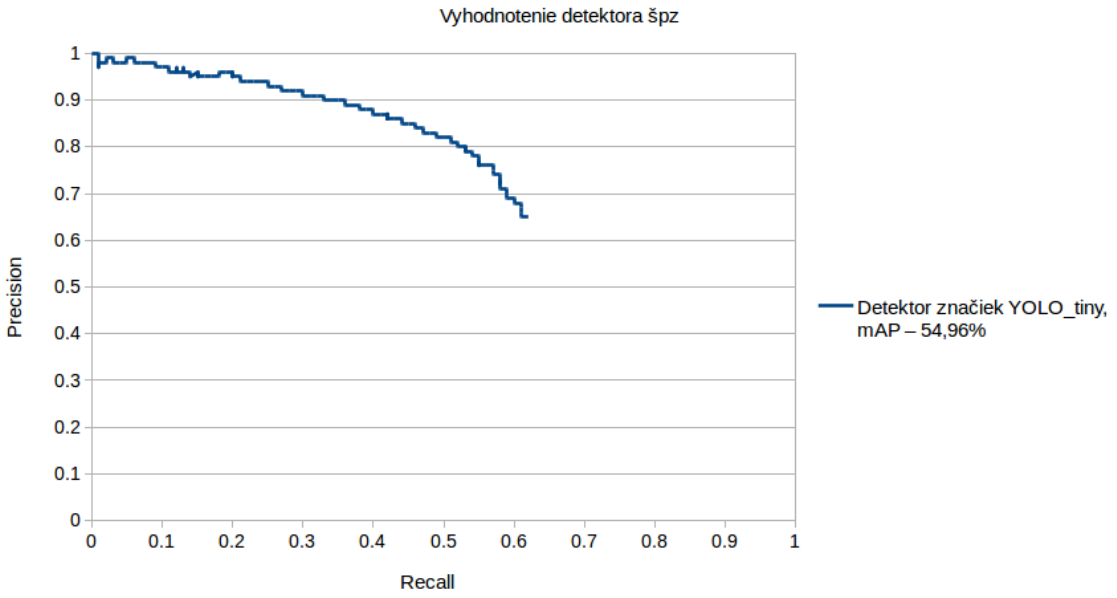
Samotné trénovanie prebiehalo iba na orezaných častiach obrázkov, obsahujúcich iba autá a takmer žiadne pozadie. Z tohto dôvodu som sa rozhodol, že detektor použitý v systéme na anonymizáciu videí bude fungovať s pred-detekciou, ktorá sa osvedčila už pri detekcii tvárí. Podobne ako pri detektore GoogLeNet with YOLO, som aj v tomto prípade vždy na celom snímku videa/obraze urobil pred-detekciu všetkých vozidiel *(autá, dodávky a autobusy)* pomocou pred-trénovaného detektoru YOLO. Následne som v týchto regionóch robil detekciu pomocou natrénovaného modelu na špz, a tým detegoval oblasť, v ktorej sa nachádzala poznávacia značka vozidla.

Vyhodnotenie vytvoreného detektoru

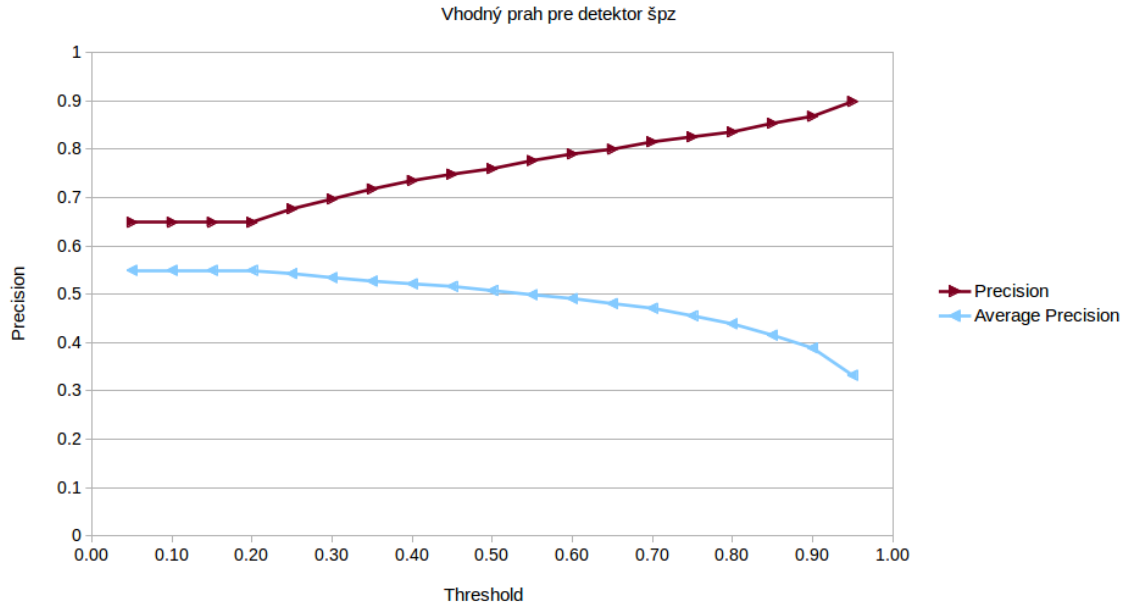
Vytvorený detektor s natrénovaným modelom bolo potrebné ešte vyhodnotiť. Vzhľadom k tomu, že môj vytvorený dataset obsahoval len 50 obrázkov na validáciu, tak vyhodnotenie na ňom by bolo značne nepresné. Z toho dôvodu som sa obrátil na Tomáše Líbala [14], ktorý vo svojej bakalárskej práci vytvoril dataset obsahujúci anotované poznávacie značky na vozidlách. Jeho dataset obsahoval viac ako 2 500 snímkov zo záznamov palubnej kamery v aute. Pre moje účely som použil iba 1 439 z nich. Jednalo sa o obrázky z Brna, Prahy a bližšie neurčených častí Českej Republiky. Taktiež som zmenil formát anotácií z formátu pre YOLO na mnou požadovaný[22]. V grafe 5.6 je možné vidieť vyhodnotenie mnou vytvoreného detektora.

Podobne ako pri detektoroch tvárí bolo aj pri tomto detektore potrebné zistiť optimálny prah istoty detekcie, aby pomer TP / FP bol rozumný. Z grafu 5.7 je možné určiť, že ideálny prah istoty je približne hodnota 0,6. Pri takto nastavenej hodnote prahu je priemerná presnosť *(mAP)* detekcie značiek 49,12 % a konkrétny pomer TP / FP je možné vidieť v grafe 5.4 úplne napravo.

[22] Viac detailov o úprave datasetu sa nachádza v odovzdaných súboroch na priloženom disku, konkrétne v súbore *anonymization/experiments/dataset/LICENSE_PLATES/dataset_xlibal00/readme.txt*.



Obr. 5.6: Vyhodnotenie vytvoreného detektora s nastrénovaným modelom[24]. Vyhodnotenie prebiehalo na datasete od Tomáše Líbala, konkrétne na 1 439 obrázkoch.



Obr. 5.7: Graf na určenie vhodného prahu istoty detekcie pre detektor špz.

[24]Hodnoty precision neklesajú k nízkym hodnotám z dôvodu, že detektor obsahuje aj pred-detekciu a tá mala nastavený prah istoty na 0,7. Z toho dôvodu nedochádzalo k väčšiemu množstvu falošných detekcii.

5.2 Eperimenty a merania s trackermi objektov

Po otestovaní a odmeraní presností detektorov objektov bolo potrebné zistiť ideálny typ trackeru objektov pre anonymizačný systém. Tento experiment som sa rozhodol vyhodnotiť na základe rýchlosti trackerov, teda toho, za aký čas sú schopné jednotlivé trackovacie algoritmy spracovať jedno video. Druhým faktorom hodnotenia bola kvalita sledovania, ktorú pri sledovaní poskytujú. Kvalitu je možné vyhodnotiť dvoma spôsobmi, objektívne a subjektívne. Objektívne hodnotenie by spočívalo v anotácii každého snímku vo videu a následným automatickým porovnaním s výstupom trackovacieho algoritmu. Avšak v mnohých prípadoch bolo aj bez potreby merania jasné, že niektoré sledovacie algoritmy sú horšie[25] ako ostatné. Z toho dôvodu som sa rozhodol, že subjektívne vyhodnotenie kvality bude postačujúce.

