FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Kamerový bezpečnostní systém s rozpoznáváním obrazu

Studijní program: B0613A140005 Informační technologie
Studijní obor: Aplikovaná informatika

Autor práce: **Jaroslav Pryjmak**
Vedoucí práce: Ing. Jana Kolaja Ehlerová, Ph.D.
Ústav nových technologií a aplikované informatiky

Liberec 2023

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Zadání bakalářské práce

Kamerový bezpečnostní systém s rozpoznáváním obrazu

Jméno a příjmení:	**Jaroslav Pryjmak**
Osobní číslo:	M20000071
Studijní program:	B0613A140005 Informační technologie
Specializace:	Aplikovaná informatika
Zadávající katedra:	Ústav nových technologií a aplikované informatiky
Akademický rok:	2022/2023

Zásady pro vypracování:

1. Proveďte rešerši dostupných systémů pro rozpoznávání obličeje či postavy a zabezpečovacích systémů založených na rozpoznávání obrazu.
2. Navrhněte vlastní systém s použitím minipočítače, připojením k webové aplikaci a možností vzdáleného ovládání.
3. Implementujte a otestujte systém v poloreálném provozu. Zajistěte zabezpečení systému.
4. Proveďte kritické zhodnocení, navrhněte alternativy k navrženému řešení a možnosti rozšíření.

Rozsah grafických prací:	dle potřeby
Rozsah pracovní zprávy:	30 – 40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] HOGGART, S. G. *Mathematics of digital images: creation, compression, restoration, recognition*. Cambridge: Cambridge University Press, 2006. ISBN 0-521-78029-2.

[2] MONK, Simon. *Raspberry Pi cookbook*. Second edition. Beijing: O'Reilly, 2016. ISBN 978-1-491-93910-9.

[3] TEVAULT, Donald A.. *Mastering Linux security and hardening: secure your Linux server and protect it from intruders, malware attacks, and other external threats*. Packt, 2018. ISBN-13: 978-1788620307.

Vedoucí práce:	Ing. Jana Kolaja Ehlerová, Ph.D. Ústav nových technologií a aplikované informatiky
Datum zadání práce:	12. října 2022
Předpokládaný termín odevzdání:	22. května 2023

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2022

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

7. února 2023 Jaroslav Pryjmak

Kamerový bezpečnostní systém s rozpoznáváním obrazu

Abstrakt

S rostoucími požadavky na kvalitu zabezpečení majetku a budov je stále častějším prvkem v zabezpečovacích systémech rozpoznání obličeje. Tato práce se zabývá návrhem a implementací kamerového systému s funkcí rozpoznání obličeje za použití minipočítače Raspberry Pi. Součástí systému je webová aplikace sloužící k jeho ovládání. V navrženém systému může být pro detekci obličejů zvolen algoritmus na bázi histogramu orientovaných gradientů nebo konvoluční neuronové sítě. Výsledkem práce je funkční prototyp systému, který je schopen rozpoznat obličej a přiřadit mu jméno. Zároveň byla zhodnocena funkčnost tohoto systému a byly navrženy možné alternativy a vylepšení.

Klíčová slova: zabezpečovací systém, Raspberry Pi, rozpoznávání obličeje, detekce obličeje, histogram orientovaných gradientů, konvoluční neuronové sítě, webová aplikace

Camera security system using image recognition

Abstract

With the growing demands for quality security of property and buildings, facial recognition is becoming an increasingly common feature in security systems. This work deals with the design and implementation of a camera system with facial recognition capabilities using the Raspberry Pi mini-computer. The system includes a web application for its control. In the proposed system, a face detection algorithm based on a histogram of oriented gradients or convolutional neural networks can be selected. The result of this work is a functional prototype of the system, which is capable of recognizing a face and assigning a name to it. At the same time, the functionality of this system was evaluated, and possible alternatives and improvements were proposed.

