



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLEDOVÁNÍ A ROZPOZNÁVÁNÍ LIDÍ NA VIDEU

TRACKING AND RECOGNITION OF PEOPLE IN VIDEO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ANTONÍN ŠAJBOCH

VEDOUcí PRÁCE
SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2016

Abstrakt

Tato práce se zabývá rozpoznáváním a sledováním lidí ve videu. K rozpoznávání byla použita konvoluční neuronová síť, která extrahuje vektor rysů z přiloženého snímku obličeje. Tento extrahovaný vektor je dále klasifikován. Proces rozpoznávání musí probíhat v reálném čase a s tímto ohledem jsou také voleny veškeré použité metody. V rámci této práce byl vytvořen nový dataset obličejů, který byl získán z videozáznamů pořízených v prostorách fakulty. Videozáznamy a dataset byly použity pro experimenty k ověření přesnosti vytvořeného systému. Výsledná přesnost rozpoznávání je cca 85%. Navržený systém může být použit například k evidenci osob, počítání průchodů či k ohlášení výskytu neznámé osoby v objektu.

Abstract

The master's thesis deals with detecting and tracking people in the video. To get optimal recognition was used convolution neural network, which extracts vector features from the enclosed frame the face. The extracted vector is further classified. Recognition process must take place in a real time and also with respect are selected optimal methods. There is a new dataset faces, which was obtained from a video record at the faculty area. Videos and dataset were used for experiments to verify the accuracy of the created system. The recognition accuracy is about 85%. The proposed system can be used, for example, to register people, counting passages or to report the occurrence of an unknown person in a building.

Klíčová slova

Rozpoznávání, rozpoznávání obličejů, detekce obličejů, konvoluční neuronové sítě, sledování objektů, SVM, optický tok

Keywords

Recognition, face recognition, face detection, convolutional neural network, object tracking, SVM, optical flow

Citace

ŠAJBOCH, Antonín. *Sledování a rozpoznávání lidí na videu*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Smrž Pavel.

Sledování a rozpoznávání lidí na videu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Smrže. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Antonín Šajboch
24. května 2016

Poděkování

Na tomto místě bych rád poděkoval vedoucímu diplomové práce doc. RNDr. Pavlu Smržovi, Ph.D, za cenné připomínky a odborné rady, kterými přispěl k vypracování této práce.

© Antonín Šajboch, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Detekce objektů v obraze	4
2.1	Extrakce pozadí	4
2.2	Metody detekce	5
2.3	Detektor Viola&Jones	6
2.4	HOG detektor	8
3	Sledování objektů v obraze	11
3.1	Optický tok	11
3.1.1	Algoritmus Lucas-Kandle	11
3.1.2	KLT tracker	13
3.2	Kalmanův filtr	14
4	Rozpoznávání obličeje	16
4.1	Neuronové sítě	16
4.1.1	Obecný model umělého neuronu	16
4.1.2	Topologie	17
4.1.3	Učení neuronové sítě	17
4.2	Konvoluční neuronové sítě	18
4.2.1	Konvoluční vrstva	18
4.2.2	Vrstva podvzorkování	19
4.2.3	Techniky používané v konvolučních neuronových sítích	20
4.3	Klasifikace	20
4.3.1	KNN klasifikátor	21
4.3.2	Klasifikátor SVM	21
4.3.3	Nearest centroid	21
4.3.4	Metriky pro hodnocení klasifikátorů	22
4.4	Shluková analýza	24
4.4.1	DBSCAN	24
5	Implementace	26
5.1	Požadavky	26
5.2	Struktura systému	26
5.2.1	Předzpracování snímků	28
5.2.2	Detekce	28
5.2.3	Sledování	28
5.2.4	Rozpoznávání obličeje	30

5.3	Paralelní zpracování	32
5.4	Analýza trajektorie a počítání průchodů	33
6	Experimenty	35
6.1	Použité datové sady	35
6.2	Experimenty s extrahovaným vektorem rysů	35
6.2.1	Verifikace	36
6.2.2	Rozpoznávání pohlaví	37
6.3	Identifikace ze statických snímků	38
6.4	Experimenty s implementovaným systémem	40
6.5	Časová složitost	41
6.6	Souhrn	42
7	Závěr	43
	Literatura	44
	Přílohy	46
	Seznam příloh	47
A	Obsah CD	48

Kapitola 1

Úvod

Od dvacátého století se kriminalisté zabývají rozpoznáváním osob na základě lidské tváře. Využívají k tomu zejména portrét osoby, který je nejdůležitější částí při identifikaci osob na základě vizuálního vjemu. Při charakteristice osoby je velmi těžké popsat držení těla, styl chůze či řeč těla, ale velmi specificky lze popsat charakteristiky obličeje, jako je tvar, barva pleti a vlasů.

Pro člověka je charakteristické, že od sebe dokáže rozeznat objekty různého typu. Také dokáže na základě obličeje rozpoznat konkrétního člověka, ale v oblasti počítačového vidění se jedná o netriviální úlohu. Cílem takového vidění je tudíž to stejné, a to naučit počítač rozpoznávat lidi a použít to v oblastech, které nám usnadní život. Takové rozpoznávání se dá využít například pro sledování přístupů do budovy nebo k nalezení osob, které dříve spáchaly trestný čin.

Rozpoznávání obličejů v počítačovém vidění je stále otevřená úloha, což znamená, že doposud neexistuje systém, který by dokázal rozpoznat člověka podle obličeje se stoprocentní úspěšností.

Ve své práci využívám konvoluční neuronové sítě pro rozpoznávání osob, které jsou v této oblasti poslední dobou často využívány. Proces rozpoznávání je složen ze tří částí. Nejprve je nutné obličej ve scéně nalézt, poté jej sledovat a následně identifikovat podle již uložených vzorů obličejů v databázi.

Jak již bylo naznačeno, práce je strukturována do tří stěžejních částí, a to teoretický základ, který sestává z prvních tří kapitol, kde jsou popsány metody pro detekci obličeje, jeho sledování a následná identifikace. V dalších částech je popsána implementace systému a nakonec experimenty.

V první části se zaměříme na pužité metody pro detekci obličeje v obraze. Následovat bude přehled algoritmů pužitých ke sledování objektů v obraze. Dále se budeme zabývat extrakcí rysů z obrazu obličeje a následnou klasifikací extrahovaných vektorů.

Druhá část je praktickým výstupem práce a zabývá se implementací systému pro rozpoznávání lidí na videu za pužití metod popsanych v teoretické části práce.

V závěru vyhodnotíme pužité algoritmy a porovnáme úspěšnost algoritmů s jinými metodami zveřejnými v některých vědeckých pracích.

Kapitola 2

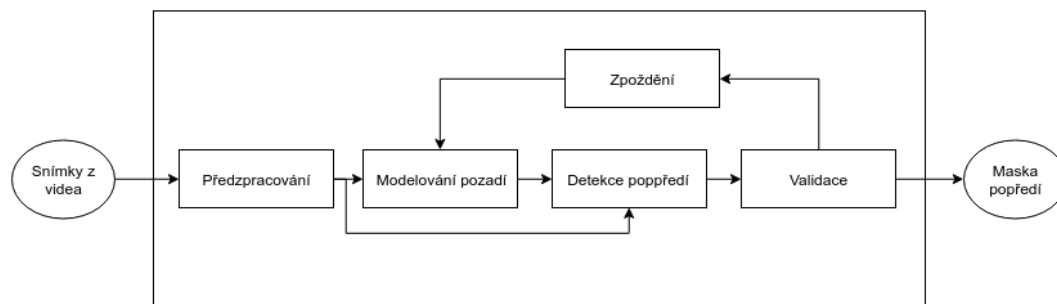
Detekce objektů v obraze

K vytvoření kvalitního a robustního systému pro identifikaci osob ve videu je potřeba v reálném čase detekovat osobu a její obličej. Detekce obličeje v reálném čase je omezujícím faktorem pro výběr vhodných metod pro detekci objektů v obraze. Je tedy nezbytně nutné zvolit takovou metodu, jejíž časová složitost bude co nejmenší při zachování jisté přesnosti.

2.1 Extrakce pozadí

Extrakcí pozadí ze sekvence snímků získáme ořezovou masku pohybujících se objektů. Veškeré metody pro extrakci pozadí vyžadují videozáznam, který byl pořízen stacionární kamerou. Detekce spočívá v segmentaci záznamu na neměnné statické pozadí a popředí, jenž představuje pohybující se objekt. Na záznamu se mohou pohybovat jiné objekty než lidé, proto je popředí bráno pouze jako předzpracování obrazu pro následné detekce lidí.

Pro odečet pozadí existuje mnoho algoritmů. Většina z nich využívá jednoduchý diagram na obrázku 2.1, který se skládá ze čtyř procesů.



Obrázek 2.1: Diagram obecné extrakce popředí ze záznamu. Převzato z [6]

Předzpracování

Nejprve se redukuje šum v jednotlivých snímcích vyhlazením [6]. Také se velmi často zmenšuje rozlišení snímků, aby byla extrakce pozadí co nejrychlejší. V některých případech se transformuje formát vstupního obrazu a to tak, že se barevný snímek převádí na šedotónový. Algoritmus pak pracuje pouze s intenzitou pixelů, což je pro většinu případů dostačující.

Modelování pozadí

Modelování pozadí je nejdůležitější částí všech algoritmů pro extrakci popředí. Vytváření modelu pozadí lze rozdělit do dvou kategorií a to rekurzivní a nerekurzivní. Nerekurzivní techniky jsou vysoce adaptivní a jsou nezávislé na snímcích, které nejsou uloženy v zásobníku [6]. Ovšem čím větší je zásobník, tím hůře se detekují pomalu pohybující se objekty. Naopak rekurzivní techniky nepotřebují zásobník předcházejících snímků. Místo zásobníku rekurzivně modifikují model pozadí na základě každého vstupního snímku [6]. V porovnání s nerekurzivními metodami potřebují rekurzivní metody méně paměti, ale chyba v modelu přetrvává delší čas.

Detekce popředí

Detekce popředí porovnává vstupní snímek s modelem pozadí a identifikuje kandidátní pixely popředí. Nejčastěji používanou metodou pro identifikaci pixelů popředí je porovnání, jak moc jsou pixely modelu pozadí a vstupního snímku rozdílné. Například pomocí následující rovnice:

$$|I_t(x, y) - B_t(x, y)| > T, \quad (2.1)$$

kde $I_t(x, y)$ je intenzita pixelu vstupního snímku na souřadnicích $[x, y]$, $B_t(x, y)$ je intenzita pixelu modelu pozadí na souřadnicích $[x, y]$ a T je práh udávající jak moc musí být intenzity rozdílné, aby byl pixel posouzen jako popředí.

Validace

Validace je proces, který zpřesňuje kandidátní pixely extrahovaného popředí. Všechny modely pozadí z obou kategorií mají tři omezení:

1. ignorují jakoukoliv korelaci mezi sousedními pixely,
2. rychlost adaptace se nemusí shodovat s rychlostí pohybujících se objektů v popředí,
3. nestacionární obrazové body, například stíny, jsou často chápány jako skutečné objekty popředí.

Důsledkem prvního problému bývají typicky malé neidentifikované oblasti rozmístěné mezi pixely identifikované jako popředí. Často se používá filtrování, které tyto malé oblasti odstraní, čímž se z těchto malých regionů stane jeden velký region popředí. Také je možné definovat minimální velikost oblasti, a pokud se detekuje menší, zahodí se.

Pokud se model aktualizuje velmi pomalu, zůstávají v popředí chybně identifikované oblasti. Aktualizuje-li se příliš rychle, můžou se v obraze oblasti ztrácet. Řešením může být použití více modelů pozadí a jejich zprůměrování.

2.2 Metody detekce

Detekce obličejů spadá do oblasti počítačového vidění a je zkoumána již řadu let. Z tohoto důvodu již existuje mnoho různých řešení daného problému. Při detekci obličejů je potřeba brát v potaz, že obličej na snímcích je různorodý. Různé obličejy se tedy od sebe liší odstínem pleti, tvarem a úhlem natočení. Část obličejů také může být překryta různými objekty,

například brýlemi či rouškou. Všechny tyto faktory musíme při detekci obličeje brát v potaz a zvolit takovou metodu, která pokryje co nejvíce zmíněných možností.

Metody detekce obličeje v obraze můžeme rozdělit podle různého přístupu do čtyř následujících kategorií [19]:

- metody založené na znalostech,
- metody založené na příznacích,
- metody založené na porovnání,
- metody založené na podobnosti.

Metody založené na znalostech fungují na principu získaných znalostí o lidském obličeji. Z těchto znalostí jsou vyvozena jednoduchá pravidla. Oči jsou symetrické, nos se nachází mezi očima a ústa jsou ve spodní části obličeje. Mezi těmito částmi obličeje existují vztahy, které jsou definovány vzájemnými vzdálenostmi a pozicemi. V obraze, ve kterém je hledán obličej, se pak postupně aplikují jednoduchá pravidla. Největším nedostatkem tohoto přístupu je správné definování pravidel. Pokud jsou pravidla příliš obecná, pak dochází k velkému počtu falešných detekcí a naopak. Jestliže jsou pravidla příliš detailní, může docházet k velkému počtu zamítnutí.

Metody založené na příznacích se snaží nalézt takové příznaky obličeje, které jsou imunní vůči různému typu osvětlení a natočení obličeje. Jako příznaky se pak dají použít například ústa, nos nebo obočí. Následně se pomocí statistického modelu snaží metody rozhodnout, zda o obličej jde nebo ne.

Metody založené na porovnávání pracují na principu porovnávání částí obrazu se šablonou obličeje nebo pouze jeho částí (oči, ústa, nos). Šablona obličeje se vytváří ručně, což není příliš efektivní, a proto jsou tyto metody používány nejméně.

U metod založených na podobnosti je jediným rozdílem, že šablona obličeje se vytváří pomocí statistické analýzy a strojového učení. Díky tomu je možné nalézt charakteristické rysy pro obrazy, na kterých se obličej nachází. Tímto vyniká klasifikátor, který dokáže rozhodnout, zda se v obraze na dané oblasti obličeje nachází či nikoliv. Tento klasifikátor musí být natrénován na velkém množství trénovacích vzorů. V trénovací sadě se musí nacházet obrazy s obličejem, ale také obrazy, kde se obličej nevyskytuje. Ve své diplomové práci jsem se rozhodl požit detektor z této skupiny, a to konkrétně detektor Viola&Jones.

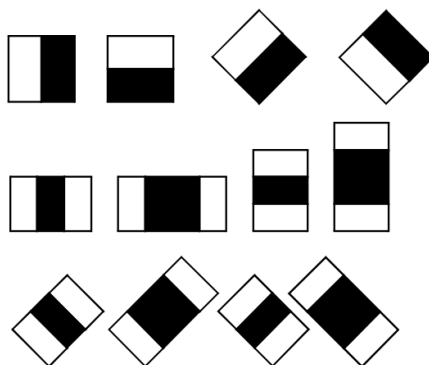
2.3 Detektor Viola&Jones

Jedna z nejznámějších metod pro detekci obličeje v obraze byla představena P.Violou a M.Jonesem v roce 2001 [20]. Detektor využívající tuto metodu je obsažen v knihovně OpenCV [14], kterou ve své práci budu používat. Pro detekci využívá předtrénovaný klasifikátor. Jedná se o velmi rychlý detektor, proto se hodí pro detekci v reálném čase. Autoři uvádí, že je možné zpracovat až 15 snímků za vteřinu. V současné době je možné zpracovat mnohem více snímků díky vývoji výpočetní techniky.

Detektor Viola&Jones pracuje pouze s obrazy ve stupních šedi. Samotná detekce využívá Haarovy příznaky. Sady těchto příznaků jsou pak testovány s využitím AdaBoost na jednotlivých částech obrazu. [20]

Haarovy příznaky jsou tvořeny obdélníkovými tvary, které mají různou orientaci a jsou bílé nebo černé. Samotná hodnota příznaku je pak dána rozdílem sumy intenzit pixelů pod

bílými obdélníky a sumou intenzit pixelů pod černými obdélníky. Haarův příznak pak může být tvořen z několika těchto obdélníkových tvarů, jak lze vidět na obrázku 2.2.