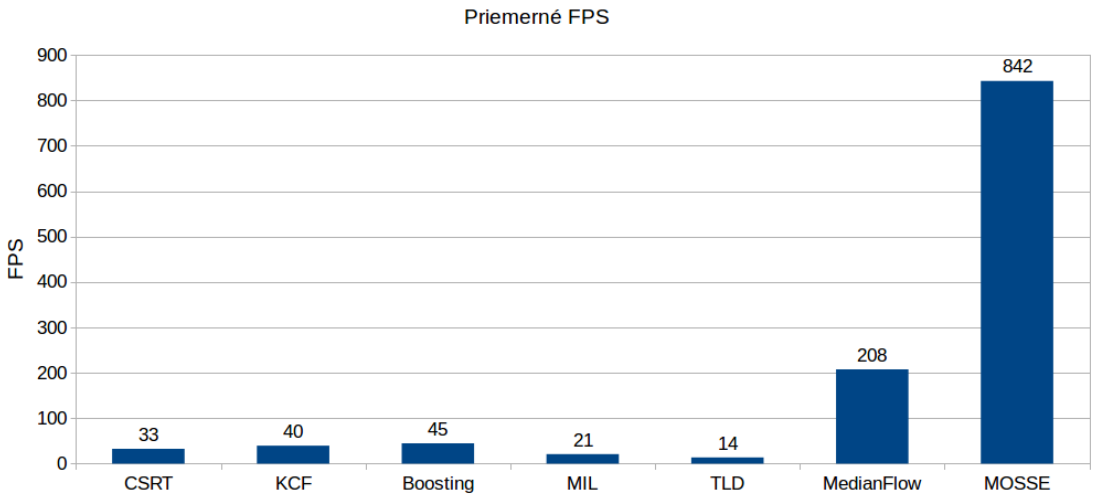
Pre účely testovania som zvolil 5 testovacích videí[26]:

1. `race.mp4` – Ako už názov napovedá jedná o závod na 100 m v šprinte. Sledovaný objekt je Usain Bolt.
   - Rozlíšenie – 1280 × 720
   - Celkový počet snímkov – 340
   - Počet snímkov / s – 30
2. `ping_pong.mp4` – Jedná sa o hru stolného tenisu, kedy sledovaným objektom je jeden z hráčov.
   - Rozlíšenie – 1280 × 720
   - Celkový počet snímkov – 300
   - Počet snímkov / s – 29, 97
3. `los_angeles.mp4` – Dopravná situácia na diaľnici v Los Angeles. Sledovaným objektom je auto, ktoré sa postupne približuje ku kamere a tým sa vo videu zväčšuje.
   - Rozlíšenie – 1280 × 720
   - Celkový počet snímkov – 300
   - Počet snímkov / s – 30
4. `soccer.mp4` – Zápas vo futbale, sledovaný objekt je futbalová lopta, z dôvodu že sa jedná o malý a rýchlo sa pohybujúci sa objekt.
   - Rozlíšenie – 1280 × 720
   - Celkový počet snímkov – 264
   - Počet snímkov / s – 30
5. `vsauce.mp4` – Video z YouTube z kanála Vsauce. Sledovaný objekt je tvár.
   - Rozlíšenie – 1920 × 1080
   - Celkový počet snímkov – 240
   - Počet snímkov / s – 24

[25] Chyba sledovania, výpadok, strata objektu, kmitanie regiónu, náhle zmeny veľkosti regiónu.
[26] Videá sa nachádzajú v priečinku *anonymization/data/experiments_vids/* na odovzdanom disku.

Na každom videu som testoval 7 trackerov, ktoré boli dostupné priamo v knižnici OpenCV. Jednalo sa o trackery, ktoré boli už popísané aj v kapitole 4.3. Na to, aby som zaistil, že každý tracker bude mať rovnaké počiatočné súradnice, som zvolil výber regiónu na 1. snímke každého videa iba raz pre všetky videá. Následne sa tento región použil na inicializáciu každého zo siedmich trackerov. Výstupy pre každú snímku boli ukladané priebežne do poľa a až po skončení všetkých sledovaní sa údaje z týchto polí použili na vizualizáciu vo videu. Každé výstupné video[27] teda obsahovalo 7 za sebou idúcich videí, každé pre jeden tracker. V ľavom hornom rohu týchto videí je aj uvedené, o aký tracker sa jednalo.

Testovanie prebiehalo na CPU Intel 6600K s frekvenciou 4,25 Ghz. Pre každé video bolo výstupom teda 7 časov[28], ktoré zodpovedali jednotlivým trackerom. Keďže každé video malo iný počet snímok, tak som sa rozhodol z dôvodu prehľadnosti, každý dosiahnutý čas previesť na priemerný počet snímok za sekundu. Takto som dostal ku každému videu 7 hodnôt FPS, pre každý tracker jednu. Pre ľahšie zobrazenie výsledkov som ešte urobil priemer pre každý tracker z piatich videí.



Obr. 5.8: Graf zobrazuje priemerné hodnoty FPS, ktoré dosiahli trackery na všetkých testovaných videách. Čím vyššie FPS tým lepší výsledok.

Ako vyplýva z grafu 5.8, tak najlepšie FPS dosiahol tracker MOSSE, no pre potreby anonymizačného systému je oveľa viac podstatná kvalita sledovania ako jeho rýchlosť. Ako som písal na začiatku tejto sekcie, porovnanie kvality som robil subjektívnym spôsobom, a to nasledovne. Kvalitu trackerov som hodnotil na stupnici od 0 – 6, pričom 0 bolo minimum bodov a 6 bolo maximum bodov. 0 bodov získali tie trackery, ktoré nedokázali označený objekt sledovať až do konca videa[29], prípadne sledovanie ani nezačalo. Ostatné trackery boli hodnotené podľa estetickej a kvalitatívnej stránky sledovania. Hodnotil som konkrétne mieru mihotania sa sledovaného regiónu, schopnosť trackeru sledovať celý objekt, ak sa

[27] Výstupné videá je možné nájsť v priečinku *anonymization/experiments/results/trackers_exp/* na odovzdanom disku.

[28] Presné výsledky je možné nájsť v súbore *anonymization/experiments/results/trackers_exp/results.txt* na odovzdanom disku.

[29] Pri videu *soccer.mp4*, nebol ani jeden tracker schopný sledovať loptu do konca videa, tak som ohodnotil tie dva, ktoré ako jediné vôbec sledovanie začali.

tento v priebehu času vo videu zväčšuje, plynulosť pohybu regiónu, schopnosť sledovať tvár a celkový dojem zo sledovania. V nasledujúcej tabuľke 5.1 je možné vidieť výsledky tohto hodnotenia.

Tracker / Video	CSRT	KCF	Boosting	MIL	TLD	Median	MOSSE
race.mp4	5	6	4	0	0	0	0
ping_pong.mp4	4	5	0	3	0	0	5
los_angeles.mp4	6	3	1	2	0	5	4
soccer.mp4	6	0	0	0	0	5	0
vsauce.mp4	3	6	1	2	0	4	5
Celkovo	24	21	6	7	0	14	14

Tabuľka 5.1: Vyhodnotenie kvality sledovania objektov jednotlivými trackermi. Bola použitá subjektívna metrika, ktorá je popísaná v sekcii 5.2

Najlepšie skóre dosiahol tracker CSRT, ktorý podľa očakávania dokázal sledovať menšie aj väčšie objekty a dokázal sa prispôsobiť aj zmenám veľkostí objektov, ale pri sledovaní tváre pri ňom dochádzalo k miernemu mihotaniu, čo by mohlo byť v konečnom systéme rušivé pre diváka. Z tohto dôvodu som sa rozhodol pre sledovanie tvárí využiť tracker KCF, ktorý dosiahol taktiež dobré kvalitatívne výsledky a navyše dokázal sledovať tvár veľmi plynulým pohybom. Tracker CSRT som zvolil pre sledovanie špz, keďže dokáže dobre sledovať aj menšie objekty, pri ktorých sa tracker KCF ukázal ako nevyhovujúci. Najrýchlejší tracker MOSSE síce nebol kvalitou najhorší zo všetkých testovaných, ale pri rýchlo sa pohybujúcich objektoch často zlyhal, a teda aj napriek jeho výhode v rýchlosti som sa rozhodol ho nepoužívať.

5.3 Návrh a implementácia anonymizačného systému

Systém by mal dokázať automaticky anonymizovať požadované objekty vo videu, ktoré dostane na vstupe. Keďže vstupné video môže byť akokoľvek dlhé, môže obsahovať rôzne objekty a tiež sa môže skladať z viacerých scén, je potrebné video najprv analyzovať. Zvolil som prístup analýzy videa na základe scén. Jedna scéna je chápaná ako jeden súvislý záber, v ktorom môže byť premenlivý počet objektov, no tieto objekty do scény musia postupne prísť a nemali by sa tam náhle zjaviť. Rozdelením videa na scény sa teda zabezpečí, že objekty, ktoré sa v scéne nachádzajú tam s najväčšou pravdepodobnosťou budú počas celej dĺžky trvania scény. Vďaka tomuto faktu je možné uplatniť detekcie objektov iba na určitých snímkach. Prvá detekcia samozrejme prebehne na prvej snímke a každá ďalšia za X snímok neskôr. Posledná detekcia prebehne na poslednej snímke scény. Snímky, na ktorých prebehla detekcia sa nazývajú kľúčové snímky. Medzi týmito snímkami je následne možné aplikovať sledovací algoritmus, aby sa detekcia nemusela robiť na každej snímke.

Základný algoritmus by teda vytvoril trackery pre všetky objekty z prvej kľúčovej snímky a sledoval by ich až do konca scény. Takéto riešenie ale nepočíta s tým, že sa v scéne môžu objaviť aj nové objekty, prípadne, že trackery môžu sledované objekty stratiť. Preto som algoritmus navrhol tak, aby na každej kľúčovej snímke prebehla kontrola, či sa v nej nenachádzajú nové objekty oproti tým, ktoré sledujú trackery. Ak sa nájdu nové objekty, tak aj pre tieto sa vytvoria nové trackery a týmto spôsobom sa postupeje až do konca scény.

Nové objekty sa však v scéne mohli objaviť aj skôr ako boli nájdené detektorom. Príklad: Na prvej snímke sa detekovala jedna tvár, vytvoril sa pre ňu tracker a začala sa sledovať.

Na snímke 25 sa objaví nová tvár, avšak detektor ju v ideálnom prípade nájde až na snímke 30. Algoritmus nebude mať žiadnu informáciu o tom, že tvár sa nachádzala vo videu už na snímkach 25 – 29, a teda na týchto piatych snímkach nebude anonymizovaná.