Keywords: security system, Raspberry Pi, face recognition, face detection, histogram of oriented gradients, convolutional neural networks, web application

Obsah

Seznam zkratek

CNN	Convolutional Neural Network (Konvoluční neuronová síť)
EXIF	Exchangeable Image File Format (Formát pro výměnu obrazových souborů)
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
HOG	Histogram of Oriented Gradients (Histogram orientovaných gradientů)
PIR	Passive Infrared Sensor (Pasivní infračervený senzor)
RPi	Raspberry Pi
SVM	Support Vector Machine (Metoda podpůrných vektorů)
TUL	Technická univerzita v Liberci
VNC	Virtual Network Computing (Virtuální počítačová síť)

1 Úvod

V současné době je poptávka po kvalitních zabezpečovacích systémech vysoká. Stále častějším požadavkem je kromě monitorování pohybu v prostoru i určení totožnosti zaznamenaných osob. Jedním z rychlých, pohodlných a bezkontaktních způsobů ověření identity osob je rozpoznávání obličeje. To se stává stále populárnějším v různých oblastech, jako je kontrola přístupu do uzavřených prostorů, sledování přítomnosti pracovníků nebo monitorování veřejných i soukromých zařízení.
Prvním krokem při řešení této práce je provedení rešerše již existujících systémů pro rozpoznávání obličeje a také zabezpečovacích systémů založených na rozpoznávání obrazu. Na základě získaných informací bude navržen vlastní zabezpečovací systém využívající minipočítač a webovou aplikaci, která umožní jeho vzdálené ovládání. Důležitým úkolem je zajištění dostatečně silného zabezpečení systému a citlivých informací, se kterými pracuje.
Hlavním cílem této práce je navržený systém úspěšně naimplementovat a otestovat v poloreálném provozu. Na základě výsledků bude provedena analýza funkčnosti systému a jeho nedostatků. Následně budou navržena možná vylepšení a alternativy ke stávajícímu návrhu.
V první kapitole jsou poskytnuty informace, které byly získány v rámci rešerše a tvoří podklad pro návrh systému.
Následně jsou v kapitole 3 podrobně představeny tři často používané algoritmy pro detekci obličeje: Viola-Jones detektor, histogram orientovaných gradientů a konvoluční neuronové sítě. Také je nastíněn způsob rozpoznávání osob na základě 128 charakteristických rysů obličeje.
Následně je v práci popsán samotný návrh systému využívající Raspberry Pi a webovou aplikaci, které spolu komunikují přes databázi. V textu je diskutováno jak hardwarové zapojení, tak i softwarové řešení.
Důraz je kladen také na kvalitní zabezpečení zajišťující ochranu citlivých dat uživatelů před ztrátou nebo odcizením.
Kapitola 6 se věnuje procesu kalibrace a jsou v ní uvedeny výsledky testování navrženého systému v poloreálném provozu a zhodnocení funkčnosti jak vybraných detkčních algoritmů, tak i celého systému globálně. V závěrečné kapitole autor poukazuje na nedostatky a navrhuje možná vylepšení.

2 Rešerše

V rámci této práce byla provedena rešerše existujících komerčních bezpečnostních systémů, které využívají detekci a rozpoznávání obličeje jako klíčovou součást svých funkcí. Tyto systémy jsou stále častěji využívány v různých odvětvích pro zvýšení bezpečnosti a pohodlí uživatelů. Kromě komerčních řešení byla také provedena analýza projektů, které se zaměřují na implementaci detekce a rozpoznávání obličeje s využitím minipočítače Raspberry Pi. Tyto projekty nabízejí přístup k technologiím rozpoznávání obličeje za nižší náklady a jsou vhodné pro výzkumné účely, vývoj prototypů nebo pro menší aplikace. Studium těchto projektů poskytuje cenný vhled do možností využití Raspberry Pi v oblasti detekce a rozpoznávání obličeje a umožňuje identifikovat klíčové aspekty, které by mohly být aplikovány v rámci této práce.