Obrázek 2.2: Haarovy příznaky. Převzato z [17]

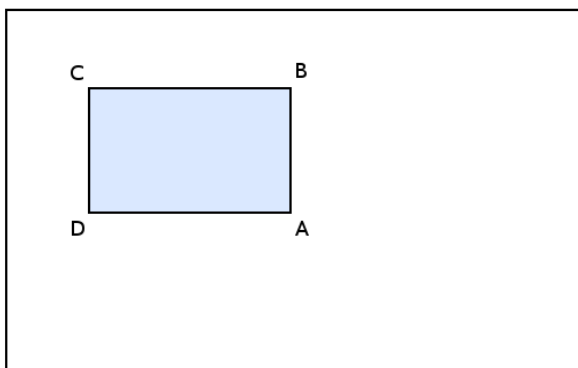
Pro detekci obličeje je možno využít znalostí, že jednotlivé přechody (např. oči-tváře, nos-ústa apod.) jsou detekovány na stejných místech a mají podobné diference. Metoda Viola&Jones využívá příznaky složené ze dvou, tří a čtyř obarvených obdélníků. Příznaky jsou zjišťovány ze snímků, které jsou transformovány na velikost 24×24 pixelů [20]. Pro sestavení detektoru se pak vyberou pouze ideální příznaky pomocí algoritmu AdaBoost.

Pro rychlejší výpočet hodnot Haarových příznaků se původní obraz převede na integrální obraz. Integrální obraz uchovává na souřadnicích x, y součet intenzit všech pixelů nacházejících se vlevo nahoře od souřadnic x, y . Z toho vyplývá, že na pozici úplně vpravo dole bude suma intenzit všech pixelů a vlevo nahoře, tedy na souřadnicích $(0, 0)$, se bude nacházet hodnota 0, což lze vyjádřit vztahem [17]:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (2.2)$$

kde $ii(x, y)$ je souřadnice v integrálním obraze a $i(x', y')$ je pozice v původním obraze.

Integrální obraz je pro výpočet Haarových příznaků velmi užitečný, protože výsledná suma intenzit pixelů v určité oblasti lze zjistit pouze ze čtyř bodů. Pokud bychom chtěli zjistit sumu intenzit v určité oblasti z normálního obrazu, museli bychom projít celou oblast pixel po pixelu. Na obrázku 2.3 můžeme vidět vymezení bloku pro výpočet sumy intenzit.



Obrázek 2.3: Obdélníková oblast v integrálním obraze

Pokud bychom chtěli vypočítat sumu intenzit v oblasti dané vnitřním obdélníkem na obrázku 2.3, tak nám postačí pouze 4 body a to A , B , C a D , které vymezují daný obdélník. Pomocí následujícího vzorce spočteme sumu intenzit vymezeného obdélníku:

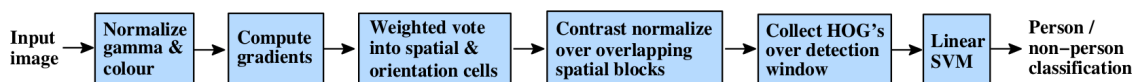
$$I = I(A) + I(C) - I(B) - I(D) \quad (2.3)$$

Pro zrychlení zpracování vstupního obrazu se používá tzv. kaskáda klasifikátorů, což umožňuje zpracování obrazu v reálném čase. Jedná se o techniku, kdy se silný klasifikátor rozdělí na několik dílčích klasifikátorů, z nichž každý je tvořen lineární kombinací slabých klasifikátorů [21]. Nově vzniklé klasifikátory pak tvoří kaskádu. Klasifikátory se poté aplikují na určitou část obrazu postupně. Pokud některý z klasifikátorů v kaskádě selže, tedy vyhodnotí, že se hledaný objekt ve zkoumané části obrazu nenachází, pak se kaskáda klasifikátorů aplikuje na jinou část obrazu. Hledaný objekt se nachází v prohledávané části obrazu pouze tehdy, pokud tak vyhodnotí všechny klasifikátory v kaskádě.

2.4 HOG detektor

Histogram of Oriented Gradients je metoda, která je založena na jednoduchých příznacích. Jedná se o typ metody statistického rozpoznávání vzorů. Hlavní výhodou tohoto detektoru je nezávislost na velikosti objektu. Metoda je použitelná především k rozpoznávání objektů, které nemění své natočení v obraze. Gradient, neboli intenzita barevné změny, se používá pro popis pixelů v obraze. Při výpočtu gradientu z černobílých obrazů se získá pouze jedna hodnota. U barevných obrazů se počítá gradient pro každou barevnou složku zvlášť.

Rozpoznávání objektu v obraze probíhá pomocí vhodně získaných příznaků, které jsou následně předávány klasifikátoru. Příznaky jsou zastoupeny tzv. HOG deskriptory. Deskriptory se skládají z většího množství histogramů, které jsou spočteny pro každý pixel v obraze. Histogramy jsou určovány lokálními buňkami v obraze o stanovené velikosti, díky čemuž jsou například velmi málo náchylné na změny osvětlení.



Obrázek 2.4: Extrahování příznaků a detekce objektu v obraze pomocí HOG deskriptorů. Převzato z [7]

Princip detekce

Nejprve projde vstupní obraz předzpracováním a následně se vypočtou gradienty pro všechny pixely v obraze. Následovně se rozdělí obraz do mřížky disjunktních buňek o stejné velikosti. Pro každou buňku se vypočte vážený histogram orientací gradientů. Obraz se rozdělí na překrývající se bloky dané velikosti obsahující určitý počet buňek. Normalizují se histogramy v každém bloku. Kombinace těchto histogramů je odeslána lineárnímu SVM klasifikátoru, který rozhodne, zdali se jedná o hledaný objekt, či nikoliv.

Výpočet gradientu

Výpočet gradientu spočívá v nalezení lokálních změn intenzit sousedních pixelů v obraze. Obraz může být definován jako funkce $f(x, y)$, jejíž výpočtenou hodnotou je intenzita pixelu

o souřadnicích x a y . Gradient G je dvousložkovým vektorem vzhledem k tomu, že funkce f je dvoudimenzionální.

$$G = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{pmatrix} \quad (2.4)$$

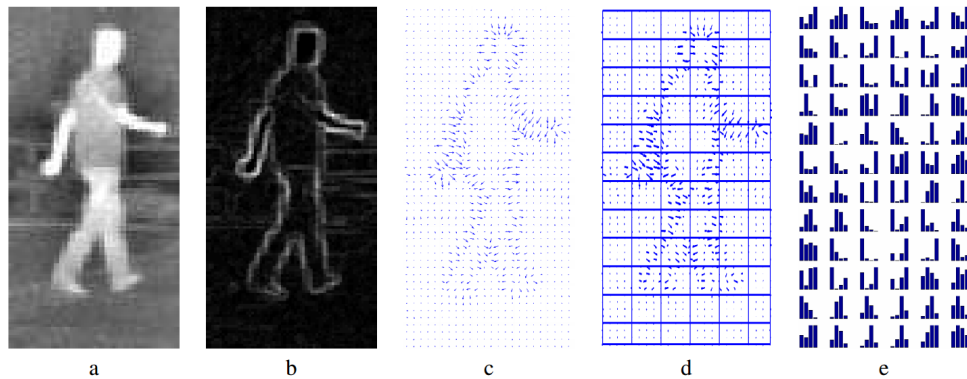
Velikost gradientu G je definována vztahem:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.5)$$

Směr gradientu G v bodě $[x, y]$ je definován jako:

$$O = \arctan\left(\frac{G_x}{G_y}\right) \quad (2.6)$$

K určení lokálního histogramu je nutné obraz rozdělit mřížkou o určité velikosti. Optimální velikost mřížky se pohybuje mezi $4 \times 4 px$ a $8 \times 8 px$, jak ukázal Dalal ve své práci [7]. Pro každou buňku je určen histogram zvlášť. Kanály histogramu jsou určeny orientací gradientů uvnitř buňky. Počet kanálů je libovolný, ale v práci [7] je ukázáno, že nejlepší rozdělení je do devíti kanálů. Na následujícím obrázku 2.5 je zobrazen princip charakterizace obrazu pomocí HOG.



Obrázek 2.5: Charakterizace osoby pomocí HOG. Převzato z [1]

Na obrázku 2.5 je vizuálně zachycen výše popisovaný postup výpočtu gradientu a určení histogramů, a to následovně:

- a. černobílý obraz zachycující člověka,
- b. gradienty z originálního obrazu,
- c. orientace gradientů,
- d. rozdělení do buňek,
- e. výpočet histogramů.

Před výsledným rozhodováním, zda se hledaný objekt nachází v obraze, je vhodné obraz rozdělit znovu do překrývajících se bloků a ty poté normalizovat. V rámci těchto bloků se provádí normalizace histogramů v jednotlivých buňkách. Normalizace se pak provádí pomocí vzorce:

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}, \quad (2.7)$$

kde f je normalizovaný vektor, v je vektor před normalizací, $\|v\|_2^2$ je velikost vektoru v a e je malá konstanta, u které příliš nezáleží na hodnotě.

Deskriptor bloku je sestaven z normalizovaných histogramů buňek, a to prostou konkatencí vektorů reprezentujících histogramy buňek. Tento deskriptor je pak předán klasifikátoru, jenž rozhodne, zda se v obraze nachází hledaný objekt.

Kapitola 3

Sledování objektů v obraze

Detekované objekty je vhodné sledovat v sekvenci snímků tak, aby nemuselo docházet na každém snímku k reidentifikaci osoby, která byla na předchozím snímku detekována a rozpoznána. Sledování objektů také umožní získat snímek obličeje z různých úhlů pohledu, což může zkvalitnit následnou identifikaci osoby. Díky sledování objektu je také možné získat trajektorii pohybujícího se objektu. Sledování objektů v obraze je jeden z problémů počítačového vidění, pro který existuje řada řešení.

3.1 Optický tok

Optický tok je založen na myšlence, že jasová hodnota pixelů v obraze má plynulý přechod, protože sousední hodnoty mají podobnou jasovou hodnotu. Rovnice vyjadřující plynulý přechod jasu je vyjádřena jako [11]:

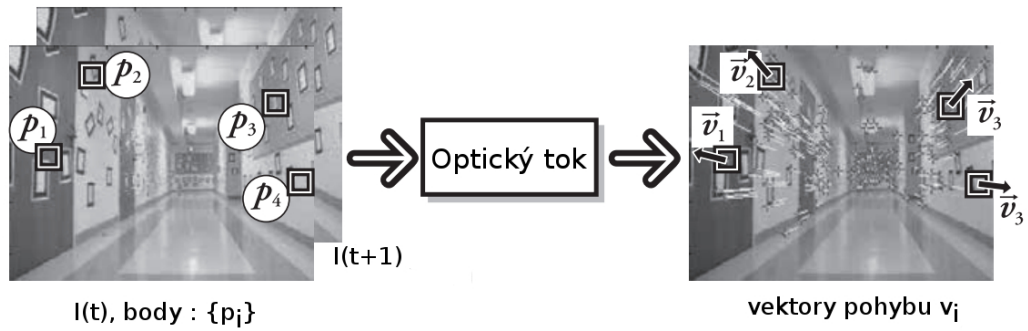
$$\frac{\partial b}{\partial t} + v \nabla b = 0, \quad (3.1)$$

kde v je vektor rychlosti a b je funkce jasu. Optický tok odráží změnu v obraze způsobenou pohybem v čase t mezi snímky. Výsledkem optického toku je vektor určující pohyb pixelu v sekvenci snímků. Teoreticky je tedy možné určit pro dva po sobě jdoucí snímky posun pixelu v souřadnicích x a y . Uvažme například ve scéně pohybující se jednobarevný míček. I přes to, že je rozdíl polohy míčku mezi snímky patrný, je obtížné určit optický tok každého pixelu míčku v obraze. V tomto případě je možné určit pouze optický tok pro hrany.

Určení optického toku pro každý pixel v obraze se označuje jako hustý optický tok. Výpočet tohoto druhu optického toku je časově velmi náročné. Naopak, pokud se nepočítá optický tok pro všechny pixely v obraze, ale počítá se pouze pro vybrané, označujeme tento optický tok jako řídký. Výpočet řídkého optického toku spočívá ve správném výběru pixelů a zvolení vhodné metody. Nejvíce používanou metodou pro výpočet řídkého optického toku je Lucas-Kandle.

3.1.1 Algoritmus Lucas-Kandle

Jedná se o jednoduchou techniku, která dokáže určit pohyb vhodně zvolených pixelů v obraze. Ideálně je pohybový vektor (u, v) vypočítán pro každý takový pixel ze dvou po sobě jdoucích snímků, jak můžeme pozorovat na obrázku 3.1.



Obrázek 3.1: Pohyb bodů p_i zachycený metodou Lucas-Kandle. Převzato z [4]

Metoda Lucas-Kandle využívá pro výpočet optického toku lokální diferenciální techniku, která je založena na následujících předpokladech:

1. sledovaný bod mezi snímky je časově neměnný,
2. bod se nepohybuje o velkou vzdálenost mezi dvěma po sobě jdoucími snímky,
3. bod se pohybuje stejně tak, jak se pohybuje jeho blízké okolí [18].

Uvažme dva po sobě jdoucí šedotónové obrázky. Hledanou změnu v ose x označme u a změnu v ose y v . Pokud v prvním předpokladu rozšíříme funkci intenzity o časovou složku t a dt reprezentující změnu snímku, můžeme předpoklad zapsat následovně:

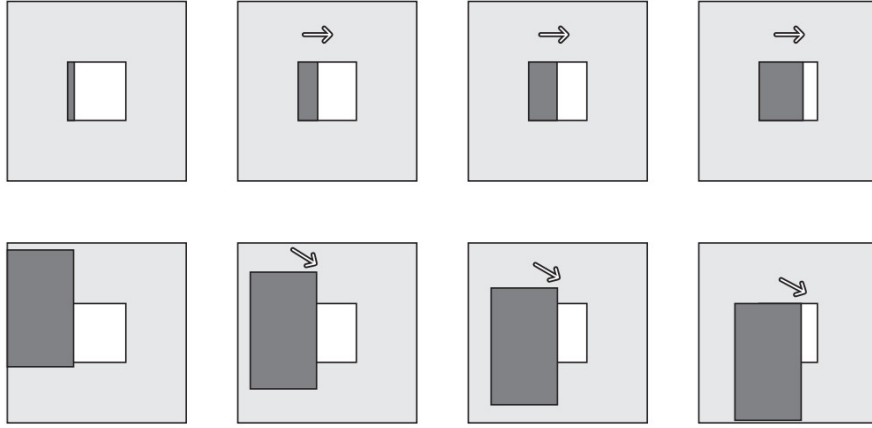
$$I(x, y, t) = I(x + u, y + v, t + dt) \quad (3.2)$$

Druhý předpoklad umožňuje vyjádřit gradienty změny intenzit pixelů v sekvenci snímků a určit směr pohybu. Spojením těchto dvou předpokladů získáme rovnici pro výpočet optického toku 3.3.

$$I_x u + I_y v + I_t = 0 \quad (3.3)$$

kde složky $I_x u$, $I_y v$, I_t jsou gradienty v příslušných směrech a neznámé u a v reprezentují pohyb v daném směru.

Z toho vyplývá, že rovnice 3.3 nelze spočítat, protože obsahuje dvě neznámé. Tato vlastnost je označována jako *aperture problem*. Jedná se o problém, kdy sledujeme pohyb objektu přes malé okno a nelze přesně určit pohyb celého objektu.



Obrázek 3.2: Aparature problem. Převzato z [4]

Na obrázku 3.2 v horní řadě sledujeme pohyb objektu přes malé okno, ale skutečný pohyb objektu lze pozorovat ve spodní řadě.