Tento problém som vyriešil spätným sledovaním nových objektov. Vždy, ak sa nájde v scéne nový objekt oproti objektom, ktoré sa aktívne sledujú, tak sa tento nový objekt sleduje spätne až do momentu jeho vzniku. Tým sa minimalizuje problém určenia presného momentu vstupu nových objektov do scény.

Môže tiež nastať situácia, kedy dôjde k výpadku trackeru určitého objektu. Objekt bude znovu detekovaný napríklad až za 5 kľúčových snímok. Jeho spätné sledovanie však nedokáže objekt sledovať až na miesto výpadku *(vypadne spätné sledovanie, objekt sa medzitým zmenil...)*. Vznikol by teda priestor, kedy by objekt opäť nebol anonymizovaný. Tento problém nie je možné úplne vyriešiť, keďže sa nedá predpovedať, kedy dôjde k výpadku a kedy sa objekt opäť detekuje. V navrhnutom algoritme som však tento problém minimalizoval tak, že ak dôjde k výpadku trackeru, tak jeho posledná pozícia sa použije ešte na ďalších 10 snímok. Takéto opatrenie by malo aspoň mierne minimalizovať dopady na výpadky anonymizácie a zároveň 10 snímok vo videu je krátky čas na to, aby to príliš rušilo diváka aj v prípade, že objekt sa vo videu už nenachádza.

```
scenes = detect_scenes(input_video)

# Object detection for each key frame in scene
for scene in scenes:
    frames_in_scene = read_all_frames(scene)
    for i in range(scene_length):
        if i is key_frame:
            detected_objects += detect_objects(frames_in_scene[i])

    # Start tracking first objects
    create_trackers(detected_objects[first_detected_object])
    for i in range(1, scene_length):
        update_trackers(frames_in_scene[i])
        if trackers_update is OK:
            all_objects[i] += new_object_posistion
        else:
            use_last_ROI_+10_frames()
            delete_bad_trackers()

        # On every next key frame check for new objects
        if i is key_frame:
            new_objects = check_for_new_objects()
            if new_object:
                create_trackers(new_objects)
                all_objects += backtrack(new_objects)
        # Append all regions of detected and tracked objects
        all_objects += detected_objects

# Use regions for anonymization and save to output video
all_frames = get_all_frames(input_video)
    for frame in all_frames:
        blured = blur_frame(frame, all_objects[frame_index])
        write_frame(output_video, blured)
```

Výpis 5.1: Algoritmus použitý v systéme na anonymizáciu videí napísaný v pseudokóde podobnému jazyku python.

Tento algoritmus je možné vidieť v jeho zjednodušenej podobe na výpise 5.1 ako pseudokód. V tomto pseudokóde sú mnohé časti zjednodušené alebo úplne vynechané, aby bol lepšie čitateľný. Konkrétne hodnoty, ktoré som použil pri implementácii som zistil pri samotnom vývoji a priebežnom testovaní systému na dátovej sade videí. Kľúčové snímky sa nachádzajú v rozostupe každých 10 snímok, taktiež pri výpadku trackeru sa jeho posledná pozícia vloží na ďalších 10 snímok. Všetky časti algoritmu sú pôvodné a navrhol som ich sám. Výnimkou je pár pomocných funkcií, ktorými som sa inšpiroval z niektorých internetových zdrojov. V zdrojovom kóde sú však všetky takéto prevzaté alebo inšpirované funkcie jasne označené a majú pri sebe uvedený aj zdroj.

Výnimkou je tiež aj nástroj na detekciu scén[30], ktorý je potrebné stiahnuť ako hotovú knižnicu pomocou príkazu pip. Okrem tejto knižnice a knižnice OpenCV nástroj nepotrebuje žiadne iné externé knižnice.

5.3.1 Užívateľské rozhranie

Aj napriek maximálnej snahe o čo najlepší výkon anonymizačného systému nie je možné, aby mal nástroj úspešnosť 100 %. Vždy môže dôjsť k nejakým nepredvídateľným situáciám, ktoré sa môžu vo vstupných videách vyskytnúť. Takýmto situáciám som sa snažil predísť tým, že som systém testoval na dátovej sade videí, v ktorej sa nachádzajú videá obsahujúce rozličné situácie.

Z dôvodu, že nie je možné zaručiť bezchybnosť anonymizácie, ktorá je v tomto prípade potrebná, rozhodol som sa k systému pridať aj jednoduché užívateľské rozhranie pre prípadné opravy nedostatkov v automatickej anonymizácii. Toto rozhranie som realizoval pomocou knižnice OpenCV a to tak, že užívateľ sa môže pohybovať vpred a vzad vo videu a hľadať prípadné chyby. Ak nejakú chybu nájde, tak jednoducho tento ne-anonymizovaný objekt označí a tracker ho začne sledovať až do momentu, kým buď tracker tento objekt nestratí *(obnoví sa anonymizácia)* alebo neprejde 50 snímkov. Po označení nedostatku užívateľ jasne vidí, aká časť bola označená a na ktorom snímku. Po takomto označení všetkých nedostatkov vo videu sa stlačením príslušnej klávesy video uloží. Ukážku užívateľského rozhrania je možné vidieť na nasledujúcom obrázku.

Obr. 5.9: Ukážka užívateľského rozhrania pre anonymizačný systém. Ovládacie prvky sú zobrazené na spodnom okraji obrazu a ovládanie sa realizuje cez klávesy, prípadne myšou pri výbere objektu. Na obrázku je vidieť aj ako vyzerá označený objekt pre anonymizáciu.

[30] https://pyscenedetect.readthedocs.io/en/latest/

5.3.2 Prístupy k anonymizácii

Ako už bolo spomenuté v kapitole 3.1.3, tak na anonymizáciu obrazu, teda aj videa, je možné použiť rôzne prístupy. Či už sa jedná o rozmazanie objektov alebo nahradenie informácie inou hodnotou, prípadne iným prístupom. Všetky tieto metódy je možné aplikovať aj prostredníctvom knižnice OpenCV, pomocou ktorej je systém implementovaný. V konečnom riešení som ale ponechal iba možnosť rozmazania, keďže mi zo všetkých dostupných metód prišla najmenej rušivá a zároveň poskytovala dobrú mieru anonymizácie.

Rozmazanie je realizované pomocou elipsovej masky vpísanej do regiónu, ktorý vyznačuje objekt. Táto elipsová maska je ešte pred použitím rozmazaná Gaussovským šumom, aby sa zjemnili jej okraje a pôsobila tak vo videu ešte menej rušivým dojmom. Následne sa cez masku pomocou Alpha blendingu spojí pôvodný rozmazaný obraz z regiónu s elipsovou maskou. Výsledok takejto anonymizácie je možné vidieť na nasledujúcom obrázku.

Obr. 5.10: Ukážka anonymizácie pomocou vytvoreného systému. Na obrázku je možné vidieť masku v tvare elipsy aplikovanú na oblasť tváre.

5.3.3 Podpora videí so zvukom

Počas vývoja systému som ho testoval na mnohých videách. Veľa z nich bolo založených práve na dialógoch a po spracovaní videa cez môj systém boli výstupné videá bez zvuku. Takéto videá teda pôsobili pomerne zvláštne, najmä ak sa na nich nachádzali ľudia, ktorí zjavne konverzovali, ale nebolo nič počuť.

Z tohto dôvodu som do systému pridal aj možnosť anonymizovať videá a pri tom zachovať pôvodnú zvukovú stopu. Na toto som využil knižnicu `ffmpeg`[31], ktorá sa dá jednoducho nainštalovať na každý PC. Systém je samozrejme plne funkčný aj bez tejto knižnice, ale ak užívateľ chce mať anonymizované video aj so zvukom, tak ju musí mať v systéme nainštalovanú. Hlas osôb vo videu nie je nijako zmenený, teda sa neanonymizuje.

5.3.4 Ovládanie výsledného programu

Systém je implementovaný ako konzolová aplikácia, teda okrem jednoduchého užívateľského rozhrania, ktoré slúži iba na dodatočné úpravy, nemá žiadne iné ovládacie prvky. Ovládanie sa deje iba pomocou prepínačov zadávaných do príkazového riadku. Ovládanie aplikácie:

[31] https://ffmpeg.org/

Použitie: `anonym.py [-h] [-d] [-i] [-o] [--audio]`

Prepínač	Význam
-h / --help	Výpis nápovedy a ukončenie programu
-d / --detection	Výber typu objektu, ktorý sa má anonymizovať:
	1 – Tváre
	2 – Poznávacie značky
	3 – Užívateľské rozhranie
-i / --input	Názov/cesta k vstupnému videu
-o / --output	Názov/cesta výstupného videa
--audio	Pre zachovanie pôvodnej zvukovej stopy[32]

5.4 Testovacia dátová sada

Vzhľadom k tomu, že tento systém je automatický, tak je potrebné ho testovať a overovať tak jeho správnu funkčnosť. Na tento účel slúži vytvorená dátová sada, ktorá pozostáva z rôznych videí. Táto sada pozostáva z 19 videí, ktorých obsah sa líši tak, aby bolo možné sledovať či systém pracuje správne. Videá sa líšia či už svetelnými podmienkami, počtom ľudí v nich, farbou ich pleti, kvalitou videa, prípadne ide o záznamy z dynamickej scény, kedy sa pohybujú ako ľudia tak aj samotná kamera. Tieto aspekty majú za úlohu otestovať reakcie systému na rôzne zmeny v obraze, ktoré môžu nastať. Videá som sa snažil vyberať podľa požiadaviek, ktoré môžu mať budúci užívatelia systému.