2.1 Komerční bezpečnostní systémy na bázi rozpoznávání obrazu

Použití komerčních bezpečnostních systémů s rozpoznáváním obrazu je již v dnešní době zcela běžné. Tyto systémy se nejčastěji používají v pracovním prostředí jako jsou kancelářské budovy, obchodní domy, banky apod.
Pro taková prostředí nabízí například společnost *i-PRO Americas Inc.* produkt *MonitorCast Access Control* a společnost *Avigilon™* produkt *Access Control Manager*. Tyto produkty nabízí kamerové systémy se schopností rozpoznání obličeje, konfiguraci databáze pro známé obličeje a webové prostředí pro management celého systému.
V případě zabezpečení domácností existují produkty od společností jako je *Wyze Labs Inc.*, kde lze použít jednodušší implementaci bezpečnostního systému v podobě jedné (či více) kamery (např. *Wyze Cam V3*) komunikující přes Wi-Fi s aplikací v telefonu přes servery výrobce. Použití rozpoznání obličeje v těchto systémech již není tak běžné.
Všechny tři zmíněné systémy mají i svá negativa: systémy *MonitorCast Access Control* a *Access Control Manager* jsou vhodné pouze pro prostředí s vyšším počtem návštěvníků a jsou méně cenově příznivé, naopak systém *Wyze Cam V3* má velmi omezené funkce.
Systém navržený v této práci se pokouší přiblížit funkčností systémům *MonitorCast Access Control* a *Access Control Manager* a cenové stránce systému *Wyze Cam V3*.

2.2 Projekty implementující rozpoznávání obličeje

Tato práce byla inspirovaná následujícími projekty, které implementují detekci a rozpoznávání obličeje s využitím Raspberry Pi:

- *Raspberry Pi based access control using face recognition*[1]
- *Raspberry Pi face recognition*[2]
- *Face recognition with OpenCV, Python and deep learning*[3]
- *Face detection with dlib (HOG and CNN)*[4]
- *Real Time Face Recognition with Raspberry Pi and OpenCV*[5]

Projekt [1] řeší implementaci systému v rámci (IoT) s využitím Raspberry Pi Model 3 B+ jako hlavní komponentu. Systém je navržen tak, aby byl po stisku tlačítka schopen rozpoznat známé osoby z databáze a upozornit administrátora na jejich přítomnost prostřednictvím mobilního zařízení. Použitá metoda detekce obličeje vychází z HOG algoritmu a rozpoznání obličeje je založeno na předtrénované neuronové síti z knihovny *dlib*. Výsledek rozpoznání je poslán administrátorovi, který může rozhodnout o povolení či zamítnutí přístupu návštěvníka.

Projekty [2, 3, 5] se zaměřují pouze na detekci a rozpoznávání obličejů pomocí Raspberry Pi, bez konstruování komplexnějšího bezpečnostního systému. Projekty [2, 3] testují všechny tři metody detekce: Viola-Jones detektor, HOG (histogram orientovaných gradientů) a CNN (konvoluční neuronová síť), zatímco projekt [5] implementuje pouze kaskádu klasifikátorů pro detekci obličejů. Všechny tři projekty využívají předtrénovanou neuronovou síť pro rozpoznávání obličeje a kombinují knihovny *OpenCV* a *dlib*. Tyto projekty také testují real-time rozpoznávání, avšak kvůli hardwarovému omezení je rozpoznávání velmi pomalé.
Další projekt [4] se zaměřuje na detekci obličejů pomocí knihovny *dlib*, avšak pouze s využitím HOG a CNN metod. Tento projekt nezahrnuje rozpoznávání obličeje, ale pouze detekci.

Ze zmíněných projektů byla převzata myšlenka použití Raspberry Pi s kamerou jako základu pro rozpoznávací systém. Z projektů [1, 3] byla převzata možnost použití již natrénované neuronové sítě pro extrakci 128 prvků obličeje pro rozpoznávání (sekce 3.4). Tyto projekty často využívají open-source neuronovou síť vyvinutou týmem z *OpenFace*[6] a algoritmy pro porovnávání výsledných vektorů poskytnutých neuronovou sítí.
Z výsledků zmíněných projektů vyplývá, že algoritmus na bázi CNN je nejpřesnější, ale také hardwarově a časově náročný. Viola-Jones detektor pracuje nejrychleji, ale je méně přesný. Algoritmus na bázi HOG představuje kompromis mezi přesností a náročností, což jej činí vhodným kandidátem pro implementaci na Raspberry Pi. Projekt [1] uvádí čas inicializace cca 8 min a 20 s a čas detekce a rozpoznávání 20 s. V rámci této práce by mohly být tyto časy minimalizovány pro optimální chod zabezpečovacího systému.