Abychom mohli rovnici 3.3 řešit, musíme zavést dodatečné omezení. Algoritmus Lucas-Kandle nefunguje pouze pro jeden bod, ale vyžaduje více bodů ke sledování a pro každý bod p_1, \dots, p_i , je možné vytvořit rovnici:

$$\begin{aligned}
 I_x(p_1)u + I_y(p_1)v + I_t(p_1) &= 0 \\
 &\vdots \\
 I_x(p_i)u + I_y(p_i)v + I_t(p_i) &= 0
 \end{aligned} \tag{3.4}$$

Jestliže vyřešíme tyto rovnice některou z numerických metod, například metodou nejmenších čtverců, získáme vektor optického toku.

Při pomalu se pohybujících objektech, při jasové stálosti a při prostorové koherenci pracuje Lucas-Kandle nejlépe. Prostorová koherence značí, že se spolu se sledovaným bodem také pohybují okolní pixely.

3.1.2 KLT tracker

Kandle-**L**ucas-**T**omasi tracker je algoritmus, který pro sledování objektu mezi dvěma po sobě jdoucími snímky využívá odhad optického toku. Pro výpočet optického toku používá metodu Lucas-Kandle.

Tento algoritmus vyhledává vzorový obraz $T(x)$ ve vstupním snímku $I(x)$, kde x je sloupcový vektor nesoucí souřadnice obrazu $[x, y]^T$. Vzor může být definovaný jako jeden bod nebo jako bod s okolím. Pohyb v obraze je poté definován maticí $W(x; p)$, kde w je vektor parametrů pohybu [2]. Pro postupný posun pak platí:

$$W(x; p) = \begin{bmatrix} x + p_1 \\ x + p_2 \end{bmatrix}. \tag{3.5}$$

Jelikož je více rovnic než proměnných, algoritmus hledá podobnost vzoru $T(x; p)$ v obraze $I(x)$ pomocí metody nejmenších čtverců.

Tento algoritmus je určen pro vyhledávání významných bodů mezi snímky. Uvedme například dva po sobě jdoucí obrazy z videa, oznažené jako A a B . Pak je jasová hodnota

těchto obrazů určena jako $A(x, y)$ a $B(x, y)$. Algoritmus pak hledá bod z obrazu A v obraze B . Hledaný bod označme jako $u = [o_x, o_y]$, pak algoritmus hledá v obraze B bod v , který může být definován jako:

$$v = u + d = [u_x + d_x, u_y + d_y], \quad (3.6)$$

kde vektor d nazýváme jako optický tok v bodě u .

Hiearchický přístup Je známé, že v praxi při zpracování sekvence snímků se často mohou vyskytovat pohyby přes velkou oblast obrazu. V našem případě může nastat situace, kdy bude docházet ke zpracování každého k -tého snímku, což může způsobovat velké pohyby napříč obrazem. Tohle je problém u diferenciálních metod výpočtu optického toku, kde se snižuje přesnost prostorových a časových gradientů s velkými pohyby. Řešením je použití hierarchického přístupu, pomocí kterého je vytvořena série snímků se sníženou velikostí [3]. Výsledkem je sada snímků se sníženým rozlišením uspořádaných v pyramidové datové struktuře, jako je Laplacova či Gaussova pyramida [5]. Při výpočtu je odhadnuta hodnota optického toku pro snímek s nejnižším rozlišením, která charakterizuje optický tok většího pohybu. Získané hodnoty jsou následně promítnuty do následující úrovně pyramidy, kde jsou hodnoty použity jako inicializační a následně jsou zpřesněny. Poté jsou opět promítnuty do nižší vrstvy pyramidy, kde jsou opět zpřesněny. Tento proces se opakuje tak dlouho, dokud není dosaženo nejnižší úrovně pyramidy.

3.2 Kalmanův filtr

Kalmanův filtr je jeden z nejrozšířenějších algoritmů pro filtraci zašumělých dat. Jedná se o rekurzivní algoritmus pro odhad dynamického chování systému. Na základě předcházejících měření odhaduje následující stav.

Velmi často se používá pro odhad polohy x_k , která je vyjádřena lineární diferenční rovnicí na základě měření skutečného stavu z_k :

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (3.7)$$

$$z_k = Hx_k + v_k, \quad (3.8)$$

kde A je převodní matice $n \times n$, B je matice $n \times c$ obsahující kontrolní vstupy, H je matice $m \times n$, w_k je procesní šum a v_k je chyba měření.

Cyklus Kalmanova filtru se skládá z predikce (odhad stavu) a korekce (aktualizace hodnoty). Krok predikce se skládá z rovnic realizujících výpočet následujícího stavu a kovariance chyby:

$$\hat{x}_k^- = Ax_{k-1} + Bu_{k-1} \quad (3.9)$$

$$P_k^- = AP_{k-1} + A^T + Q \quad (3.10)$$

Korekce polohy pak přidá hodnoty získané měřením a aktualizuje proměnné pro výpočet nového stavu následovně:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3.11)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (3.12)$$

$$P_k = (I - K_k H) P_k^- \quad (3.13)$$

kde K_k je Kalmanův zisk. Udává jaká je váha nové informace ve srovnání s odhadem. Následně je možné aktualizovat předpověď x_k a kovarianci chyby odhadu P_k . Z takto korelovaných hodnot je následně možné znovu predikovat následující polohu. Tento algoritmus se při sledování objektu v obraze používá s nějakou doplňující metodou, která dokáže přesněji určit polohu sledovaného objektu.

Kapitola 4

Rozpoznávání obličeje

Obličej je taková část těla, která nám dává na první pohled znát, že daný člověk je unikát. V obličejí vidíme nejvíce rysů, které jsou známy na první pohled a pro identifikaci člověka jsou dostačující. Někdy však může být identifikace člověka podle obličeje dosti zavádějící, protože obličej dvou odlišných lidí si mohou být velmi podobné, například jednovaječná dvojčata jsou podle obličeje téměř nerozeznatelné.

Pro rozpoznávání obličejů z obrazu existuje celá řada metod ale v posledních letech se v počítačovém vidění velmi často používají konvoluční neuronové sítě, které dosahují v této oblasti velmi kvalitních výsledků.

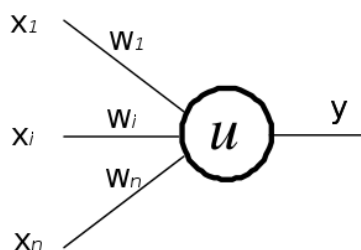
4.1 Neuronové sítě

Neuronové sítě jsou inspirovány biologickými neuronovými sítěmi živých organismů. Jedná se o výpočetní model používaný především v umělé inteligenci. Přední vlastností tohoto modelu je schopnost učit se. Skládá se z umělých neuronů, kdy neurony jsou vzájemně propojeny a transformují vstupní signály pomocí přenosových funkcí na výstup. Každý neuron může mít různý počet vstupů, ale pouze jeden výstup.

Je popsána celá řada modelů umělého neuronu, ovšem nejpoužívanějším je model popsaný W. McCullochem a W. Pittsem [22].

4.1.1 Obecný model umělého neuronu

Na následujícím obrázku 4.1 můžeme vidět obecný model umělého neuronu. Skládá se ze vstupů x_1-x_n , vstupních vah w_1-w_n a vnitřního prahu u , jehož pomocí se spočte výstup y .



Obrázek 4.1: Model umělého neuronu. Převzato z [22]

Výstup y se spočte pomocí funkce:

$$y = g(u) = g(f(\vec{x})), \quad (4.1)$$

kde f je bázová funkce, g je aktivační funkce a \vec{x} je vstupní vektor neuronu. Bázová a aktivační funkce se poté volí podle požadovaných vlastností a potřeb neuronové sítě.

4.1.2 Topologie

Jednotlivé umělé neurony se propojují do rozsáhlejších neuronových sítí, které dokáží řešit jak lineárně separovatelné problémy, tak i lineárně neseparovatelné problémy. Propojení neuronů se dá rozdělit do několika následujících skupin [22]:

- plně propojená síť,
- plně propojená symetrická síť,
- vrstvosvá síť,
- acyklická síť,
- dopředná síť.

4.1.3 Učení neuronové sítě

Učením neuronové sítě se myslí vhodné nastavení vah jednotlivých neuronů v síti. Učení neuronových sítí je možné provádět více způsoby, například pomocí křížové validace, adaptivního trénování nebo evolučních strategií. Nejpoužívanější metodou je však algoritmus backpropagation.

Backpropagation

Algoritmus backpropagation byl odvozen pro vícevrstvé dopředné neuronové sítě. Jeho úkolem je najít optimální nastavení vah neuronů tak, aby jejich velikosti byly z hlediska řešeného problému optimální. Hledá se globální minimum chybové funkce. Učení tedy nastavení vah probíhá v opačném směru, od výstupu ke vstupu. K učení vícevrstvé neuronové sítě se používá zejména sigmoidální aktivační funkce, definovaná vztahem:

$$y = \frac{1}{1 + e^{-x}}, \quad (4.2)$$

kde x je výstupní hodnota bázové funkce a y je výsledný výstup neuronu.

Při trénování neuronové sítě je vstupem algoritmu backpropagation trénovací množina T . Každému i -tému prvku množiny T odpovídá dvojice (\vec{x}_i, \vec{d}_i) , kde \vec{x}_i je vstupním vektorem neuronové sítě a \vec{d}_i je požadovaným výstupním vektorem neuronové sítě.

Před tím, než se může neuronová síť začít učit, musíme ji nejprve inicializovat. Při inicializaci sítě se jednotlivé váhy nastaví na náhodnou hodnotu z intervalu $\langle -0,5; 0,5 \rangle$ a globální chyba $GIErr$ se nastaví na 0 [22, 10]. Pro inicializaci sítě se také mohou použít jiné metody, například simulované žíhání nebo genetické algoritmy.

Učení probíhá v cyklech. Pro každý prvek z množiny T se vypočítá odezva sítě pro vstupní vektor. Vypočítaný vektor se porovná s vektorem požadovaným, čímž se určí lokální chyba sítě E . Následně se určí globální chyba pomocí vzorce [22]:

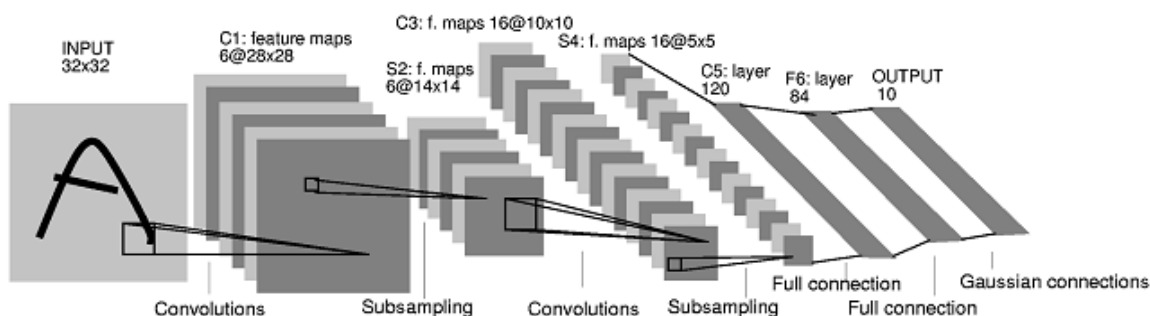
$$GErr = GErr + 0,5 * E \quad (4.3)$$

Učení sítě se opakuje tak dlouho, dokud není splněna ukončovací podmínka, která může být například dosažení požadované přesnosti nebo překročení časového limitu [10, 22].

4.2 Konvoluční neuronové sítě

Poprvé navrhl konvoluční neuronovou síť K. Fukushima v roce 1980, kterou použil pro rozpoznávání vzorů v obraze [9]. Od té doby se tyto neuronové sítě stále zlepšovaly. V roce 2012 pak A. Krizhevsky vyhrál pomocí konvoluční neuronové sítě soutěž ILSVRC v rozpoznávání obličejů z obrazu [15].

Konvoluční neuronové sítě jsou speciálním typem dopředných neuronových sítí, které jsou určeny pro klasifikaci strukturovaných dat. Využívají několika druhů vrstev neuronů. Takovou obecnou konvoluční síť můžeme vidět na obrázku.



Obrázek 4.2: Ukázka konvoluční neuronové sítě. Převzato z [16]

Při zpracování obrazových dat je vhodné, aby odezva sítě nebyla závislá na posunu, měřítku nebo deformaci vstupních dat. Konvoluční sítě to relativně splňují díky svým vlastnostem. Je tomu tak především proto, že získávají příznaky z malých oblastí obrazu, sdílejí váhy neuronů a využívají vrstvu podvzorkování [16]. Učení konvolučních neuronových sítí se provádí pomocí algoritmu Backpropagation.

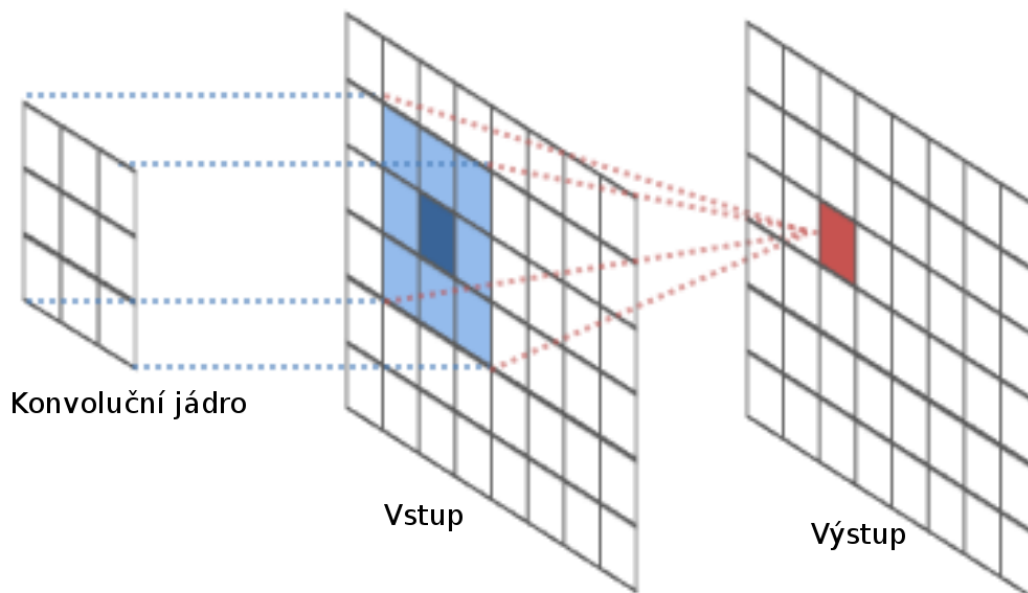
Konvoluční neuronové sítě hledají kvalitní příznaky pro rozpoznání obrazu. Postupně transformují reprezentaci obrazu na menší reprezentace, které lépe popisují informace o obsahu obrazu. Tohle vede k redukci dimenzionality vstupního obrazu a prostorové informace. V poslední fázi se pak vezme výstupní vektor nesoucí informace o obraze, který se následně předá klasifikátoru nebo regresoru.

4.2.1 Konvoluční vrstva

Významným prvkem konvolučních neuronových sítí je konvoluční vrstva. Každý neuron v této vrstvě je propojený s malým okolím v předcházející vrstvě. Je-li konvoluční vrstva vstupní vrstvou, pak je každý neuron této vrstvy propojen s malým okolím vstupního obrazu.

Neurony jsou v této vrstvě organizované do rovin, ve kterých všechny neurony sdílejí stejnou množinu vah [16]. Množina vstupů v této rovině se označuje jako příznaková mapa nebo také konvoluční jádro. Sdílení vah omezuje celkový počet výpočtů při učení neuronové sítě, čímž se snižuje výpočetní a paměťová náročnost. Neurony této vrstvy počítají především operaci nazývanou *konvoluce*.

Konvoluční vrstva je tedy složena sadou příznakových map a může být napojena na vstupní vrstvu nebo na vrstvu podvzorkování. Úkolem příznakových map je získat lokální charakteristiky obrazu pomocí konvoluce, která funguje jako filtr obrazu. Pojmeme lokální je myšleno, že budeme uvažovat pouze malou oblast přiloženého obrazu. Na obrázku 4.3 můžeme vidět ukázkou příznakové mapy..