Keďže dátová sada je tvorená primárne videami, tak v každom z takýchto videí sa nachádzajú miesta, ktoré sú určitým spôsobom problémové. Teda objekty na nich nie sú správne zdetekované alebo anonymizácia je v danom úseku videa príliš rušivá, či nedostatočná. Z tohto dôvodu som sa rozhodol tieto problémové úseky z videí vybrať, a to buď ako kratšie výseky videí alebo ako jednotlivé snímky. Na týchto krátkych úsekoch alebo snímkach som následne testoval systém počas jeho vývoja. Taktiež sa tieto krátke úseky nachádzajú aj ako samostatné videá v dátovej sade, a to z dôvodu zredukovania pamäťových nárokov pri odovzdávaní. Niektoré videá mali totiž v pôvodnej dĺžke viac ako 10 minút a viac ako 300 MB. Po vybraní krátkych úsekov (30 s – 2 min) sa táto veľkosť značne zmenšila a zároveň je stále zachovaná schopnosť videa otestovať vyvíjaný systém. Konkrétne videá s ich popisom a zdrojmi je možné nájsť v prílohe A v tejto práci.

[32] Potrebné mať nainštalovanú knižnicu *ffmpeg*

Kapitola 6

Zhodnotenie výsledkov automatickej anonymizácie

Poslednou úlohou bolo vyhodnotenie navrhnutého a implementovaného systému. Vyhodnotenie detekčných a sledovacích algoritmov bolo popísané už v sekciách 5.1 a 5.2. V tejto kapitole bude popísané vyhodnotenie systému ako celku.

Systém som sa rozhodol vyhodnotiť subjektívnym spôsobom, teda mnou a následne s pomocou iných ľudí formou dotazníka. Ja sám som systém vyhodnocoval už počas jeho implementácie, a to na sade videí. Prípadné chyby a nedostatky som opravoval vylepšovaním systému, aby som jeho chybovosť obmedzil na minimum. Keď som nadobudol pocit, že systém je dostatočne kvalitný, tak som pristúpil k hodnoteniu s pomocou mnou dotazovaných osôb.

Na toto som si zvolil dve ukážkové videá, ktoré som najprv automaticky anonymizoval, následne som ich ešte upravil pomocou užívateľského rozhrania a nahral ich na YouTube.

6.1 Vyhodnotenie anonymizácie poznávacích značiek vozidiel

Prvé video[1] obsahovalo záznam z palubnej kamery z českých ciest, ktorý som stiahol z YouTube. K tomuto videu som vytvoril dotazník, v ktorom som sa pýtal nasledujúce otázky[2]:

- **1.** Čo sa vo videu deje? – táto otázka mala za úlohu zistiť, či sú dotazované osoby schopné z videa poznať jeho hlavný účel. Teda, či anonymizácia nepoškodila informácie vo videu na toľko, že by stratilo pôvodný význam.
- **2.** Nachádzalo sa vo videu auto českej pošty? – nejednalo sa o test pozornosti, ale išlo o zistenie, či anonymizácia nepoškodila celé autá, prípadne ich inak neznehodnotila. Auto pošty bolo pomerne veľké a dobre viditeľné, takže ak by ho dotazované osoby nedokázali rozpoznať, tak by to mohlo znamenať, že systém nepracuje správne.
- **3.** Nachádzali sa vo videu reklamy? *(pútače, plagáty, billboardy)* – reklamy a pútače sú tvarovo často podobné poznávacím značkám, tiež obsahujú text na nejakom pozadí. Cieľom otázky teda bolo zistiť, či si detektor poznávacích značiek nezamieňa značky s reklamami popri ceste.

[1] https://www.youtube.com/watch?v=qFziPeQGec0, tiež je možné nájsť v priečinku *anonymization/data-/dataset_vids_blured/* na odovzdanom disku

[2] K týmto otázkam bolo vždy na výber niekoľko odpovedí a dotazované osoby mali vybrať tú, ktorá podľa nich najviac zodpovedala na danú otázku.

- **4.** Dokázali by ste identifikovať vo videu akékoľvek vozidlo podľa výrobcu? Ak áno, napíšte aspoň jednu značku výrobcu vozidiel *(do sekcie Iné)*, ktoré sa vo videu nachádzalo. – rovnako ako pri rozpoznaní auta Českej pošty išlo o rozpoznanie áut. Tento krát však už všetkých aút na základe ich výrobcov.

- **5.** Dokázali by ste identifikovať akékoľvek vozidlo vo videu podľa jeho špz? Ak áno, napíšte prosím celú špz *(do sekcie Iné)* – posledná otázka sa zameriavala na fakt, či sú dotazované osoby schopné rozpoznať akúkoľvek poznávaciu značku vo videu. Išlo o overenie, že systém pracuje správne.

Odpovede na tieto otázky je možné nájsť v prílohe B v tejto práci. Ako je možné vidieť z odpovedí dotazovaných osôb, tak väčšina z nich dokázala správne zodpovedať na všetky otázky a dokázali správne rozpoznať autá, typ situácie vo videu, dokonca aj výrobcov aút a reklamné plochy. Nedokázali však rozpoznať žiadnu registračnú značku vozidla. Z toho teda vyplýva, že anonymizačný systém funguje správne pri anonymizácií špz.

6.2 Vyhodnotenie anonymizácie tvárí ľudí

Druhé video[3] znázorňovalo kúzelné triky na verejnosti, konkrétne v meste New York. Toto video bolo pomerne dynamické, kedy sa v ňom pohybovali ako hlavní účastníci tak aj kamera. Takéto náročnejšie video som vybral zámerne, aby som otestoval systém pre náročnejšie situácie.

Opäť som pri vyhodnocovaní postupoval podobne ako pri hodnotení anonymizácie špz. Vytvoril som dotazník, ktorý obsahoval nasledujúce otázky:

- **1.** Čo sa vo videu deje? – rovnako ako pri predchádzajúcom dotazníku bolo potrebné zistiť, či anonymizácia nepoškodila alebo nezmenila pôvodný význam videa.

- **2.** Mali osoby na videu na sebe nejaké šperky? *(retiazky, náhrdelníky, prstene)* – otázka mala zistiť, či anonymizácia nepoškodila niektoré časti obrazu, na ktorých sa nenachádzali tváre. Prípadne, či anonymizácia nebola natoľko rušivá, že by si dotazované osoby nevšimli žiadne detaily na hlavných aktéroch.

- **3.** Mal niekto z osôb vo videu na oblečení nápis? – rovnako ako predchádzajúca otázka, mala aj táto za úlohu zistiť, či nedošlo k poškodeniu častí obrazu, kde sa nachádzali nápisy na oblečení. A tiež, či anonymizácia nebola príliš rušivá.

- **4.** Aký predmet bol použitý na prípadné triky vo videu? – jednalo sa o malý predmet, konkrétne balzám na pery. Tento predmet bol viac krát prikladaný k ústam kúzelníka, a teda bol tiež občas anonymizovaný. Bolo potrebné zistiť, či aj napriek tomuto faktu dokázali dotazované osoby určiť, o aký predmet sa jedná.

- **5.** Dokázali by ste niektorému z hlavných aktérov vo videu opísať tvárové črty? *(farba očí, tvar nosa, tvar pier, obočie, mimiku tváre)*. Ak áno, tak uveďte komu (do možnosti Iné), a tiež akú časť tváre viete opísať. – posledná otázka mala za účel zistiť, či bola anonymizácia dostatočná a účinná.

[3] https://www.youtube.com/watch?v=6SYhbBfm1hI, tiež je možné nájsť v priečinku *anonymization/data-/dataset_vids_blured/* na odovzdanom disku

Odpovede na tieto otázky je možné nájsť v prílohe B. Z otázok vyplýva, že anonymizácia nebola dostatočná. Napriek tomu, že nebola rušivá a väčšina dotazovaných osôb dokázala zodpovedať otázky správne, tak takmer polovica z nich dokázala určiť tvárové črty minimálne dvom hlavným aktérom vo videu. Toto bolo spôsobené dvoma faktormi.

Prvým bolo nedostatočné rozmazanie tvárí osôb, ktoré bolo spôsobené nízko nastavenou hodnotou pri použití Guassovského rozmazania. Tento problém som následne na základe tohto dotazníku opravil.

Druhým problémom bola značná dynamika videa a nedostatočné opravenie chýb v užívateľskom rozhraní. Domnieval som sa totiž, že ak na jednej snímke bude viditeľná časť čela osoby, tak si to počas prehrávania videa dotazované osoby nevšimnú. Viac ako polovica dotazovaných osôb, ktorí odpovedali na otázky, video sledovala až príliš pozorne, resp. v spomalenom režime, prípadne za neustáleho zastavovania videa a sústredila sa na chyby pri anonymizácii. Toto samozrejme nebola ich chyba a mal som predpokladať takúto možnosť. Z tohto dôvodu som video opravil[4], venoval som viac času a pozornosti aj menším nedostatkom a opäť som vytvoril rovnaký dotazník, no už s opraveným videom. Tento dotazník som však nechal vyplniť inú skupinu dotazovaných osôb, aby nedošlo k skresleniu výsledkov.