2.3 Vývoj open-source knihoven a frameworků pro detekci a rozpoznávání obličeje

Existuje řada open-source knihoven a frameworků, které usnadňují vývoj a implementaci detekce a rozpoznávání obličeje. Mezi ně patří:

- *OpenCV* - populární knihovna pro zpracování obrazu a počítačové vidění, která obsahuje nástroje pro detekci a rozpoznání obličeje. Knihovna OpenCV je primárně určena pro jazyk C++, ale také poskytuje rozhraní pro Python, Java a MATLAB.

- *Dlib* - knihovna pro strojové učení a vývoj aplikací, která zahrnuje nástroje pro detekci a rozpoznávání obličeje. Knihovna Dlib je napsaná v C++ a poskytuje rozhraní pro Python.

- *TensorFlow* a *Keras* - populární frameworky pro vývoj a trénování neuronových sítí, které mohou být použity pro detekci a rozpoznávání obličeje. TensorFlow, vyvinutý společností Google, podporuje širokou škálu jazyků, včetně jazyků Python, C++, Java a JavaScript. Keras je vyšší úroveň API pro TensorFlow a Theano, který je zaměřen na snadný vývoj hlubokých učebních modelů, a je primárně určen pro jazyk Python.

Vývojáři mohou využít těchto knihoven a frameworků pro rychlý vývoj a prototypování detekce a rozpoznávání obličeje na různých platformách včetně Raspberry Pi. Tyto open-source nástroje také podporují výměnu znalostí a zkušeností mezi vývojáři, což urychluje výzkum a inovace v oblasti detekce a rozpoznávání obličeje.

3 Detekce a rozpoznávání obličeje

Detekce a rozpoznávání obličeje představují dva základní kroky ve zpracování obrazu, které umožňují identifikaci osob v digitálních snímcích. Tyto kroky lze rozdělit do dvou hlavních fází.
První fáze, detekce obličeje, zahrnuje schopnost algoritmu:

- Lokalizovat všechny obličeje ve snímku, a to i v případě různých úhlů záběru, velikostí nebo změn osvětelní

- Analyzovat a sledovat jednotlivé obličeje, aby bylo zajištěno, že i při změnách vzhledu způsobených perspektivou nebo osvětlením, je stále rozpoznáván jako ten samý obličej

Algoritmy, které splňují tyto požadavky a jsou často používány v oblasti rozpoznávání obličejů, zahrnují Viola-Jones detektor, HOG s lineárním SVM klasifikátorem a konvoluční neuronové sítě (CNN). Tyto algoritmy byly zvoleny, protože mají dobrou reputaci v oblasti rozpoznávání obličejů a jsou široce používány v praxi. Každý z těchto algoritmů představuje unikátní přístup k detekci a rozpoznávání obličejů, což umožňuje porovnat jejich výkonnost a zjistit, který z nich nejlépe vyhovuje konkrétním potřebám projektu. Navíc, tyto algoritmy mají rozsáhlou dokumentaci a dostupné zdroje pro implementaci, což usnadňuje jejich začlenění do projektu.

Druhá fáze, rozpoznávání obličeje, vyžaduje, aby algoritmus byl schopen:

- Identifikovat unikátní rysy obličeje, jako jsou proporce obličeje, velikost a poloha očí, tvar nosu a úst, které umožňují rozlišení jednotlivých osob

- Porovnat zjištěné unikátní rysy s databází známých obličejů a na základě tohoto porovnání přiřadit obličej k správně osobě

Metody založené na neuronových sítích, zejména hlubokých konvolučních neuronových sítích (DCNN), se pro rozpoznávání obličejů staly velmi účinnými. Tyto sítě jsou schopny efektivně zpracovávat a klasifikovat složité vzory a struktury v obličeji díky své hluboké architektuře a schopnosti učit se hierarchické rysy. DCNN získávají svůj výkon z mnoha konvolučních a plně propojených vrstev, které dokáží zachytit a rozlišit jemné detaily a rozdíly mezi jednotlivými obličeji. Tímto způsobem se tyto metody staly vynikajícím nástrojem pro rozpoznávání obličejů s vysokou úrovní přesnosti a robustnosti v různých podmínkách osvětlení a natočení obličeje.