Obrázek 4.3: Struktura systému. Převzato z [12]

Díky neuronům konvoluční vrstvy je možné z obrázku získat základní příznaky, jako jsou hrany, rohy apod. Příznaky jsou pak v dalších vrstvách vzájemně zkombinované, čímž se získávají příznaky na vyšší úrovni. Odpadá tedy složité předzpracování a získávání příznaků.

Dále využijeme techniky sdílení vah. Tento neuron zkopírujeme pro každý podobraz dané velikosti. Všechny takto vzniklé neurony sdílejí váhy včetně prahu. Při rozpoznávání obrazů je vhodné rozpoznávat více příznaků, proto bývá příznakových map použito více.

Každá příznaková mapa sdílí váhy svých neuronů, ovšem mezi sebou nesdílí žádnou.

4.2.2 Vrstva podvzorkování

Vrstva podvzorkování (Max-Pooling) slouží ke zmenšení velikosti vstupního obrazu a pracuje nezávisle na barevné hloubce. Velikost obrazu je transformována pomocí funkce *max/mean*, která je aplikována na nepřekrývající se čtvercové bloky vstupního obrazu [12]. Výstupem této vrstvy je zmenšený obraz, jehož velikost se spočte následovně:

$$W = (W_{in} - F) / S + 1 \quad (4.4)$$

$$, H = (H_{in} - F) / S + 1 \quad (4.5)$$

kde $W_{in} \times H_{in}$ je vstupní rozměr obrazu, $W \times H$ je výstupní rozměr obrazu, F je velikost čtvercové oblasti a S je krok.

Vrstva podvzorkování výrazně snižuje výpočetní náročnost konvoluční neuronové sítě.

4.2.3 Techniky používané v konvolučních neuronových sítích

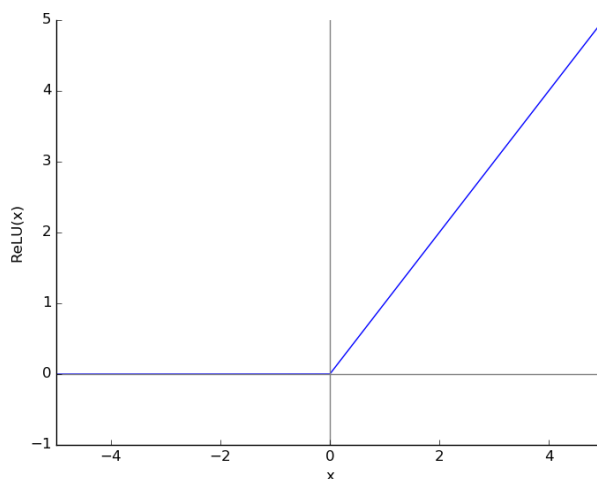
Ke zvýšení efektivity a ke zkrácení doby trénování konvolučních neuronových sítí se používají různé metody. Mezi nejpoužívanější techniky patří data augmentation, **Rectified Linear Unit** a dropout.

ReLU představuje zdroj nelinearity u konvolučních neuronových sítí. Jedná se o aktivační funkci, která je definována jako:

$$\text{ReLU}(x) = \max(0, x) \quad (4.6)$$

Funkce f pracuje s výstupem bázové funkce, tedy s váhovým součtem vstupů neuronu. Záporné hodnoty se vyhodnocují jako nulová hodnota, kdežto nezáporné hodnoty zůstávají nezměněny.

Konvoluční neuronové sítě, které používají ReLU, se trénují rychleji a je tak možné pracovat s většími sítěmi. Tato metoda také zabraňuje saturaci, tudíž není nutně potřebná normalizace vstupů neuronu, ale i přesto se v některých případech používá.



Obrázek 4.4: Aktivační funkce ReLU.

Data augmentation je metoda, která uměle navyšuje velikost trénovacího datasetu. Navyšení datasetu snižuje možnost přetrénování neuronové sítě. Tato metoda využívá transformace obrazu, jako je například oříznutí, natočení nebo převrácení obrazu.

Metoda *dropout* spočívá v nastavení nulového výstupu u neuronů skrytých vrstev s pravděpodobností 0,5. Jelikož neurony, které mají nulový výstup se nepodílejí na učení neuronové sítě, dojde tak ke zrychlení učení. Sítě využívající tuto metodu mění pro každý vstup svou architekturu.

4.3 Klasifikace

Klasifikace je zobrazení z množiny objektů do množiny předem definovaných tříd. Klasifikátor je algoritmus provádějící klasifikaci. Algoritmy mohou být jednoduché a klasifikovat na základě definovaných pravidel, nebo složitější a využívat mechanismů strojového učení. Výkonost klasifikátorů se posuzuje dle úspěšnosti rozdělení testovací množiny do definovaných tříd.

4.3.1 KNN klasifikátor

K Nearest Neighbours je klasifikátor založený na porovnávání vzdáleností. Natrénování KNN spočívá ve vhodném rozdělení trénovacích dat do prostoru. Při klasifikaci je vektor umístěn do prostoru a na základě trénovacích dat se vybere k nejbližších sousedů. Vektor je pak zařazen do kategorie na základě takto vybraných sousedů. Parametr k je zadán manuálně a musí splňovat podmínku $1 \leq k$.

Vzdálenost je dána jako suma všech vzdáleností od testovaného vektoru k jeho k nejbližším sousedům. Pro výpočet vzdálenosti se nejčastěji používá Euklidovská nebo Manhatanská metrika.

Euklidovská metrika vzdálenosti vektorů A a B:

$$M_E(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (4.7)$$

Manhatanská metrika vzdálenosti vektorů A a B:

$$M_E(A, B) = \sum_{i=1}^n |a_i - b_i| \quad (4.8)$$

Objekt je následně klasifikován do třídy, do které patří většina z nejbližších sousedů.

4.3.2 Klasifikátor SVM

Support Vector Machine je algoritmus strojového učení, který se používá jak ke klasifikaci, tak k regresi. V úloze klasifikace hledá nadrovinu, která optimálně rozděluje trénovací data do dvou kategorií. Snahou je získat nadrovinu, která je od jednotlivých kategorií co nejvíce vzdálena. Hlavním problémem této metody je nalezení optimální nadroviny, která správně odděluje kategorie a přitom je od kategorií maximálně vzdálena. Čím více je vzdálena, tím získáme lepší výsledky při testování. Vektorům, jenž jsou nejbližší nadrovině, se říká support vectors. Trénovací data není obvykle snadné lineárně rozdělit. Místo toho, aby se použila nějaká nelineární křivka, namapují se trénovací data do vyšší dimenze. Díky tomu se data stanou lineárně separovatelné.

Ke zjednodušení klasifikačního problému slouží několik různých jader, kupříkladu *radial basis function* nebo sigmoidální funkce. Každé jádro má zpravidla několik parametrů, které je nutné správně nastavit pro trénovací data. Nejspolehlivější metodou pro nastavení těchto parametrů je, že se pro každou kombinaci parametrů vytvoří model a použije se ten nejlepší, což může být dosti časově náročné.

SVM lze modifikovat tak, aby jej bylo možné použít pro klasifikaci vícedimenzionálních dat. Jednou z metod, jak klasifikovat do více tříd, je použít pro každou třídu oddělený klasifikátor, který bude klasifikovat, zda vzorek do dané třídy patří, či nikoliv. Spojením těchto klasifikátorů pak lze klasifikovat do každé třídy zvlášť.

4.3.3 Nearest centroid

Metoda Nearest Centroid je klasifikační model, který přiřazuje hledanou třídu k přiloženému vektoru, na základě trénovací množiny. Trénovací množina klasifikátoru je definována jako:

$$T = \{(\vec{x}_1, \vec{y}_1), \dots, (\vec{x}_n, \vec{y}_n)\}, x_i \in X, y_i \in Y, i = 1 \dots n \quad (4.9)$$

,

kde T je trénovací množina, x_i je vektor charakterizující třídu y_i , X je množina vektorů a Y je množina všech tříd. Při trénování klasifikátoru se spočte centroid všech přiložených vektorů, pomocí následujícího vzorce:

$$\vec{\mu}_l = \frac{1}{|C_l|} \sum_{i \in C_l} \vec{x}_i, \quad (4.10)$$

kde C_l je množina všech vektorů náležící třídě $l \in Y$.

Při klasifikaci je vektor \vec{x} klasifikován do třídy y následovně:

$$y = \operatorname{argmin}_{l \in Y} \|\vec{\mu}_l - \vec{x}\| \quad (4.11)$$

Při implementaci a průběžném testování klasifikačních metod bylo zjištěno, že tato nevykazuje kvalitní výsledky klasifikace extrahovaného vektoru rysů. Pokud bude v databázi osob u některé osoby větší množství obličejů dojde s velkou pravděpodobností k chybné predikci. Na základě získaných znalostí z experimentů s extrahovaným vektorem rysů z obrazu obličeje, byla tato metoda upravena aby dosahovala lepších výsledků při klasifikaci vektoru rysů.

Úprava klasifikace spočívá v nalezení lokálních centroidů pro každou osobu v databázi, čímž odpadá trénování původního klasifikátoru. A následnou klasifikaci vektoru (\vec{x}), tedy nalezení nejbližšího centroidu k takto vytvořeným lokálním centroidům.

Z trénovací množiny se hledají lokální centroidy zvláště pro $\forall t_l \in T$, kde $l \in Y$, pomocí shlukové analýzy, metodou DBSCAN, která je popsána v podkapitole 4.4.

4.3.4 Metriky pro hodnocení klasifikátorů

Existence celé řady metodik pro testování klasifikátorů může mít za následek obtížně porovnatelné výsledky. Větší část našeho systému je biometrickým systémem, jelikož se snažíme určit identitu osoby na záznamu. Některé metriky pro měření úspěšnosti systému budou právě proto orientovány na testování kvality biometrických systémů, což je mírně odlišné od obyčejné klasifikace. U biometrických systémů hraje velkou roli skóre porovnání dvou vzorků, které taktéž může být chápáno jako metrika vzdálenosti testovacího vzorku od klasifikované třídy, také nazýváno jako míra příslušnosti. U běžné klasifikace se většinou s tímto skóre dále nepracuje.

Aby bylo možné vyhodnotit úspěšnost klasifikace, je nutné rozdělit datovou sadu na trénovací a testovací vzorky. Existuje mnoho způsobů rozdělení datové sady, například náhodným výběrem, metodou *bootstrap* nebo metodou *leave one out*. Při testování klasifikátoru je nutné, aby byly definovány třídy, do kterých má být testovací vzorek zařazen. Pak je možné srovnat zařazení testovacích vzorků klasifikátorem do dané třídy se skutečností. Aby bylo možné kvantifikovat výsledky testování, je nutno zavést metriky charakterizující klasifikátor. Tyto metriky jsou uvedeny a popsány tabulkou 4.1.

Zkratka	Název veličiny	Popis
TP	True positive	vzorek A byl přijat jako vzorek A
TN	True negative	vzorek A byl odmítnut jako vzorek B
FP	False positive	vzorek A byl přijat jako vzorek B
FN	False negative	vzorek A byl odmítnut jako vzorek B

Tabulka 4.1: Popis použitých metrik. Převzato z [8]

Z metrik v tabulce 4.1 lze snadno vypočítat TPR(True Positive Rate) a FPR(False Positive Rate). TPR měří podíl správných klasifikací ku očekávanému počtu správných klasifikací a FPR měří podíl správných odmítnutí vzorků ku očekávanému počtu správně odmítnutých vzorků. TPR a FPR jsou dány následujícími vzorci:

$$TPR = TP / (TP + FN) \quad (4.12)$$

$$FPR = FP / (FP + TN) \quad (4.13)$$

Další možností popisu úspěšnosti klasifikátoru jsou poměrové veličiny FAR, neboli míra chybného přijetí, a FRR, neboli míra chybného odmítnutí. Tyto veličiny se často používají pro hodnocení biometrických systémů a jsou dány následujícími vztahy:

$$FAR = \frac{\text{Počet porovnání rozdílných vzorů vedoucích ke shodě}}{\text{Celkový počet porovnání rozdílných vzorů}} \quad (4.14)$$

$$FRR = \frac{\text{Počet porovnání shodných vzorů vedoucích k neshodě}}{\text{Celkový počet porovnání shodných vzorů}} \quad (4.15)$$

Nejvíce používanou metodou pro provonání klasifikátorů je křivka ROC(Receive Operating Characteristics). ROC křivka (obr. 4.5) nejčastěji reprezentuje závislost FAR na FRR nebo TPR na FPR se změnou prahu T , proto je možné tyto křivky sesumarizovat do jedné. Obecně lze touto křivkou znázornit závislost i na jiných měřích při nastavení prahu T . Práh T udává minimální skóre vzorku od klasifikované třídy. Je-li skóre menší než práh T , pak je testovací vzorek klasifikován do dané třídy, v opačném případě je testovací vzorek odmítnut.

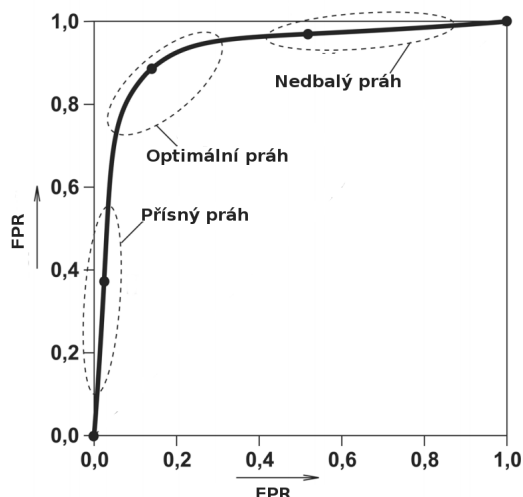
Pomocí výše popsaných metrik lze hodnotit pouze binární klasifikace. Jedná se o klasifikaci pouze do dvou tříd. Ovšem v našem případě je potřeba vyhodnotit kvalitu identifikace osob, což je proces klasifikace do více tříd. V další části se proto zaměříme na metriky testování identifikace biometrických systémů.

Pro vyhodnocení míry úspěšnosti identifikace osob je definována metrika IR(identification rate) [13].

$$IR = \frac{\text{Počet správně identifikovaných osob při zvoleném rank-}t}{N} \quad (4.16)$$

kde $rank-t$ je počet osob vrácených klasifikátorem a N je velikost testovací množiny. Parametr $rank-t$ není třeba uvádět, pokud $rank-t = 1$.

IR může být sumarizována pro rozdílné hodnoty rank-t pomocí křivky CMC (Cumulative Match Characteristic). Tato křivka pak reprezentuje závislost přesnosti identifikace na počtu vrácených osob klasifikátorem.



Obrázek 4.5: Ukázka ROC křivky pro měření úspěšnosti klasifikace.

Taktéž se zavádí míry FPIR(false positive identification rate) a FNIR(false negative identification rate), které jsou definovány následovně:

$$FPIR = \frac{\text{Počet chybně identifikovaných osob jako známá osoba při zvoleném rank-}t}{\text{Celkový počet neznámých osob}} \quad (4.17)$$

$$FNIR = \frac{\text{Počet chybně identifikovaných osob při zvoleném rank-}t}{\text{Celkový počet známých osob}} \quad (4.18)$$

Porovnáním FPIR a FNIR při zvoleném prahu T je získána informace o úspěšnosti klasifikace při zvoleném prahu T , který udává minimální vzdálenost vzorku od klasifikované třídy. Metriky FPIR resp. FNIR jsou téměř totožné s metrikami FAR resp. FRR, proto lze taktéž použít křivku ROC k jejich porovnání.