Odpovede z nového dotazníka je možné nájsť opäť v prílohe B. Ako sa na odpovediach dotazovaných osôb ukázalo, tak zvýšenie miery rozostrenia a dôkladnejšia úprava anonymizácie pomocou užívateľského prostredia zabezpečili, že anonymizácia bola v tomto prípade na lepšej úrovni. Tento krát sa takmer nikomu nepodarilo rozpoznať žiadne zásadné tvárové črty, na základe ktorých by bolo možné identifikovať hlavných aktérov vo videu a zároveň stále väčšina dokázala odpovedať správne na otázky ohľadom videa.

Na týchto dvoch hodnoteniach (sekcie 6.2 a 6.1) s pomocou dotazovaných osôb bolo jasne preukázané, že anonymizačný systém je funkčný a dokáže do veľkej miery automaticky anonymizovať dva druhy objektov vo videách. V prípade špz je výsledok o niečo lepší, keďže sa jedná o menšie plochy vo videu a výsledok tak nepôsobí príliš rušivým dojmom.

Anonymizácia tvárí je taktiež pomerne úspešná, ale pri dynamických scénach – ako na testovacom videu, sa ukázalo, že vytvára falošné detekcie a občas môže pôsobiť rušivým dojmom. Pri menej náročných videách sa tieto problémy objavujú zriedkavejšie.

6.3 Možnosti pokračovania v projekte

Systém umožňuje automaticky anonymizovať 2 typy objektov. Najlogickejšou možnosťou rozšírenia by teda bolo pridanie ešte viacerých typov objektov, ktoré by systém dokázal automaticky rozpoznať a následne vhodne anonymizovať. Samozrejme nie všetky objekty sú vhodné na anonymizáciu, preto spomeniem iba tie, pri ktorých by bola anonymizácia vhodná.

Osoby by boli asi najvhodnejším typom objektu na ďalšie pokračovanie v projekte. Dokázať a vhodne anonymizovať celé postavy osôb by ale vyžadovalo zvoliť iný prístup ako rozmazanie, keďže rozmazanie by v tomto prípade pôsobilo príliš rušivo. Osoby by sa dali anonymizovať napríklad zmenou farby ich oblečenia, rozmazaním ich pohybu alebo úplným vymaskovaním pomocou pozadia.

Taktiež by systém mohol anonymizovať aj tetovania, ktoré sa tiež v niektorých prípadoch dajú použiť na identifikáciu osôb. V prípade tetovaní by stačilo zvoliť anonymizovanie aj

[4] https://youtu.be/wzY3gq5vWcE, tiež je možné nájsť v priečinku *anonymization/data/dataset_vids_blured/*

formou rozmazania, avšak problém by nastal pri detekcii. Tetovania sú totiž rôznych farieb a tvarov a bolo by teda obtiažne ich vo videu presne detekovať.

Systém by sa nemusel zameriavať iba na anonymizáciu osôb a identifikačných prvkov. Mohol by napríklad poskytovať aj anonymizáciu rôznych značiek výrobcov, značky nápojov, jedál, logá ich výrobcov alebo konkrétne tvary charakteristické pre isté výrobky. Takáto anonymizácia by mohla byť vhodná napríklad pri vysielaní videa v televízii z dôvodu nechcenej propagácie výrobcu.

Z funkčných častí, ktoré systém už obsahuje by sa dala vylepšiť napríklad detekcia tvárí, aby sa pri menej kvalitných záznamoch alebo dynamických scénach objavovalo menšie množstvo falošných detekcií, ktoré následne pôsobia rušivo pre diváka. Taktiež by bolo možné doplniť podporu anonymizácie iba pre niektoré tváre namiesto všetkých. Takýto prístup by mohol byť žiadaný napríklad, ak iba časť osôb vo videu neudelila súhlas na zverejnenie videa. Toto by sa následne dalo kombinovať s vylepšeným užívateľským rozhraním, kde by si užívateľ mohol zvoliť, ktoré tváre chce anonymizovať. Pre vylepšenie detekcie tvárí by bolo taktiež možné natrénovať vlastný model pre konkrétny detektor objektov a zvýšiť tak presnosť a aj mieru detekcie tvárí.

Anonymizácia poznávacích značiek by sa tiež dala vylepšiť, napríklad rozšírením dátovej sady použitej pri trénovaní a následným pretrénovaním modelu na tejto väčšej sade. Rovnako by bolo možné miesto modelu založenom na YOLO-tiny použiť úplný model YOLO, a tým tiež zvýšiť presnosť a mieru detekcií, prípadne zvážiť použitie úplne iného vhodného detektora.

Poslednou časťou, ktorú by bolo možné vylepšovať je užívateľské rozhranie. Aktuálne rozhranie je naozaj veľmi jednoduché a základné. Pôvodne ho systém ani nemal obsahovať, ale občasné chyby a výpadky ma prinútili ho do systému pridať. Toto rozhranie by sa mohlo realizovať napríklad formou plnohodnotného grafického užívateľského rozhrania a celý systém by teda mohol mať podobu počítačovej aplikácie.

Kapitola 7

Záver

V tejto diplomovej práci boli uvedené prípady, kedy môže byť prínosné použitie systému pre automatickú anonymizáciu videí. Taktiež tu boli rozobrané a popísané technológie, ktoré sú potrebné pre vytvorenie takéhoto systému, či sa jednalo o rôzne druhy detektorov objektov alebo trackerov objektov. Bolo podrobne vysvetlené ako tieto technológie fungujú a aké princípy pri svojej činnosti využívajú. Pre uvedenie čitateľa do problematiky boli uvedené aj rôzne metódy anonymizácie pre rôzne typy dát a boli uvedené aj konkrétne príklady takýchto anonymizácii.

Na základe teoretických poznatkov, ktoré autor počas vypracovania práce nadobudol, bol navrhnutý systém pre automatickú anonymizáciu videí. Boli vyhodnotené a otestované najvhodnejšie metódy na detekciu typov objektov, ktoré systém umožňuje anonymizovať. V prípade detekcie poznávacích značiek, kedy nebol k dispozícii žiadny voľne-dostupný detektor českých a slovenských značiek, bol tento vytvorený. Pre jeho vytvorenie bola zhotovená a následne vhodne anotovaná trénovacia dátová sada, na ktorej bol následne natrénovaný model pre tento detektor.

Po vyhodnotení a otestovaní všetkých potrebných súčastí bol implementovaný automatický anonymizačný systém. Táto implementácia prebiehala postupne za neustáleho testovania systému. Počas tohto procesu boli opravované odpozorované nedostatky a chyby. Pre účel testovania systému počas implementácie bola vytvorená sada videí, ktorá obsahuje videá rôznych situácií, ktoré by mal systém zvládať. Na základe testovania vyvíjaného systému sa autor tejto práce rozhodol do systému pridať aj jednoduché užívateľské rozhranie, ktoré vhodne dopĺňa funkčnosť systému a robí tak z neho plnohodnotný nástroj na anonymizáciu videí.

V poslednej časti práce boli prezentované výsledky z vyhodnotenia vytvoreného systému, ktoré boli nadobudnuté na základe užívateľských testov. Aj na základe týchto testov boli odhalené určité nedostatky systému. Tie boli následne opravené a znovu otestované. V poslednej časti práce boli tiež navrhnuté viaceré možnosti pokračovania v projekte.

Literatúra

[1] Babenko, B.; Yang, M.-H.; Belongie, S.: *Visual Tracking with Online Multiple Instance Learning*.

[2] Banitalebi-Dehkordi, A.; Pourazad, M. T.; Nasiopoulos, P.: *The Effect of Frame Rate on 3D Video Quality and Bitrate*.

[3] Birnstill, P.; Ren, D.; Beyerer, J.: A User Study on Anonymization Techniques for Smart Video Surveillance. 08 2015, doi:10.1109/AVSS.2015.7301805.

[4] Cernocký, J.: *Zpracování řečových signálů: 4. LPC*. [Online; navštívené 07.01.2019]. URL https://www.fit.vutbr.cz/study/courses/ZRE/public/pred/04_lpc/04_lpc.pdf

[5] David S. Bolme, J. R. B.; Draper, B. A.; Lui, Y. M.: *Visual Object Tracking using Adaptive Correlation Filters*.

[6] Estephan, H.; Sawyer, S.; Wanninger, D.: *Real-Time Speech Pitch Shifting on an FPGA*. [Online; navštívené 07.01.2019].
URL http://www56.homepage.villanova.edu/scott.sawyer/fpga/II_frequency_shifting.htm

[7] Girshick, R. B.: Fast R-CNN. *CoRR*, ročník abs/1504.08083, 2015, 1504.08083.

[8] Girshick, R. B.; Donahue, J.; Darrell, T.; aj.: Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, ročník abs/1311.2524, 2013, 1311.2524.

[9] Grabner, H.; Grabner, M.; Bischof, H.: *Real-Time Tracking via On-line Boosting*.

[10] João F. Henriques, R. C.; Martins, P.; Batista, J.: High-Speed Tracking with Kernelized Correlation Filters.

[11] Kalal, Z.; Mikolajczyk, K.; Matas, J.: Forward-Backward Error: Automatic Detection of Tracking Failures. 2010.

[12] Kalal, Z.; Mikolajczyk, K.; Matas, J.: Tracking-Learning-Detection. 2010.

[13] Kumar, M.: *Strategy for Design and Building Multimedia Data Type. International Journal of Computer Applications*, 2013.

[14] Líbal, T.: *Detekce registrační značky vozidla ve videu*. Bakalářská práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2019.

[15] Lukežič, A.; Vojíř, T.; Čehovin Zajc, L.; aj.: Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. *International Journal of Computer Vision*, 2018.