3.1 Viola-Jones detektor

Viola-Jones detektor je známý především pro svou rychlost detekce. Tento detektor zpracovává obrazy ve stupních šedi. Pro trénink detektoru se používá AdaBoost algoritmus, který kombinuje základní klasifikátory do kaskády na základě souboru pozitivních a negativních vzorků. Jako základní klasifikátor jsou zde použity Haarovy příznaky s určenou prahovou hodnotou. Pro zrychlení algoritmu se používá integrální obraz. Viola-Jones detektorem se zabývají práce [7, 8, 9].

3.1.1 Integrální obraz

Pro rychlé výpočty obdélníkových příznaků se používá reprezentace obrazu nazývaná integrální obraz. Hodnota integrálního obrazu na souřadnicích x, y je rovna součtu pixelů vlevo nahoře od souřadnic x, y včetně pixelu na souřadnicích x, y.
Hodnota na souřadnicích x, y se tedy vypočítá jako:

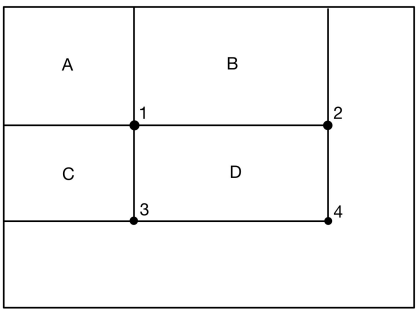
$$ii(x, y) = \sum_{x' \leq x, y \leq' y} i(x', y'), \tag{3.1}$$

kde $ii(x, y)$ jsou hodnoty integrálního obrazu a $i(x, y)$ jsou hodnoty pixelů vstupního obrazu.

31	2	4	33	5	36
12	26	9	10	29	25
13	17	21	22	20	18
24	23	15	16	14	19
30	8	28	27	11	7
1	35	34	3	32	6

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

Obrázek 3.1: Číselné hodnoty levé matice představují hodnoty pixelů vstupního obrazu. Hodnoty barevných polí pravé matice se rovnají součtu hodnot polí v rámečku stejné barvy levé matice.



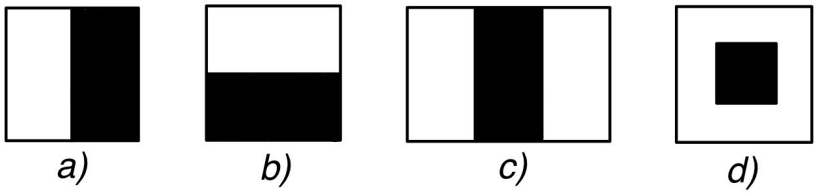
Obrázek 3.2: Součet pixelů v obdélníku D lze vypočítat jako $4 + 1 - (2 + 3)$.