4.4 Shluková analýza

Shluková analýza je podobná klasifikaci. Taktéž jsou zde analyzovaná data seskupována do tříd. Na rozdíl od klasifikace předem neznáme třídu do které budou data seskupena. Shlukování, neboli shluková analýza, je proces, při kterém jsou data seskupována do skupin, které se nazývají shluky. Data uvnitř shluku jsou si velmi podobná a data mezi shluky jsou naopak co nejvíce rozdílné. Díky shlukové analýze je možné detekovat a eliminovat odlehlé hodnoty. V této práci je shluková analýza použita k eliminaci odlehlých vektorů extrahovaných z obrazu obličeje a ke zpřesnění klasifikace.

4.4.1 DBSCAN

Celým jménem *Density-Based Clustering Method Based on Connected Regions with Sufficiently High Density* je shlukovací metoda, jejíž shluky jsou tvořeny maximálními množinami bodů spojených na základě hustoty. Metody shlukové analýzy založené na hustotě

tvoří shluky s velkou hustotou objektů oddělené oblastmi s malou hustotou objektů. Tato metoda je schopna objevovat shluky libovolného tvaru v prostorových databázích a odfiltrout zašumělé nebo odlehlé hodnoty.

Metoda DBSCAN je definována následujícími pojmy:

- ϵ -okolí je okolí objektu o poloměru ϵ ,
- jádro je objekt, jehož ϵ -okolí obsahuje minimálně $MinPts$ objektů,
- přímá dosažitelnost objektu p je přímo dosažitelná na základě hustoty z objektu q , v množině objektů D , jestliže p se nachází v ϵ -okolí objektu q a q je jádro,
- dosažitelnost objektu p je dosažitelná na základě hustoty z objektu q v množině objektů D , existuje-li řetězec objektů p_1, \dots, p_n , kde $p_1 = q$ a $p_n = p$, takový, že objekt p_{i+1} je přímo dosažitelný na základě hustoty z objektu p_i , pro $p_i \in D$, $1 \leq i \leq n$,
- spojení objektů objekt p je spojený na základě hustoty s objektem q , v množině objektů D , existuje-li objekt $o \in D$ takový, že objekty p a q jsou dosažitelné na základě hustoty z objektu o .

Z této definice vyplývá, že dosažitelnost na základě hustoty je tranzitivním uzávěrem relace přímé dosažitelnosti na základě hustoty. Algoritmus DBSCAN nejprve prohledá databázi, nalezne jádra a vytvoří příslušné shluky. Následně algoritmus hledá objekty, které jsou přímo dosažitelné z jader shluků a to tak dlouho, dokud již není možné připojit žádný objekt k žádnému shluku.

Kapitola 5

Implementace

Tato kapitola popisuje praktickou část práce, kde je předložen návrh, postup vývoje a implementační detaily systému pro sledování a rozpoznávání lidí na videu. Implementace probíhala současně s některými experimenty, aby bylo možné použít metody a algoritmy, které dosahují nejlepších výsledků. Veškeré používané postupy a algoritmy jsou popsány předcházejícími kapitolami.

Úspěšnost vytvořeného systému pro sledování lidí na videu je ovlivněna umístěním bezpečnostní kamery. Kamera by měla být umístěna tak, aby byly zachyceny obličeje procházejících osob a nacházeli se v záběru co nejdéle to bude možné. Také by se neměli vyskytovat v záběru kamery žádné pohybující se objekty jiné než lidé, což může mít negativní dopad na sledování osob.

5.1 Požadavky

Systém pro sledování a rozpoznávání lidí ve videu je implementován v jazyku *Python 2.7* s použitím knihoven *OpenCv 3.1.0*¹, *dlib*², *OpenFace*³ a frameworku *torch*⁴. Systém bude spuštěn pod operačním systémem GNU/Linux s grafickou kartou NVIDIA s hardwarovou a softwarovou podporou architektury CUDA. Požadavek na přítomnost grafické karty NVIDIA není striktní, systém lze spustit i bez této grafické karty, ale za cenu snížení výkonnosti systému. Existenci potřebných knihoven, výjma frameworku *torch*, lze ověřit spuštěním přiloženého skriptu `requirements.py`.

5.2 Struktura systému

Mimo rozpoznávání bylo také implementováno sledování trajektorie pohybujících se osob na záznamu, rozpoznání pohlaví, evidence průchodů a oznámení výskytu neznámých nebo předem definovaných osob. Při implementaci byl kladen důraz na přesnost použitých metod spolu s jejich časovou složitostí. Výběr vhodného klasifikátoru byl ovlivněn možností aktualizace za běhu systému tak, aby bylo možné upravovat množinu rysů osob a přidávat nově detekované osoby.

¹Knihovna *OpenCv 3.1.0* dostupná z url: <http://opencv.org/>.

²Knihovna *dlib* dostupná z url: <http://dlib.net/>.

³Knihovna *OpenFace* dostupná z url: <http://github.com/cmusatyalab/openface>.

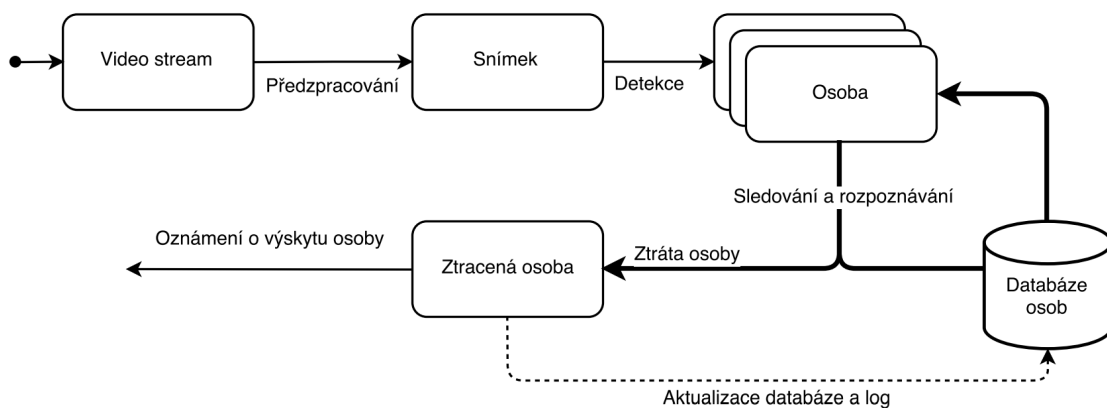
⁴Framework *torch* dostupný z url: <http://torch.ch/>

Do systému byla implementována možnost oznámení o výskytu některé ze známých osob nebo o výskytu neznámé osoby. Také je možné v rámci systému ukládat trajektorie pohybujících se osob. Jedná se pouze o trajektorii ve 2D prostoru. Díky implementaci uchování informace o trajektorii je možné také například počítat průchody chodbou nebo trajektorie k dalším analýzám.

V rámci implementace bylo také vytvořeno rozhraní pro testování systému, díky kterému je možné ověřit přesnost rozpoznávání a popřípadě upravit některé parametry systému, aby bylo dosaženo lepší přesnosti.

Na obrázku 5.1 můžeme vidět zjednodušený model systému. Systém dovoluje zpracovávat jak videozáznam, tak přímý stream ze záznamového zařízení. Ovšem pro některé zařízení musí být vytvořen patřičný konektor, který propojí záznamové zařízení s tímto systémem, čímž je zajištěna možnost připojit k tomuto systému jakékoliv záznamové zařízení.

Při návrhu struktury musela být zohledněna řada faktorů, které by mohly mít negativní dopad na časovou složitost zpracovávání snímků. Aby bylo dosaženo co nevyšší rychlosti, musely být některé části systému paralelizovány, což je detailněji popsáno v kapitole 5.3. Implementovaný systém také dokáže zpracovávat více videí současně. K tomu je ale zapotřebí dostatečně velký výpočetní výkon.



Obrázek 5.1: Struktura systému.

Jak můžeme vidět na obrázku 5.1, nejprve je každý snímek předzpracován a následně jsou na takto předzpracovaném snímku detekovány osoby. Detekované osoby jsou následně sledovány napříč videozáznamem a to tak dlouho, dokud se vyskytují v sekvenci po sobě jdoucích snímků. Detekované a sledované osoby jsou následně rozpoznány na základě existující databáze osob. Opustí-li sledovaná osoba scénu nebo dojde k ukončení sledování osoby, uloží se informace o výskytu osoby a aktualizuje se vektor rysů v databázi. Aktualizace vektoru rysů nastane jen v případě, jsou-li extrahované vektory rysů dostatečně kvalitní. Jestliže je aktivní oznámení o výskytu neznámé osoby nebo některé z osob z databáze, odešle se emailová zpráva na nastavenou emailovou adresu s tímto upozorněním.

Implementovaný systém také umožňuje sledovat a rozpoznávat osoby z více zdrojů videa. Proces rozpoznávání zůstane stejný jako při zpracování z jednoho zdroje. Počet video zdrojů je omezen pouze výpočetním výkonem a dostupnou kapacitou operační paměti.

5.2.1 Předzpracování snímků

Aby byla detekce a sledování co nejpřesnější, je u každého vstupního snímku systému provedena korekce osvětlení, změna velikosti, extrakce pozadí a převod na šedotónový obraz. Korekce osvětlení není vždy potřeba, proto je možné ji vypnout.

Nejdříve se provede korekce osvětlení a to tak, že se na šedotónovém obrazu spočte průměrná intenzita obrazových bodů. Tato hodnota je pak porovnána se střední hodnotou maximální intenzity a na základě tohoto porovnání je pak upravena intenzita všech pixelů obrazu.

Pro extrakci pozadí stačí velmi malý obraz, díky čemuž se proces extrakce mnohonásobně zrychlí. Na zmenšeném obraze se vyhladí šum, čímž se zabrání chybné detekci velmi malých pohybujících se objektů. Následně je z extrahovaného pozadí vytvořena ořezová maska nesoucí informace o pohybujících se objektech. Velikost masky je pak transformována zpět na původní velikost obrazu. Masky je použita pro detekci osob a obličejů.

5.2.2 Detekce

Detekce obličejů je prováděna pomocí ořezové masky na obraze ve stupních šedi. Masky byla vytvořena na základě extrakce pozadí, tedy detekce probíhá pouze v místech obrazu, kde byl detekován pohyb. Obličejů se detekují pomocí algoritmu Viola&Jones, který je obsažen v knihovně OpenCV. Výstupem je pak seznam čtverců, kde každý čtverec obsahuje právě jeden detekovaný obličej.

Parametry detektoru Viola&Jones jsou nastaveny tak, aby byla detekce rychlá, přičemž se také zvýší nepřesnost detektoru a jsou detekovány i objekty, které nejsou obličejů. Tato nepřesnost je následně odstraněna opětovnou detekcí již detekovaného objektu s přísnějším nastavením parametrů detektoru a detekcí klíčových bodů obličejů jako je nos, oči nebo obrys obličejů. Opětovná detekce se provádí pouze na nově detekovaných obličejích. Díky tomuto postupu je detekce obličejů několikrát rychlejší, než kdyby se prováděla pouze detekce s jedním přísným detektorem.

Detekce těla osoby je implementována pomocí detektoru HOG, který je taktéž součástí knihovny OpenCV. Jedná se o časově náročnou operaci, proto je tato detekce prováděna na zmenšeném obraze, na kterém je aplikována maska detekovaných pohybujících se objektů. Parametry tohoto detektoru jsou optimalizovány tak, aby byla časová složitost co nejmenší a přesnost co největší. Ovšem větší důraz je kladen na časovou složitost. Těla osob jsou pro určení identity irelevantní, proto je možné tuto detekci v systému vypnout. Těla osob jsou použity pouze ke zpřesnění následného sledování, jelikož se jedná o daleko větší objekt než je obličej. V některých situacích je tato detekce přítěží, například pokud se na scéně pohybují i jiné objekty než lidé (dveře, dopravní prostředky, atd.), což lze do jisté míry ovlivnit nastavením parametrů této metody. Zvýšení přesnosti detekce vede ke zvednutí nároků na výpočetní výkon.

5.2.3 Sledování

Detekované objekty jsou sledovány napříč videozáznamem. Není pak nutné provádět na každém snímku opakované rozpoznávání osob, z čehož vyplývá snížení výpočetní náročnosti, ale také otevírá mnoho dalších možností využití tohoto systému. Sledování objektů je opět implementováno tak, aby byla časová složitost co nejmenší při zachování ucházející přesnosti. Proto bylo sledování sestaveno z více algoritmů, které jsou používány velmi úsporně.

Sledování objektů je sestaveno z následujících algoritmů:

- KLT tracker,
- Kalmanův filtr,
- Teplate matching.

Proces sledování pak probíhá v několika krocích. U nově detekovaných objektů se nejprve inicializují obrazové body ke sledování pomocí algoritmu KLT tacker. Inicializuje se Kalmanův filtr na střed sledovaného objektu a připraví se šablona ke sledování. Veškeré použité algoritmy pracují s obrazem ve stupních šedi.

Sledování již inicializovaných objektů probíhá tak, že se na aktuálním snímku pomocí algoritmu KLT určí posunutí sledovaných bodů pro každou osobu. U každé sledované osoby se určí střed napříč všemi sledovanými body. Pokud se některé body v průběhu sledování příliš vzdálí od ostatních, jsou pak znovu inicializovány, nebo jsou odstraněny. V okolí těchto bodů a v okolí predikce Kalmanova filtru se pomocí algoritmu *template matching* hledá šablona sledovaného objektu. Šablona objektu je v průběhu sledování aktualizována. Nalezne-li algoritmus *template matching* shodu, pak je opravena pozice Kalmanova filtru a jsou mírně upraveny sledované body. Nenalezne-li žádnou shodu, je pozice Kalmanova fitru upravena až po nějakém čase, nebo až je predikovaná hodnota mimo obraz.

Tato posloupnost použití sledovacích algoritmů zajišťuje, že je možné sledovat objekt, pokud je na velmi krátký čas překryt jiným objektem. Tento jev není zajištěn použitím algoritmů samostatně, ale jejich kombinací. Proces sledování můžeme vidět na obrázku 5.2, kde lze také pozorovat zvyšující se počet extrahovaných obličejů.



Obrázek 5.2: Ukázka sledování osoby.

Sledované objekty jsou odečteny od detekovaných. Tím je získána množina nově vyskytujících se osob. Tento rozdíl je implementován pomocí výpočtu procentuálního protnutí sledovaných a detekovaných objektů. Taktéž je tímto rozdílem vypočtena a aktualizována množina přesnějších pozice sledovaných objektů.

Implementované sledování není příliš kvalitní, a proto bylo rozšířeno. Nově detekované obličeje osob jsou porovnávány se sledovanými také pomocí extrahovaných vektorů, což zkvalitňuje výsledné sledování. Taktéž pokud je sledovaný objekt ztracen, zůstává ještě nějakou dobu uchován. Objeví-li se v této době obličej znovu v sekvenci snímků, je opět přidán

ke sledování bez ztráty získaných informací z předešlého sledování. Proces porovnávání rysů dvou osob se v biometrii označuje jako verifikace. V kapitole 6 je proces verifikace patřičně otestován. Díky těmto rozšířením je sledování dostatečně robustní.

5.2.4 Rozpoznávání obličeje

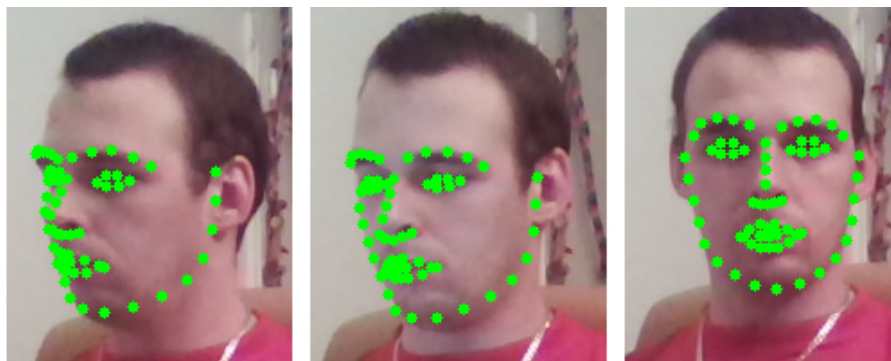
Určení identity osoby z videozáznamu se provádí ve dvou krocích. Nejprve je extrahován vektor rysů pomocí konvoluční neuronové sítě z přiloženého snímku obličeje a následně je tento vektor klasifikován.