[16] López, L. S.: *Local Binary Patterns applied to Face Detection and Recognition*. [Online; navštívené 14.01.2019].
URL https://upcommons.upc.edu/bitstream/handle/2099.1/10772/PFC_LauraSanchez_(LBP_applied_to_FaceDetection&Recognition).pdf

[17] Raghunathan, B.: *The Complete Book of Data Anonymization: From Planning to Implementation*. Auerbach Publications, 2013, ISBN 978-1439877302.

[18] Redmon, J.; Farhadi, A.: YOLOv3: An Incremental Improvement. *CoRR*, ročník abs/1804.02767, 2018, 1804.02767.

[19] Rek, P.: *Knihovna pro návrh konvolučních neuronových sítí*. Diplomová práca, Vysoké učení technické v Brně, Fakulta informačních technologií, 2018.
URL http://www.fit.vutbr.cz/study/DP/DP.php?id=3868

[20] S R, B.; S. Karthikeyan, D.: A survey on moving object tracking using image processing. 02 2017, doi:10.1109/ISCO.2017.7856037.

[21] Singapore, P. D. P. C.: *GUIDE TO BASIC DATA ANONYMISATION TECHNIQUES*. [Online; navštívené 07.12.2018].
URL
https://iapp.org/media/pdf/resource_center/Guide_to_Anonymisation.pdf

[22] Viola, P.; Jones, M.: Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, ročník 1, Dec 2001, ISSN 1063-6919, s. I–I, doi:10.1109/CVPR.2001.990517.

Príloha A

Datová sada na testovanie

V tejto prílohe je uvedený zoznam testovacích videí, ktoré boli použité pri vývoji a následnom testovaní systému pre anonymizáciu videí.

Zoznam videí

Nasledujúci zoznam obsahuje videá, ktoré boli použité pri vývoji a testovaní. Väčšina z videí bola prevzatá z YouTube. Pri každom z videí, ktoré boli prevzaté z YouTube je aj odkaz na dané video. Videá boli vyberané tak, aby čo najlepšie pokryli všetky testované nároky systému. V zátvorke pri každom videu je aj jeho názov, pod akým sa nachádza v datovej sade, konkrétne v priečinku *anonymization/data/dataset_vids* na odovzdanom disku.

1. **The Graham Norton Show** *(show.mp4)* – krátky ústrižok z jedného dielu tejto talkshow. Toto video bolo zvolené najmä kvôli výraznej gestikulácií všetkých zúčastnených. Výrazná gestikulácia a mimika tváre môže zmeniť črty tváre v niektorých prípadoch až natoľko, že systém tvár nesprávne vyhodnotí a v dôsledku toho sa nemusí anonymizovať. https://www.youtube.com/watch?v=DRz_FOPLLXk

2. **Walking Through the City** *(súčasť videa multiple_vids.mp4)* – prechádzanie mestom plného ľudí. Toto video bolo zvolené kvôli veľkému počtu osôb, ktoré sa nachádzajú v obraze v rovnakom čase. Tento fakt má za úlohu otestovať, či systém dokáže detegovať väčšie množstvo ľudí/tvárí na jednom obraze. To aj v prípade, že sú často krát v obraze deformovaní, najmä rýchlymi pohybmi kamery.
https://www.youtube.com/watch?v=EDWpEhytG0c

3. **The secrets to decoding facial expressions** *(súčasť videa multiple_vids.mp4)* – video obsahujúce takmer cez celý obraz, tvár ženy, ktorá predvádza rôzne mimiky tváre. Úlohou tohto videa je otestovať, či systém dokáže rozpoznať tvár aj pri použití výraznejšej mimiky tváre, a taktiež či systém dokáže dostatočne anonymizovať tvár, ak je tá v obraze jedinou dominantou.
https://www.youtube.com/watch?v=B0ouAnmsO1Y

4. **Dash Cam** *(Dash_Cam_faces.mp4)* – videá nahraté na palubnú kameru v aute. Tieto videá boli zvolené kvôli použitiu v praxi, kedy je v niektorých prípadoch protiprávne zverejňovať identitu ľudí na internete bez ich súhlasu. Zároveň sú videá vybrané z rôznych situácií, teda sa líšia svetelné podmienky, umiestnenie kamery vo vozidle, kvalita záznamu. Všetky tieto aspekty majú za úlohu otestovať funkčnosť

systému na anonymizáciu videí.
https://www.youtube.com/watch?v=iT3lSud3LAU, odkazy sa originálne videá, z ktorých bol zostrih vytvorený[1][2].

5. **Pulp Fiction – scéna z filmu** *(pulp.mp4)* – Video obsahuje jednu scénu z tohto filmu. Bolo zvolené pre potreby testovacej dátovej sady, kvôli rôznym farbám pleti aktérov v tomto videu. Systém teda nemusí správne detegovať niektoré tváre. Taktiež som toto video zakomponoval do sady z dôvodu, že som použil snímky z neho aj pri písaní textovej správy a teda by bolo vhodné, aby ho systém vedel správne spracovať.
https://www.youtube.com/watch?v=pvAhRcUofDk

6. **Trik s kartami** *(card_trick.mp4)* – V tomto videu predvádza kúzelník trik s kartami. Toto video som zvolil z dôvodu, že sa jedná o jeden dlhý a statický záber. Má za úlohu otestovať nečakané výpadky systému a schopnosť spracovať aj dlhé scény.
https://www.youtube.com/watch?v=Lw-PtDEHRiw

7. **Scéna zo seriálu Friends** *(friends.mp4)* – Scéna, ktorá zachytáva veľa osôb v pozadí. Mala za úlohu otestovať, či systém dokáže anonymizovať aj menšie tváre v pozadí videa. Taktiež som chcel zistiť, ako systém zvládne prudké a náhle pohyby aktérov. V dátovej sade je iba výstrižok pôvodnej scény z YouTube.
https://www.youtube.com/watch?v=jbRVoTL5djs

8. **Rozhovor** *(interview.mp4)* – Jedná sa o úryvok zo show Jimmy Kimmela, v ktorom sa nachádzajú osoby poskytujúce rozhovor na kameru. Toto video som zvolil z dôvodu, že sa v ňom nachádza veľa ľudí, rôznych farieb pleti a môžu mať na tvárach rôzne predmety (pokrývky hlavy, okuliare). Taktiež sa jedná o príklad reálnej scény, kedy osoba môže poskytovať rozhovor pre určité médium a nechce, aby bola zverejnená jej identita.
https://www.youtube.com/watch?v=wJdNrCeUdhc

9. **Rozhovor s deťmi** *(kids.mp4)* – Rovnako ako v predchádzajúcom príklade sa jedná o reálnu scénu. Avšak osobami vo videu sú iba deti. Video má otestovať schopnosť systému anonymizovať aj detské tváre, keďže sa môžu od dospelých osôb mierne odlišovať.
https://www.youtube.com/watch?v=HkUGSNNy4dI

10. **Magické triky na ulici** *(magic_50s.mp4)* – Toto video bolo použité aj pri testovaní systému pomocou dotazovacích formulárov. Jedná sa o pomerne dynamickú scénu z ulíc mesta New York, v ktorej sa nachádza väčšie množstvo ľudí. Ľudia sa v scéne rôzne pohybujú a zároveň sa pohybuje aj kamera, ktorá scénu sníma. Video malo otestovať ako sa systém dokáže vysporiadať s takouto dynamickou scénou.
https://www.youtube.com/watch?v=NrNl8IEHruc

11. **Záznamy z palubných kamier** *(dashcam[1-6].mp4)* – Jedná sa o záznamy z palubných kamier áut. Tieto videá majú otestovať schopnosť systému anonymizovať poznávacie značky automobilov. Sú tu obsiahnuté scény za rôznych obrazových podmienok (rôzne počasie, rôzna kvalita záznamu, iný zdroj záznamu).
https://www.youtube.com/watch?v=9K-irRVXgQs

[1] https://www.youtube.com/watch?v=6rOy2RI_Ddw,
[2] https://www.youtube.com/watch?v=dJudHAE_9xU

https://www.youtube.com/watch?v=cLR3r7h4o68 *(*dashcam[2-4].mp4 - zdroj YouTube[3])

12. **Blog o náramkových hodinkách** *(súčasť videa multiple_vids.mp4)* – Video, v ktorom muž prezentuje svoje hodinky a pri tom veľa krát prejde rukou cez svoju tvár. Prechádzanie rukami cez tvár malo overiť schopnosť systému sa vysporiadať s problémom, ak je tvár osoby na malý časový okamih prekrytá iným predmetom.
https://www.youtube.com/watch?v=4korZEd-S-U

13. **Dlhá kompilácia záznamov z palubnej kamery** *(kompilace.mp4)* – Jedná sa o dlhú 11 minút dlhú kompiláciu, ktorá obsahuje rôzne zábery z palubnej kamery v aute. Video malo za úlohu overiť ako sa systém vysporiada s veľmi dlhým videom.
https://www.youtube.com/watch?v=8lPUVNGKbZQ

[3] K týmto videám som presný zdroj stratil.