3.1.2 Haarovy příznaky

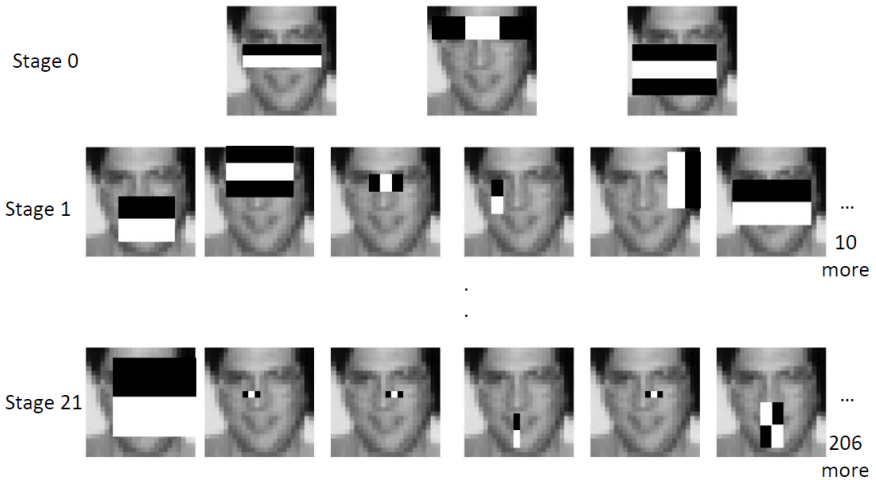
Všechny tváře mají společné rysy, jako například tmavší oblast kolem očí než okolní pokožka a světlejší oblast nosu než oblast očí. Haarovy příznaky, které jsou odvozeny od obdélníkových příznaků, umožňují určit, která oblast je světlejší a která je tmavší. Podle konkrétní části obličeje se vybere správný typ příznaku: hranový, čárový nebo středový. Každý obdélník má přiřazenou váhu, která je určena podle jeho barvy a velikosti. Bílá oblast má váhu $w_0 = -1$ a černá oblast váhu w_1 vypočtenou jako podíl ploch bílé a černé oblasti. Váhy se používají pro výpočet odezvy Haarového příznaku na vstupní obraz.

$$f(x) = w_0 r_0 + w_1 r_1, \tag{3.2}$$

kde $f(x)$ je odezva Haarového příznaku na vstupní obraz x, w_0 je váha bílé oblasti r_0 a w_1 je váha černé oblasti r_1.



Obrázek 3.3: Příklad a) vertikálního hranového, b) horizontálního hranového, c) čárového a d) středového příznaku.



Obrázek 3.4: Aplikace haarových příznaků na obličeji při klasifikování v kaskádě. Převzato z [10].

3.1.3 Lineární klasifikátor

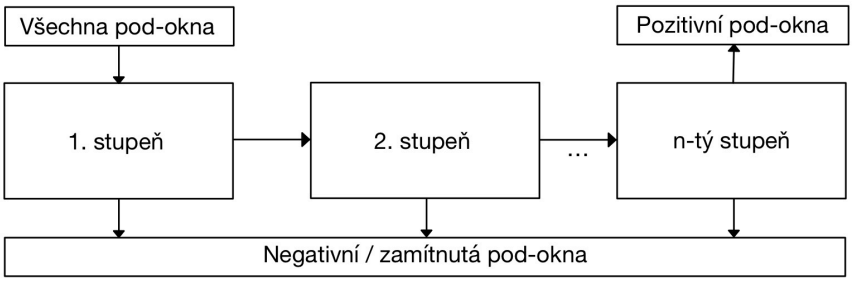
Cílem lineárního klasifikátoru je přiřadit data do jedné ze dvou tříd (pozitivní a negativní). Lineární slabý klasifikátor se skládá z příznaku, polarity a prahové hodnoty. V následující rovnici je popsáno, jakým způsobem klasifikátor určuje, do které třídy obraz x patří:

$$h_j(x) = \begin{cases} 1 & pf(x) < p\Theta \\ 0 & jinak, \end{cases} \tag{3.3}$$

kde f je hodnota Haarova příznaku, p je polarita a Θ je prahová hodnota lineárního klasifikátoru.
Při aplikaci klasifikátoru na vstupní obraz x se nejdříve vypočte odezva příznaku. Poté se podle polarity p určí, zda je pozitivní klasifikační třída nad ($p = 1$) nebo pod ($p = -1$) prahovou hodnotou Θ. Nakonec se porovná odezva příznaku s prahovou hodnotou Θ a podle toho se rozhodne, do které třídy obraz x patří, pozitivní ($h(x) = 1$) nebo negativní ($h(x) = 0$).

3.1.4 Kaskáda klasifikátorů

Kaskáda klasifikátorů se skládá z několika fází, přičemž každá fáze zahrnuje několik slabých klasifikátorů. Každá fáze kaskády je silným klasifikátorem s vlastní limitní hodnotou, podle které se rozhoduje, zda je aktuální pod-okno pozitivní nebo negativní. Cílem kaskády je zrychlit detekci objektů tak, že každá fáze předává pozitivní pod-okno do další fáze a současně zamítá negativní pod-okno.