Rozpoznávání osob podle obličeje z videozáznamu je velmi obtížné, detekovaný obličej má velikost řádově desítek obrazových bodů, což velmi komplikuje proces rozpoznávání. Ve většině případů se tento typ identifikace provádí na obrazech o velikosti řádově stovky obrazových bodů.

V rámci této práce bylo vytvořeno několik modelů konvoluční neuronové sítě pro extrakci rysů obličeje z obrazu, ale tyto modely dosahovaly nízké úspěšnosti (cca. 60%). Tato nízká úspěšnost byla zaviněna malou trénovací datovou sadou a nízkým výpočetním výkonem a malou operační pamětí. Proto byla k vytvoření konvoluční neuronové sítě použita knihovna OpenFace, která dokáže i z velmi malého obrazu extrahovat silný vektor rysů. Model konvoluční neuronové sítě, který je obsažen v této knihovně, byl trénován na datové sadě o velikosti asi 500000 vzorků za použití grafické karty Tesla K40 několik dnů.

Knihovna obsahuje předtrénovaný model konvoluční neuronové sítě pro extrakci rysů obličeje z přiloženého obrazu, který je implementován pomocí frameworku *torch*. Model je možné dotrénovat na vlastní datové sadě nebo jej jakkoliv modifikovat.

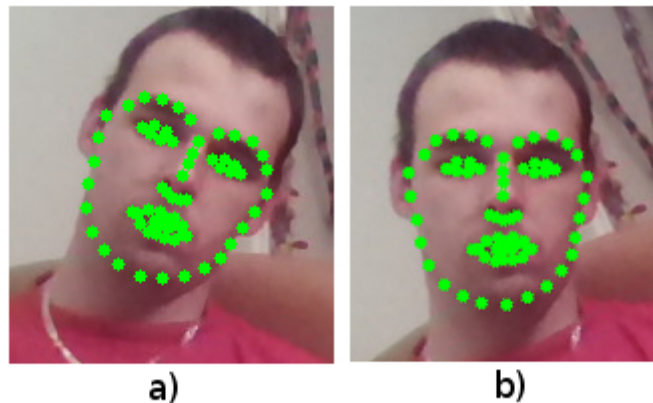
Před extrakcí rysů je nejprve potřeba upravit velikost vstupního obrazu na 91×91 px a zarovnat každý obličej tak, aby se nacházel ve středu obrazu a byl co nejvíce natočen čelně. Takové zarovnání je prováděno pomocí detekce klíčových bodů obličeje, jako jsou oči, nos, ústa a obrys obličeje, jak můžeme vidět na obrázku 5.3.



Obrázek 5.3: Detekce klíčových bodů obličeje.

Nalazení klíčových bodů obličeje se provádí pomocí konvoluční neuronové sítě, jejíž předtrénovaný model byl použit z knihovny *dlib*. Detekce těchto bodů je velmi přesná. Pokusíme-li se detekovat tyto body na obraze bez obličeje, nebudou žádné body detekovány. Díky této vlastnosti je odstraněna nepřesnost detektoru Viola&Jones. Detekované body jsou následně použity pro zarovnání obličeje tak, aby se každý obličej nacházel přesně ve středu. Zarovnání používá detekované středy očí a střed brady (obr. 5.4).

Vstupem konvoluční neuronové sítě je zarovnaný obraz obličeje o velikosti 91×91 px a výstupem je vektor rysů ve 128dimenzionálním prostoru. Extrahované rysy pak dostatečně



Obrázek 5.4: Zarovnání obličeje na základě detekce klíčových bodů. Na obrázku a) je nezarovnaný obličej s detekovanými klíčovými body obličeje a na obrázku b) je zarovnaný obličej

přesně charakterizují vstupní obraz obličeje, jak je později ukázáno v kapitole 6.

Existuje-li neprázdná databáze obličejů, je možné na základě extrahovaného vektoru rysů určit identitu osoby. Databáze osob je implementována pomocí *MongoDb*. Jedná se o NoSQL dokumentově orientovanou databázi, jejíž jeden záznam sestává z dvojice klíč-hodnota zapsaných ve formátu JSON. Hodnotou může být nejenom základní datový typ, ale i pole hodnot. Také lze jednotlivé dokumenty do sebe vnořovat. *MongoDB* si zakládá na třech klíčových vlastnostech a to: vysoký výkon, vysoká dostupnost a automatická horizontální škálovatelnost databáze.

Identita osob lze určit například pomocí klasifikátoru SVM, *K-nearest neighbors*, nalezení nejbližšího centroidu nebo jednoduchým porovnáním euklidovské vzdálenosti extrahovaného vektoru s vektory z databáze. Žádanou vlastností klasifikátoru je možnost dotrénovat klasifikátor za běhu systému. Dalším požadavkem je rychlost nalezení identity na základě přiloženého vektoru rysů. Vzhledem k tomuto požadavku a porovnáním klasifikátorů v úloze nalezení identity k extrahovanému vektoru rysů byl implementován klasifikátor hledající nejbližší lokální centroid napříč databází osob.

Metoda nalezení nejbližšího lokálního centroidu rozděljuje vektory rysů extrahované z obrazu obličeje do shluků pocházející od stejné osoby přidávané do databáze na základě shlukové analýzy DBSCAN, která byla popsána v podkapitole 4.4. Tato shluková analýza navíc odstraňuje odlehlé vektory což snižuje pravděpodobnost chybné klasifikace. Vektory uvnitř shluků jsou si podobné a proto je možné nad těmito vektory spočítat lokální centroid.

Z takto vytvořených shluků jsou vypočteny lokální centroidy pomocí kterých je následně určena identita osoby porovnáním euklidovské vzdálenosti vektoru rysů napříč všemi lokálními centroidy všech osob. Extrahovaný vektor rysů je klasifikován na základě nalezení nejbližšího lokálního centroidu. Centroid, kterému je vektor rysů nejbližší, pak náleží stejné osobě jako vektor rysů.

Vzdálenost mezi vektory a centroidy je definována euklidovskou metrikou. Tato metoda klasifikace byla vytvořena na základě získaných poznatků o extrahovaném vektoru rysů a prováděných experimentů při návrhu systému.

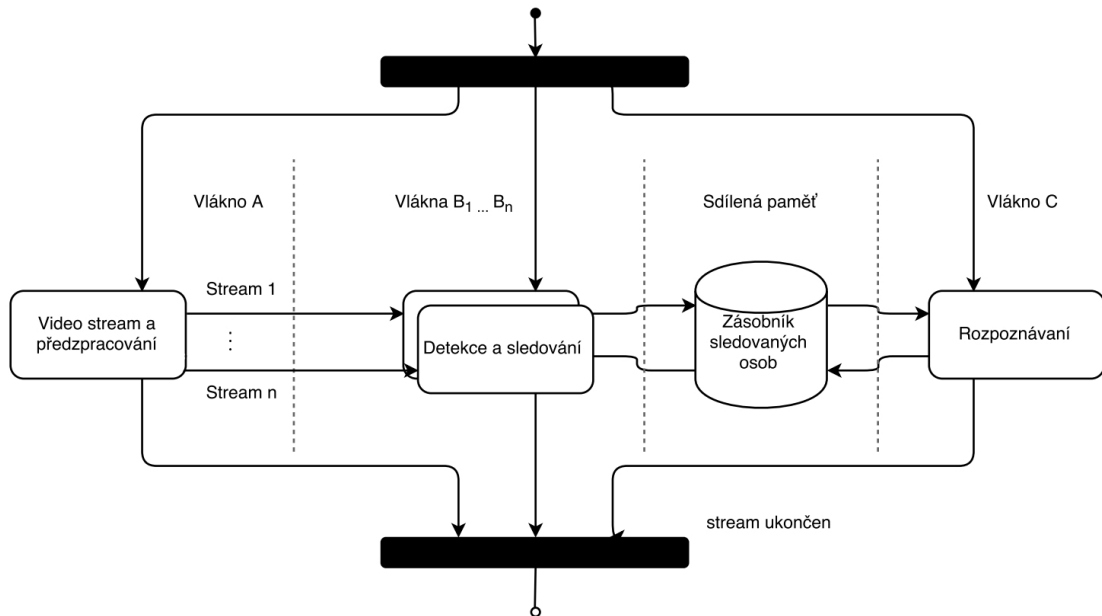
Z videa je ale získáno více obličejů jedné osoby, tudíž i klasifikace proběhne vícekrát. Ze všech takto získaných identit je zapotřebí vybrat pouze jednu. Výběr te správné identity, která náleží osobě na videu je prováděn tak, že se vybere ta, která se vyskytuje ze všech

nejvíce. Takový výběr zpřesňuje úspěšnost rozpoznání osoby, jak je ukázáno v kapitole 6, spolu s úspěšností výše popsaných klasifikátorů.

V rámci prováděných experimentů bylo zjištěno, že je možné dle extrahovaného vektoru rysů určit pohlaví osob na záznamu pomocí klasifikátoru KNN. Tato funkcionality byla implementována do systému, který také dokáže s přesností cca. 97% určit pohlaví detekované osoby. Při vkládání vektoru rysů do databáze je pro každý vkládaný vektor určeno pohlaví. Pohlaví osoby je pak získáno jako nejčastěji se vyskytující hodnota.

5.3 Paralelní zpracování

Proces rozpoznávání osob je relativně časově náročnou operací. Není možné provádět veškeré úkony jako je detekce, sledování a rozpoznávání osob na každém snímku sekvenčně. Proto jsou tyto úkony rozděleny na procesy, které se provádí paralelně. Detekce osob a jejich sledování jsou procesy, které jsou úzce spjaty. Jak již bylo řečeno, detekované osoby mohou být již sledovány a je potřeba zjistit, které osoby jsou detekovány nově, aby mohly být přidány ke sledování. Princip činnosti vytvořených vláken je znázorněn na obrázku 5.5.



Obrázek 5.5: Diagram komunikace vláken systému.

Počet všech vláken je $n + 2$, kde n je počet všech zpracovávaných videí. Vlákno A postupně předzpracovává snímky z každého n -tého videa a posílá je příslušnému vláknu B_n .

Každé vlákno $B_{1...n}$ využívá ke své práci zásobník (FIFO) určený pro krátkodobé skladování jednotlivých snímků ze záznamu, což vyžaduje dostatečnou velikost operační paměti. Snímky jsou pak postupně odebírány a provádí se na nich detekce a sledování osob. Zásobník má omezenou kapacitu na 70 snímků. Pokud dojde k jeho naplnění, tak se čeká, dokud není ze zásobníku některý snímek odebrán. Aby nedocházelo k přeplnění zásobníku, jsou nastaveny váhy procesů detekce a sledování, které udávají jak často se má daný proces provádět. Jinými slovy jsou zavedeny priority procesů, díky kterým je možné vynechat u některých snímků detekci nebo sledování. Vynecháním detekce nebo sledování je možné rychleji odbavovat předzpracované snímky ze zásobníku při zavedení jisté chyby.

S každým vstupním videem téměř dvojnásobně stoupá výpočetní a paměťová náročnost systému. Vlákna $B_1 \dots B_n$ provádějí na každém vstupním snímku nejvíce časově náročné operace. Vlákno C je vzhledem k počtu vstupních videí nejvíce vytíženo v případě, že se na každém videu nachází několik osob, které je potřeba identifikovat.

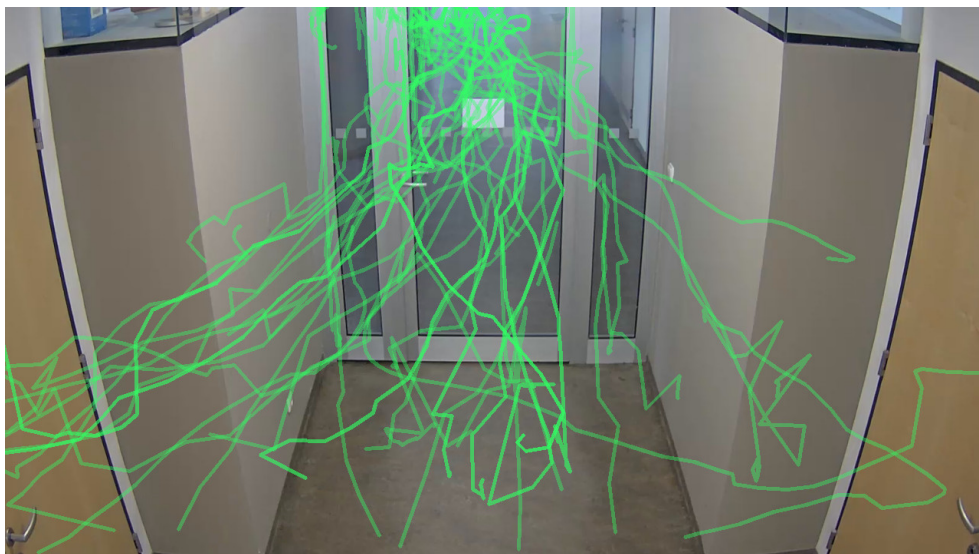
Detekované osoby vláknů $B_{1 \dots n}$ jsou následně přidány do zásobníku sledovaných osob, jak je naznačeno na obrázku 5.5. Zásobník je sdílený pro všechny vlákna B a vlákno C , přičemž každá osoba v zásobníku má identifikační číslo videa, na kterém byla detekována. Osoby v zásobníku jsou pak sledovány na každém dalším snímku. Dojde-li ke ztrátě informace o poloze některé ze sledovaných osob, není již dál sledována, ale ze zásobníku je odstraněna až ve chvíli, kdy je porovnána s databází známých osob.

Nově detekované obličeje jsou přiřazeny sledované osobě v zásobníku. Vícenásobná detekce osob, které se nacházejí v zásobníku vede ke zpřesnění rozpoznávacího procesu. Projde-li procesem rozpoznávání více obličejů stejné osoby, je možné vyfiltrovat odlehlé vektory rysů, což je experimentálně potvrzeno v kapitole 6.

Vlákno C opakovaně kontroluje obsah zásobníku sledovaných osob a rozpoznává ještě nerozpoznané osoby. Pokud se v zásobníku nachází osoba, která již není sledována, ale je známa její identita, odstraní ji a zaznamená informaci o výskytu, trajektorii a pokud je pro identifikovanou osobu nastaveno oznámení o výskytu odešle tuto informaci na nastavený email. Oznámení o výskytu lze nastavit každé osobě v databázi a také je možné nastavit dodatečný práh určující s jakou přesností musí být osoba identifikována, aby bylo oznámení o výskytu provedeno. Toto vlákno také rozpoznává pohlaví osoby.

5.4 Analýza trajektorie a počítání průchodů

Bezpečnostní systémy tohoto typu také často disponují možností analyzovat trajektorie pohybu osob. Díky této analýze je možné například zjistit do kterých dveří sledované osoby nejčastěji chodí, nebo kde se pohybovala nějaká konkrétní osoba v nějakém časovém rozmezí. Implementovaný systém nabízí možnost vykreslení trajektorie konkrétních osob v obraze za dané období do statického obrazu. Systém lze použít také k vykreslení zadané množiny osob, jak lze vidět na obrázku 5.6.



Obrázek 5.6: Diagram komunikace vláken systému.

Další užitečnou informací pro bezpečnostní systémy tohoto typu je počítání výskytů osob ve sledované oblasti. Díky této informaci je později možné například dohledat, kdo se v oblasti vyskytoval v daném období, jak často se určitá neznámá osoba pohybovala v objektu nebo k jednoduché evidenci příchodů nebo odchodů.

System eviduje výskyty známých i neznámých osob do databáze, ze které je později možné získat statistiky průchodů. Tuto funkci lze zřejmě využít pouze v případě připojení bezpečnostní kamery k systému. Pokud je systém použit k analýze videozáznamu, není možné určit přesný čas výskytu osob.