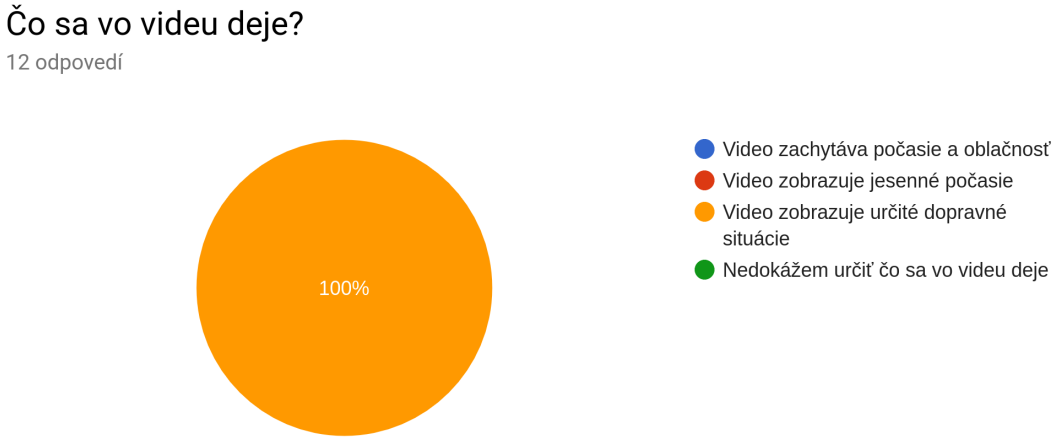
Príloha B

Odpovede na dotazníky

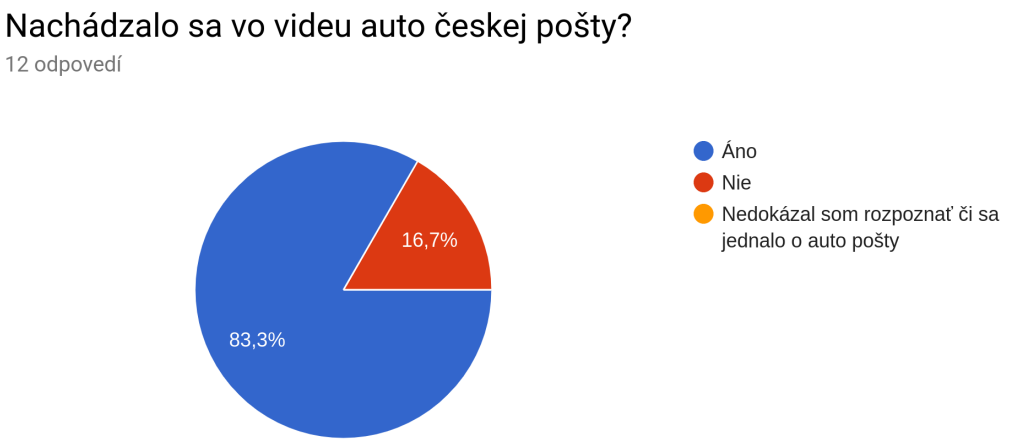
V tejto prílohe sú uvedené odpovede na dotazníky, ktoré sa týkali vyhodnotenia funkčnosti anonymizačného systému. Sú rozdelené na 3 sekcie, podľa dotazníkov.

Dotazník na vyhodnotenie anonymizácie ŠPZ

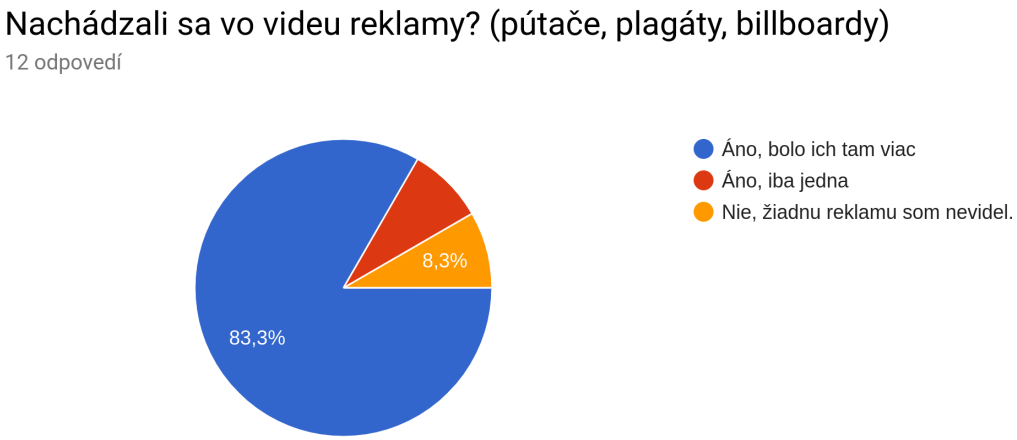
Na nasledujúcich grafoch je možné vidieť odpovede dotazovaných osôb na jednotlivé otázky v tomto dotazníku.



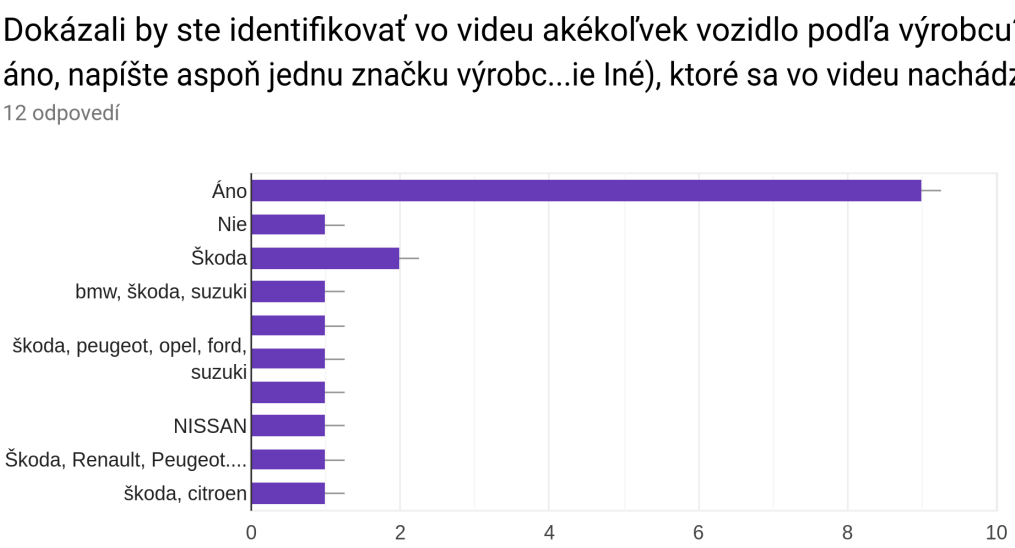
Obr. B.1: Odpoveď 1



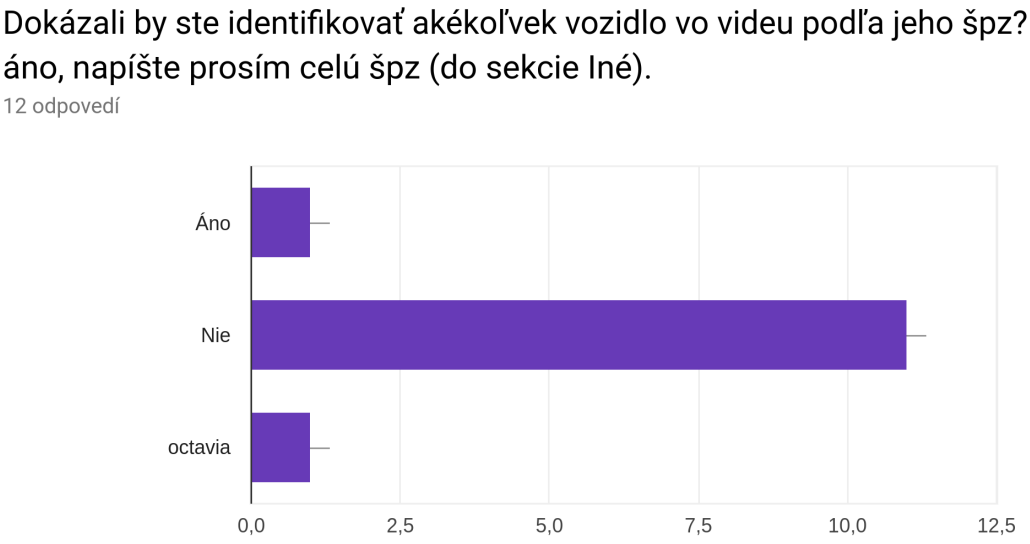
Obr. B.2: Odpoveď 2



Obr. B.3: Odpoveď 3



Obr. B.4: Odpoveď 4

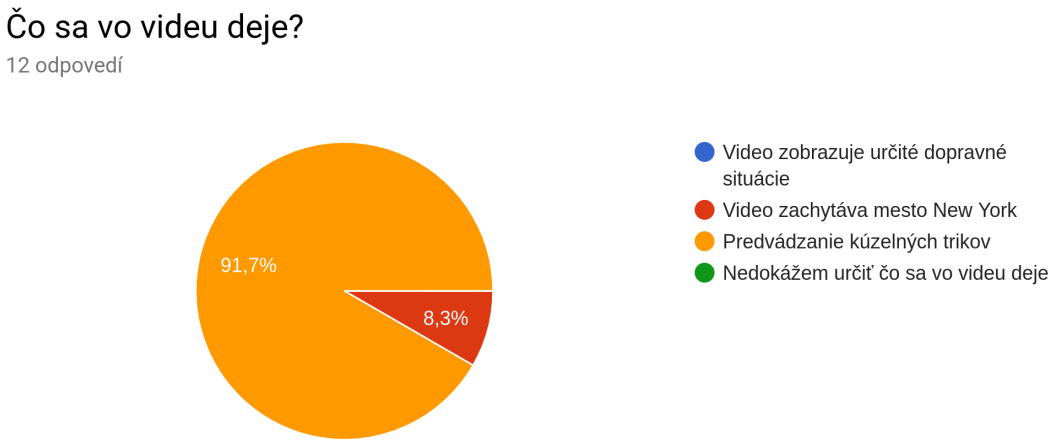


Obr. B.5: Odpoveď 5[1]

[1]Respondent, ktorý zvolil možnosť Áno, pravdepodobne nesprávne pochopil otázku, pretože do sekcie Iné vyplnil odpoveď octavia.