Kapitola 6

Experimenty

V následující kapitole budou zhodnoceny použité postupy, které byly popsány v předcházejících kapitolách. Většina experimentů probíhala před samotnou implementací a probíhala v pořadí, ve které byl systém vyvíjen. Díky tomuto postupu bylo možné implementovat algoritmy, které dosahovaly nejlepších výsledků.

6.1 Použité datové sady

K experimentům se statickými snímky byla převážně použita datová sada LFW¹, která obsahuje zarovnané obličej tak, aby byl každý obličej rovnoběžně s vertikální hranou obrazu. Všechny obrazy mají rozměry $250 \times 250px$ a jsou roztříděny do složek podle identity, kde název složky odpovídá jménu a příjmení osoby. Z datové sady byly odstraněny všechny identity, které měly pouze jeden obraz obličej. Datová sada obsahuje po úpravě 751 identit a 4,128 obrazů obličej.

V rámci této práce také vznikla datová sada určená především k experimentům s videozáznamem. V prostorách fakulty byly pořízeny videozáznamy z chodby ústavu ÚPGM. Z těchto videozáznamů byly extrahovány obličej procházejících osob, které byly ručně rozděleny podle identit. U každého obličej byla také extrahována informace o jeho pozici a snímku, na kterém se nachází. Díky těmto extrakcím a úpravám bylo možno vytvořit dvě datové sady. První datová sada je tvořena pouze snímky obličejů, kterým je přiřazena identita. V následujícím textu bude tato datová sada označována jako FAKULTA_A. Tento dataset obsahuje 140 osob s více než 30,000 obličej. Druhá datová sada je pouze rozšíření videozáznamů o extrahované informace určující polohu obličej na jednotlivých snímcích. V následujícím textu bude tato datová sada označena jako FAKULTA_V. Díky takto vzniklému datasetu bylo možné ověřit spolehlivost implementovaného systému.

6.2 Experimenty s extrahovaným vektorem rysů

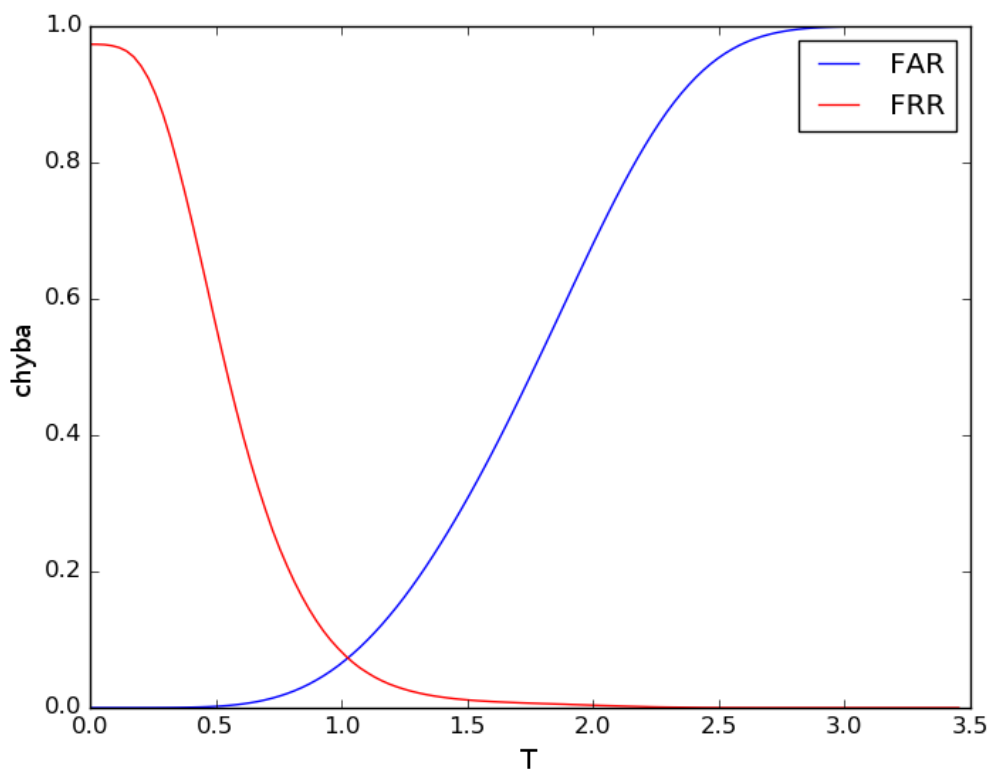
Identifikace osob je obtížným klasifikačním procesem. V našem případě se jedná o klasifikace do N tříd, kde N je počet osob v databázi. Extrahované rysy od stejné osoby si musejí být velmi podobné a naopak rysy od rozdílných osob musí být co nejvíce odlišné. Na tento fakt lze poukázat následujícím experimentem, který porovnává extrahované rysy obličejů napříč celým datasetem. Jedná se o test verifikace osob, kdy se snažíme rozhodnout, zda pochází dva obličej od stejné osoby. Tento experiment si klade za cíl ověřit vzdálenosti

¹Labeled Faces in the Wild dostupná z www: <http://vis-www.cs.umass.edu/lfw/>

extrahovaných vektorů a poukázat na separovatelnost odlišných osob pomocí extrahovaného vektoru. Vzdálenost mezi vektory je definována euklidovskou metrikou.

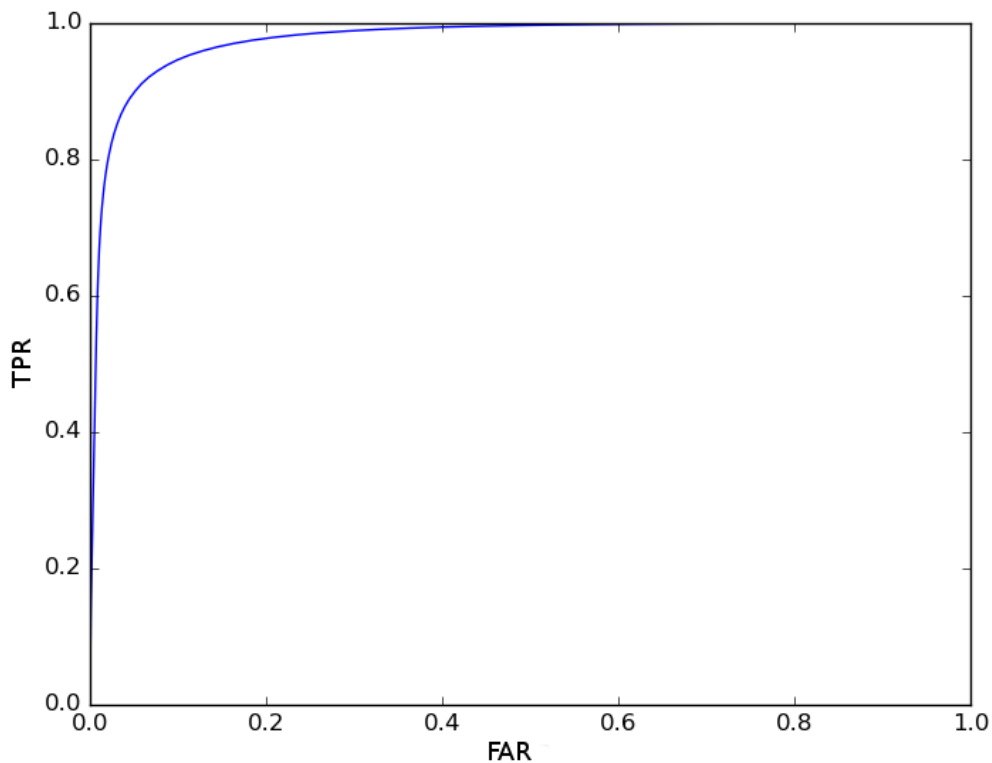
6.2.1 Verifikace

Následující experiment byl prováděn na datové sadě LFW. Experiment byl vyhodnocen vykreslením křivek FAR a FRR, při nastavení prahu T , jak můžeme vidět na obrázku 6.1 a 6.2. Práh určuje hranici vzdálenosti, kdy jsou dva porovnávané vektory obličejů shodné či nikoliv. Míra FAR nám v tomto případě udává pravděpodobnost, že dva odlišné vektory budou vyhodnoceny jako shodné, tedy pocházející od stejné osoby. Naopak míra FRR udává pravděpodobnost, že budou vektory stejných osob vyhodnoceny jako rozdílné. Cílem je míry FAR a FRR minimalizovat. Protnutí křivek udává optimální nastavení prahu T tak, aby byly obě míry minimální.



Obrázek 6.1: Graf Far/Frr reprezentující výsledky verifikace.

Výsledek tohoto experimentu ukazuje, že chybovost obou měř je 7.3% při nastavení prahu T na 1.037. Z čehož vyplývá, že 7.3% z celkového počtu 4,128 porovnávaní bylo chybně klasifikováno jako neshoda a 7.3% bylo chybně klasifikováno jako shoda. Výsledná úspěšnost je tedy 93.7%, což je velmi kladný výsledek experimentu. Výsledek tohoto experimentu ukazuje, že vektory rozdílných osob jsou dostatečně odděleny a vektory pocházející od stejných osob jsou si blízko.



Obrázek 6.2: Křivka ROC reprezentující výsledky verifikace.

6.2.2 Rozpoznávání pohlaví

Méně náročnějším klasifikačním úkolem je klasifikace pohlaví osoby. Určení identity osoby může být pozitivně ovlivněno zjištěním pohlaví identifikované osoby, pokud bude určeno s velkou přesností. Pak lze eliminovat identifikace, které vedou k neshodnému pohlaví. Také může být tato informace využita při analýzách průchodů nebo trajektorií.

K těmto experimentům byly upraveny datové sady LFW a FAKULTA_A. Úprava spočívala v rozdělení obličejů osob podle jejich pohlaví. Tyto upravené datové sady byly spojeny do jedné, na které byl prováděn tento test.

Experiment již není potřeba provádět na videozáznamu, protože datová sada FAKULTA_A byla vytvořena extrakcí z videozáznamu a předpokládá se, že by bylo dosaženo stejných výsledků. Byl použit klasifikátor KNN s parametrem k , který byl volen experimentálně. Průběh tohoto experimentu byl vnesen do tabulky 6.1, kde lze pozorovat závislost parametru k na úspěšnosti klasifikace.

Z tabulky 6.1 je zřejmé, že nastavení parametru k nehraje příliš velkou roli při nastavení $k > 10$. Z dosažených výsledků vyplývá, že extrahovaný vektor je natolik silný, že dokáže správně oddělit pohlaví osob. Jelikož je celková úspěšnost velmi dobrá, bylo implementováno rozpoznávání pohlaví osob pomocí KNN klasifikátoru s nastaveným parametrem $k = 30$ do vyvíjeného systému.

Parametr k	TP	FP	Celková přesnost
3	12,353	1,207	91.0%
5	12,746	814	93.99%
10	13,142	306	97.77%
20	13,274	218	97.87%
30	13,275	217	97.89%

Tabulka 6.1: Závislost parametru k na úspěšnosti klasifikace.

6.3 Identifikace ze statických snímků

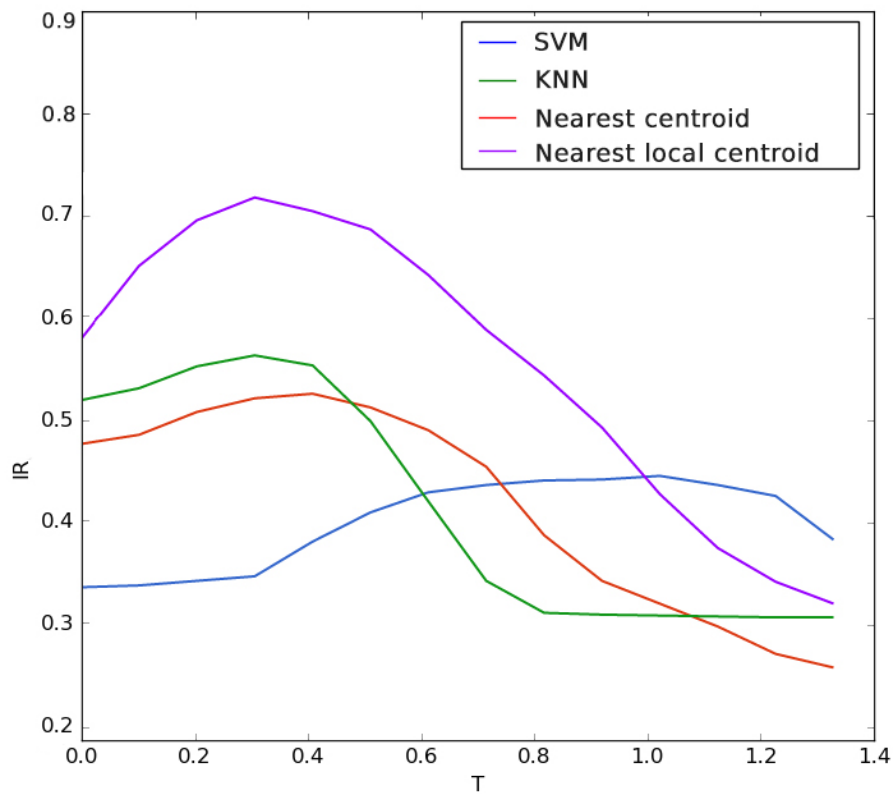
Klasifikace extrahovaného vektoru, tedy určení identity osoby, je další částí experimentů. Klasifikace extrahovaného vektoru rysů je náročnější úlohou než pouhopouhá verifikace. Při výběru klasifikátoru bude brán zřetel jak na úspěšnost klasifikace, tak na možnost přidávat další vektory do trénovací množiny klasifikátoru za běhu systému bez nutnosti opětovného trénování. Ovšem v rámci této části experimentů budou také vyhodnocovány klasifikátory, které neumožňují aktualizovat trénovací množinu za běhu systému.

Rozpoznávání osob ve videu se jeví jako obtížnější než rozpoznávání osob ze statických obrazů. Ve většině videozáznamů jsou snímky nekvalitní a obličej osoby rozostřený. Na druhou stranu se obličej stejné osoby vyskytuje na více snímcích, což může být v důsledku velmi pozitivní. Proto před samotnými experimenty s videozáznamy bude nejprve experimentováno pouze se statickými obrazy, aby bylo možno porovnat úspěšnost obou procesů klasifikace.

Datová sada použitá k těmto experimentům byla FAKULTA_A v kombinaci s LFW. Následující experiment bude porovnávat úspěšnost zvoleného klasifikátoru vzhledem k nastavení prahu T . Pro tento experiment byly vybrány klasifikátory KNN, SVM a nearest centroid. Klasifikační metoda nearest centroid hledá v databázi osob nejbližší centroid ke klasifikovanému vektoru. Tato metoda byla vytvořena experimentálně v rámci této práce.

Na obrázku 6.3 můžeme vidět závislost nastavení prahu T na míře úspěšnosti IR jednotlivých klasifikátorů. Datová sada byla náhodně rozdělena na trénovací (63%) a testovací množiny (37%), které jsou disjunktní. Experiment byl prováděn 20× a do grafu byly vyneseny průměrné hodnoty úspěšnosti jednotlivých klasifikátorů.

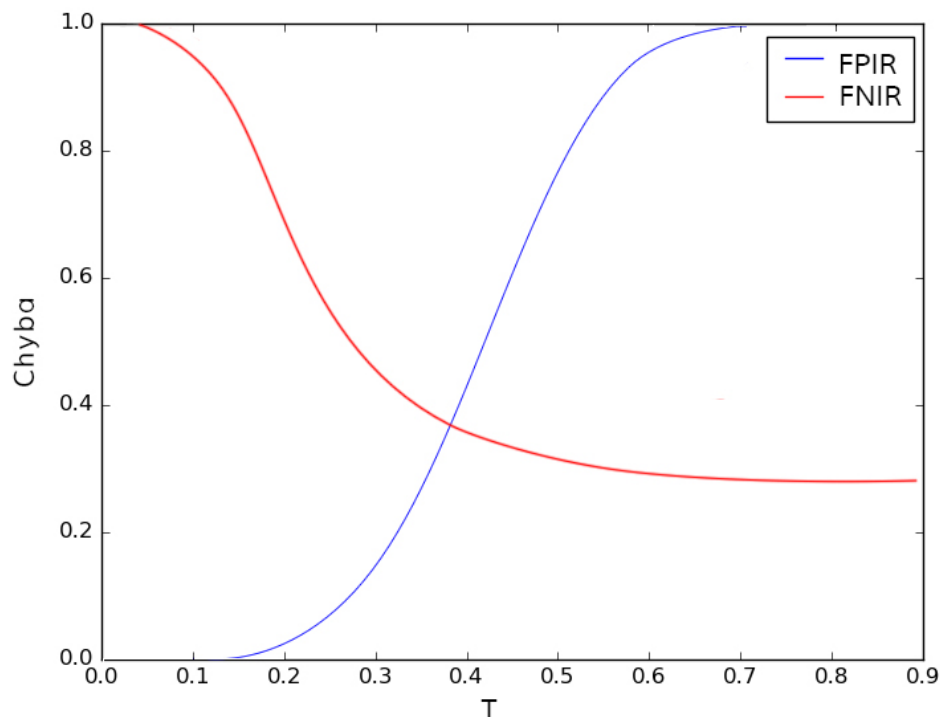
Nejúspěšnější klasifikační metodou byla metoda nalezení nejbližšího lokálního centroidu, která dosáhla úspěšnosti 72%. Jelikož tato klasifikační metoda dosáhla daleko lepších výsledků než ostatní, porovnejme křivky FPIR a FNIR (obr. 6.4).



Obrázek 6.3: Míra úspěšnosti IR vzhledem k nastavení prahu T .

Z obrázku 6.4 je zřejmé, že při nastavení prahu T na 0.39, což je místo protnutí křivek na ose x , bude méně než polovina klasifikací neúspěšných, přesněji 39% osob bude chybně klasifikováno. Tedy buď bude známá osoba klasifikována jako neznámá a nebo naopak.

Úspěšnost identifikace je nižší než předcházející verifikace napříč všemi klasifikátory. To především proto, že identifikace osob je problém $1:N$, kdežto verifikace pouze $1:1$. Při klasifikaci do N tříd může vzniknout situace, že extrahovaný vektor klasifikované osoby může být blíže některé třídě do které nenáleží. Ovšem kdyby proběhlo více klasifikací různých vektorů pocházejících od stejné osoby, pravděpodobně by bylo možné lépe určit správnou identitu. Jednalo by se o klasifikaci $M:N$, kde M je počet obličejů stejné osoby.



Obrázek 6.4: Křivka FPIR/FNIR při změně prahu T .

6.4 Experimenty s implementovaným systémem

Předcházející experimenty ukázaly, že zvolené postupy jsou vhodné pro rozpoznávání osob ze statických snímků. Systém musí správně rozpoznávat osoby na videu, což přináší spousty komplikací. Jednou z komplikací je nutnost provádět všechny algoritmy v reálném čase, což rozšiřuje sadu potřebných experimentů.

Pro co nejlepší identifikaci je potřeba získat co nejvíce snímků obličeje od jedné osoby tak, aby mohla být provedena identifikace vícekrát a výsledky zprůměrovány. Tudíž výstup klasifikátoru ještě nerozhoduje o identitě osoby, ale je potřeba provést další operace pro výběr identity z provedených klasifikací.

Aby bylo možné získat co nejvíce obrazů obličeje od stejné osoby, je nutné nejprve otestovat úspěšnost sledování objektů v obraze. K tomuto typu experimentů nebyl vytvořen žádný dataset, proto bylo sledování objektů prováděno pozorováním. Sledovací algoritmy pracovaly správně do chvíle, dokud nebyla sledovaná osoba překryta jiným, větším objektem po delší čas. Tato ztráta je kompenzována opětovnou detekcí a následným navázáním na předcházející sledování.

Implementovaný systém rozpoznává obličeje osob postupně v průběhu sledování, díky čemuž je maximalizována úspěšnost rozpoznání osoby. Identita osoby je známá až na konci sledování.

Následující experimenty byly prováděny na anotované datové sadě FAKULTA_V, byly několikrát opakovány a průměrné hodnoty jsou prezentovány tabulkou 6.2. Správná identita osoby byla určena protnutím trajektorie pohybující se osoby s reálnou pozicí obličeje. Při každém testu byla smazána databáze implementovaného systému a následně bylo přidáno

několik náhodně vybraných obličejů každé osoby. Práh T byl nastaven na hodnotu 0.591, která se v průběhu testování ukázala jako nevhodnější.

Video soubor	Počet správně rozpoznaných	Počet chybně rozpoznaných	IR	Počet výskytů osob na záznamu
20160309144212.mp4	14	2	0.88	16
20160308140907.mp4	34	8	0.81	42
20160308150908.mp4	24	6	0.8	30
20160308160908.mp4	22	4	0.85	26
20160308170908.mp4	18	6	0.75	24
20160308180909.mp4	13	2	0.87	15
20160309084210.mp4	15	6	0.71	21
20160309094210.mp4	16	2	0.89	18
20160309104210.mp4	17	3	0.85	20
20160309114211.mp4	21	5	0.81	26
20160309124211.mp4	19	8	0.7	27
20160309134212.mp4	21	5	0.81	26
20160309144212.mp4	15	6	0.71	21
20160310074219.mp4	13	1	0.93	14
20160310084219.mp4	18	3	0.86	21
20160310094220.mp4	16	4	0.8	20
20160310154534.mp4	21	5	0.81	26
20160310164534.mp4	18	3	0.86	21
20160310144534.mp4	38	7	0.84	45
20160310124533.mp4	17	3	0.85	20
Celkem	390	89	0,81	479

Tabulka 6.2: Testy na videosouborech 1.

Tabulka 6.2 prezentuje výsledky získané testováním úspěšnosti rozpoznávání na videozáznamech. Úspěšnost systému pro sledování a rozpoznávání osob se napříč všemi videi pohybuje v rozmezí 70% - 93% a celková úspěšnost na všech videích je 81%.

6.5 Časová složitost

Implementovaný systém dokáže zpracovat cca 20 snímků za vteřinu na stroji s procesorem Intel Core i5, grafickou kartou NVIDIA GeForce M820 a operační pamětí RAM 8GB při velikosti snímku $1920 \times 1080px$, pokud se v obraze nevyskytuje žádný pohyb. Je-li detekován pohyb, rychlost zpracování je snížena na cca 16 snímků za vteřinu, což je ovlivněno velikostí pohybující se oblasti na videu. Dataset FAKULTA_V byl z větší části nasnímán bezpečnostní kamerou umístěnou naproti otevírajícím se dveřím, tedy rozsah detekovaného pohybu byl v důsledku neustálého otevírání a zavírání těchto dveří velký. Tabulka 6.3 uvádí časovou složitost jednotlivých operací. Časová složitost jednotlivých akcí byla měřena pomocí knihovny *time*.

Akce	Časová složitost [ms]	Poznámka
Předzpracování obrazu	50	
Detekce obličeje	80	Závislé na rozsahu pohybu
Sledování jednoho objektu	30	Závislé na rozsahu pohybu
Extrakce vektoru rysů	152	
Klasifikace	20	
Určení pohlaví	10	

Tabulka 6.3: Časová složitost akcí prováděných systémem.

6.6 Souhrn

Společnost Facebook Inc. v roce 2015 představila na konferenci CVPR ² v Bostonu systém pro identifikaci osob ze statických snímků, nejenom ze snímků obličeje, který dosahuje úspěšnosti 83%. Ovšem společnost Facebook Inc. má k dispozici obrovské množství trénovacích dat a obrovský výpočetní výkon, jejich neuronová síť byla trénována na superpočítači. Zvolené postupy v této práci dokáží identifikovat osobu ze statického snímku podle obličeje s přesností cca 72% na datové sadě FAKULTA_A a LFW. Při rozpoznávání z videa bylo dosaženo přesnosti cca 81% na datové sadě FAKULTA_V.

Z extrahovaného vektoru rysů obličeje lze také určit pohlaví osoby s přesností 97.89%. V úloze verifikace bylo dosaženo úspěšnosti 93.7%

Implementovaný systém dokáže zpracovávat více než 20 snímků za vteřinu, při rozlišení $1920 \times 1080px$ s úspěšností cca 81%. Identifikace osob na videozáznamu dosahuje leppších výsledků zejména proto, že na videu je zachyceno více obličejů stejné osoby a je tak možné provést celý proces rozpoznávání vícekrát, což ale ne vždy vede k lepším výsledkům. Občas nastává situace, kdy jsou všechny zachycené obrazy chybně rozpoznány jako jedna jiná osoba, než která se na snímku skutečně vyskytuje. Implementovaný systém také dokáže kvalitně určit pohlaví osoby na videozáznamu a sledovat trajektorii pohybu.

²Conference on Computer Vision and Pattern Recognition URL: <http://www.pamitc.org/cvpr15/>

Kapitola 7

Závěr

V rámci této práce byla nastudována problematika detekce, sledování objektů v obraze a rozpoznávání osob podle obličeje. Praktickým výsledkem práce je systém pro rozpoznávání a sledování osob na videu a prostředky pro testování systému. Součástí praktické části práce bylo také shromáždění testovací sady videozáznamů, ze kterých byly extrahovány obrazy obličejů s přesnými souřadnicemi, čímž byla vytvořena část anotací k videům. Takto získané obličeje byly ručně roztríděny podle identity, což byla druhá část anotací k videům. Výsledné anotace k jednotlivým videům jsou tvořeny přesnou pozicí, na které se nachází konkrétní osoba. Na základě této datové sady bylo možné otestovat úspěšnost rozpoznávání implementovaného systému. Datová sada obsahuje 120 různých osob a celkový počet obličejů je 32,750. Datová sada byla vytvořena ze 67 hodinových videozáznamů.

Samotný systém je implementován v jazyce Python s více pracovními vlákny. Databáze osob je implementována pomocí NoSQL databázového systému MongoDB. K extrakci rysů obličeje z obrazu byly použity konvoluční neuronové sítě. Pro klasifikaci extrahovaného vektoru rysů byl použit modifikovaný klasifikátor *Nearest Centroid*. Modifikace klasifikátoru spočívá v upravení způsobu trénování a to tak, že vektory příslušející dané osobě jsou podrobeny shlukové analýze metodou DBSCAN, která patřičně oděluje podobné vektory a odstraňuje odlehlé hodnoty a zašumělá data. Nad takto vytvořenými shluky jsou vypočteny lokální centroidy. Identifikace osoby z obrazu pak probíhá na základě porovnání euklidovské vzdáleností extrahovaného vektoru rysů a lokálních centroidů. Osoba s nejbližším lokálním centroidem k vektoru rysů je výsledkem procesu klasifikace.

Nejprve bylo provedeno testování rozpoznávání na statických snímcích s úspěšností 72%. Následně bylo na základě vytvořeného datasetu zjištěno, že implementovaný systém dokáže rozpoznávat osoby z videozáznamu s přesností 81%. Tento rozdíl je zřejmě způsoben tím, že na videu je tatáž osoba vícekrát rozpoznávána a identita osoby je určena na základě všech provedených procesů rozpoznávání současně. Také byly provedeny experimenty s extrahovaným vektorem rysů a bylo zjištěno, že je možné na základě vektorů rysů určit pohlaví osoby s přesností 97.89%

Stejně jako se neustále vyvíjí a zlepšují metody pro rozpoznávání obličejů, je taktéž možné vyvíjet a zlepšovat implementovaný systém tak, aby dosahoval lepších výsledků a to až do té doby, dokud nebude rozpoznávání obličeje stoprocentní. Pro další rozvoj systému by bylo vhodné vytvořit rozhraní pro připojení IP kamer, zlepšit přesnost rozpoznávání a snížit výpočetní nároky systému.

Literatura

- [1] Bertozzi, M.; Broggi, A.; Del Rose, M.; aj.: A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier. 30 2007-oct. 3 2007, doi:10.1109/ITSC.2007.4357692.
- [2] Baker S., M. I., Goss R.: *Lucas-Kanade 20 years on: a unifying framework*. International Journal of Computer Vision, 2004, s. 221–255.
- [3] Barron, J. L.; Fleet, D. J.; Beauchemin, S. S.: Performance of Optical Flow Techniques. *Int. J. Comput. Vision*, ročník 12, č. 1, Únor 1994: s. 43–77, ISSN 0920-5691, doi:10.1007/BF01420984.
URL <http://dx.doi.org/10.1007/BF01420984>
- [4] Bradski, G. R.: *Learning OpenCV*. Sebastopol: O'Reilly, c2008, ISBN 9780596516130.
- [5] Burt, P.; Adelson, E.: The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, ročník vol. 31, č. issue 4, 1983: s. 532–540, ISSN 00962244, doi:10.1109/TCOM.1983.1095851.
- [6] Cheung, S.-C. S.; Kamath, C.: Robust Background Subtraction with Foreground Validation for Urban Traffic Video. *EURASIP J. Appl. Signal Process.*, ročník 2005, Leden 2005: s. 2330–2340, ISSN 1110-8657, doi:10.1155/ASP.2005.2330.
- [7] Dalal, B., N.; Triggs: Histograms of Oriented Gradients for Human Detection. <http://lear.inrialpes.fr/pubs/2005/DT05/>.
- [8] Drahanský, M.; Orság, F.: *Biometrie*. [Brno: M. Drahanský], 2011, ISBN 9788025489796.
- [9] Fukushima, K. a. S. M.: Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position [online]. 1982, s. 267-285 [cit. 2015-12-23]. DOI: 10.1007/978-3-642-46466-9_18. Dostupné z: http://link.springer.com/10.1007/978-3-642-46466-9_18.
- [10] Gutarra, E.: RMS and Backpropagation for Feedforward Neural Networks. 29 January 2011.
- [11] Hlaváč, V.; Šonka, M.; Boyle, R.: *Image processing analysis and machine vision*. c1999.
- [12] Hradiš, M.: *Deep CNN for Computer Vision*. 2014 [cit. 2015-12-10].

- [13] ISO/IEC: *19795-1:2006 - Information technology – Biometric performance testing and reporting – Part 1: Principles and framework*.
URL http://www.iso.org/iso/catalogue_detail.htm?csnumber=41447
- [14] Itseez: Open Source Computer Vision Library.
<https://github.com/itseez/opencv>, 2015.
- [15] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, editace F. Pereira; C. Burges; L. Bottou; K. Weinberger, Curran Associates, Inc., 2012, s. 1097–1105.
- [16] Lecun, Y.; Bottou, L.; Bengio, Y.; aj.: Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998, s. 2278–2324.
- [17] Lienhart, R. a. J. M.: *An extended set of Haar-like features for rapid object detection*. In: *Proceedings. International Conference on Image Processing*. IEEE, 2002, s. I-900-I-903. DOI: 10.1109/ICIP.2002.1038171. ISBN 0-7803-7622-6.
- [18] Lucas, B. D.; Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, s. 674–679.
- [19] Ming-hsuan, K. D. A. N., Y.: *Detecting faces in images: A survey*. IEEE Transactions on Pattern Analysis and Machine Inteligence, 2002, vol. 24, no. 1.
- [20] Viola, J. M., P.: *Rapid object detection using a boosted cascade of simple features*, *Computer Vision and Pattern Recognition, 2001*. CVPR 2001, 2001, proceedings of the 2001 IEEE Computer Society Conference on, 2001, vol.1, s.I-511,I-518.
- [21] Viola, J. M., P.: *Robust real-time face detection*. International Journal of Computer Vision, 2004, vol. 57, no. 2, s. 137-154.
- [22] Zbořil, F.: SFC Soft Computing. (přednáška), 2013.

Přílohy

Seznam příloh

A Obsah CD

48

Příloha A

Obsah CD

Přiložené DVD obsahuje následující položky:

1. src - zdrojové kódy implementovaného systému
2. data - data určené k testování systému
3. doc - zdrojové kódy LaTeXu technické zprávy
4. readme.md - návod na použití systému