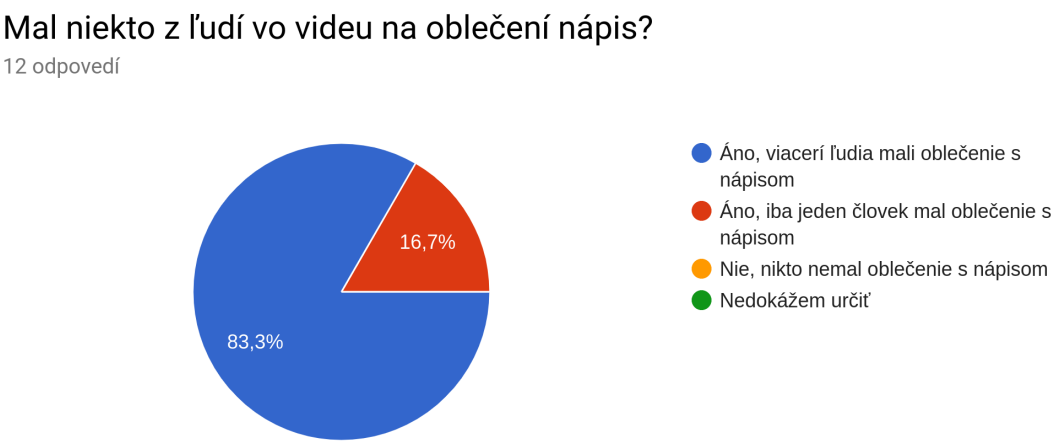
Dotazník na vyhodnotenie anonymizácie tvárí

Na nasledujúcich grafoch je možné vidieť odpovede dotazovaných osôb na jednotlivé otázky v tomto dotazníku.



Obr. B.6: Odpoveď 1

Obr. B.7: Odpoveď 2

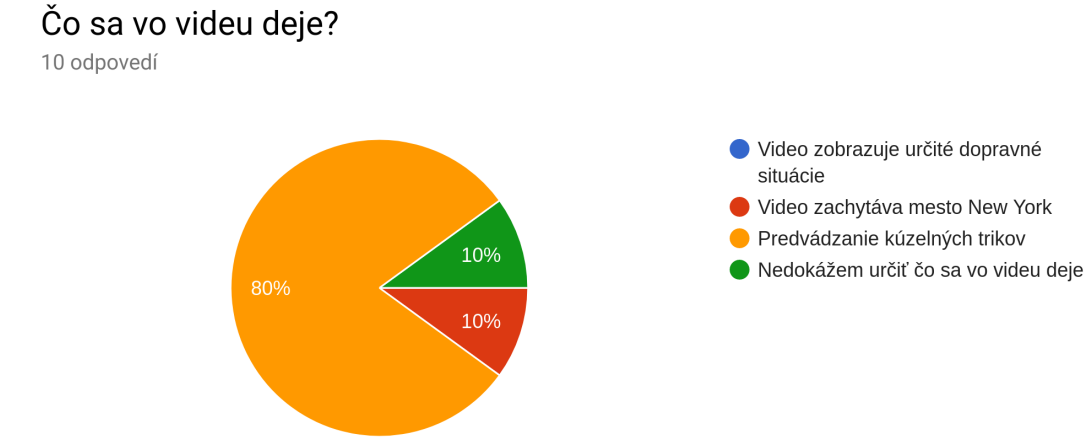


Obr. B.8: Odpoveď 3

Obr. B.9: Odpoveď 4

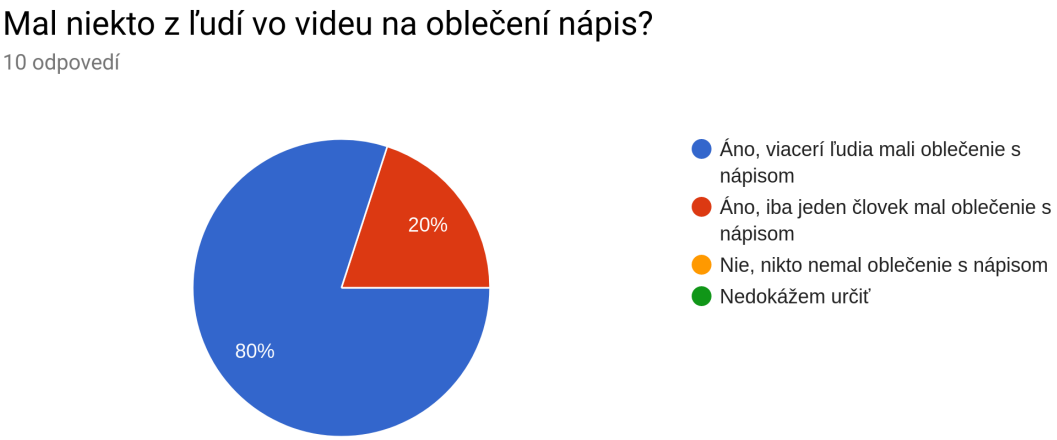
Obr. B.10: Odpoveď 5

Druhý dotazník na vyhodnotenie anonymizácie tvárí



Obr. B.11: Odpoveď 1

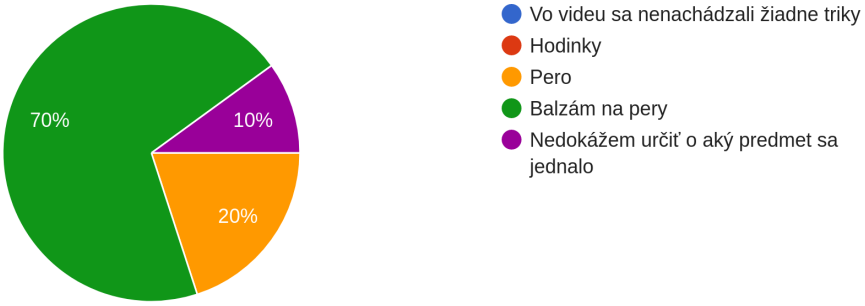
Obr. B.12: Odpoveď 2



Obr. B.13: Odpoveď 3

Aký predmet bol použitý na prípadné triky vo videu?

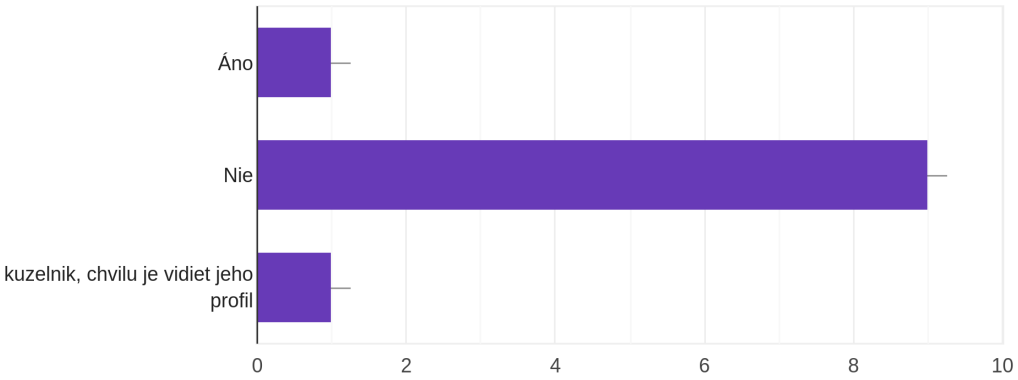
10 odpovedí



Obr. B.14: Odpoveď 4

Dokázali by ste niektorému z hlavných aktérov vo videu opísať tvárové črty? (farba očí, tvar nosa, tvar pier, obočie, ... Iné), a tiež akú časť tváre viete opísať.

10 odpovedí



Obr. B.15: Odpoveď 5

Príloha C

Obsah priloženého disku

Priložené sú dva disky typu DVD. Každý z nich obsahuje jeden súbor typu zip. Z oboch diskov je potrebné tieto súbory skopírovať do spoločnej zložky a následne ich rozbaliť pomocou programu `7Zip` alebo `WinRAR`[1]. Po rozbalení bude vzniknutá súborová štruktúra vyzerať nasledovne:

```
.
├── anonymization/ ..... Priečinok obsahujúci zdrojové súbory.
│   ├── anonym.py ..... Spúšťací súbor.
│   ├── data/ ..... Priečinok obsahujúci datasety a dáta potrebné pre chod programu.
│   ├── experiments/ ..... Priečinok obsahujúci experimenty, skripty k experimentom a výsledky experimentov. Tiež obsahuje aj trénovacie a testovacie datasety.
│   ├── face_detector.py
│   ├── functions.py
│   ├── lp_detector.py
│   ├── object_detector.py
│   ├── req.txt
│   └── scene_detector.py
├── latex/ ..... Priečinok obsahujúci zdrojové súbory (LaTeX) tejto správy.
├── poster/ ..... Priečinok obsahujúci plagát.
├── readme.txt ..... Súbor obsahujúci podrobný návod na spustenie a ovládanie programu.
├── thesis/ ..... Táto správa vo formáte pdf.
└── video/ ..... Priečinok obsahujúci video.
```

[1] S týmito programami bolo rozbaľovanie testované.