Obrázek 3.5: V každé fázi se určí, zda je pod-okno pozitivní a pokračuje do další fáze, nebo je negativní a vyloučí se.

První fáze by měla být co nejefektivnější a zamítnout co nejvíce pod-oken. Následující fáze zahrnují všechny klasifikátory z předchozí fáze. Kaskáda je trénována tak, aby dosáhla vysoké úrovně přesnosti detekce a nízkého počtu falešně pozitivních detekcí.

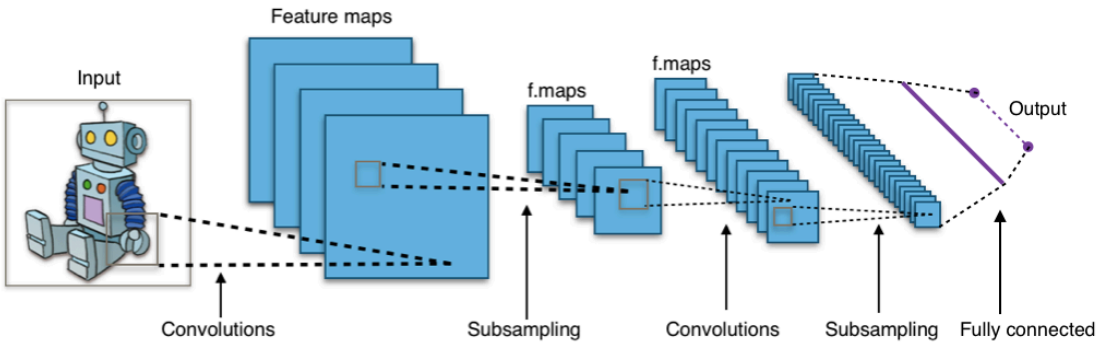
3.2 Konvoluční neuronová síť

Konvoluční neuronové sítě (CNN) představují speciální kategorii hlubokých neuronových sítí, které byly původně navrženy pro účely zpracování obrazu a analýzy

vizuálních dat. Na rozdíl od klasických neuronových sítí, CNN obsahuje více hlubokých vrstev, které umožňují efektivnější zpracování prostorových a hierarchických informací z obrazu. Architektura CNN zahrnuje váhy, zkreslení (bias) a nelineární aktivační funkce, podobně jako u běžných neuronových sítí. Neurony v CNN jsou uspořádány ve více vrstvách ve formě volumetrických struktur, což umožňuje zachování prostorových vztahů mezi pixely vstupního obrazu. V posledních letech se staly stěžejními nástroji pro řadu úloh spojených s počítačovým viděním, jako je detekce objektů, rozpoznávání obličejů, segmentace obrazu a další. Tato kapitola pojednává o základních konceptech a vrstvách konvolučních neuronových sítí, jejich vlastnostech a funkcích. [11, 12]

3.2.1 Konvoluční vrstva

Konvoluční vrstva představuje základní stavební prvek konvolučních neuronových sítí a umožňuje sítím učit se a rozpoznávat vzory a struktury v obrazech. Provádí konvoluci na vstupních datech s použitím konvolučního jádra a rozpoznává lokální rysy, přičemž snižuje počet parametrů potřebných k učení. Jako první vrstva extrahuje příznaky ze vstupních obrazů pomocí konvoluce mezi vstupním obrazem a filtrem o velikosti MxM.
Výstupem je mapa příznaků, která poskytuje informace o obraze, jako jsou rohy a hrany. Tato mapa je následně předána dalším vrstvám k naučení dalších příznaků. Konvoluční vrstvy v CNN využívají především zachování vzdálenosti mezi pixely. [11, 12, 13]



Obrázek 3.6: Architektura CNN, která se skládá z konvoluční vrstvy, pooling (subsampling) vrstvy a plně propojené vrstvy. Konvoluční a pooling vrstvy se typicky střídají a hloubka každého filtru se zvyšuje zleva doprava, zatímco výstupní velikost se zmenšuje. Převzato z [14].

3.2.2 Pooling vrstva

Pooling vrstva představuje další důležitou součást konvolučních neuronových sítí, jejímž cílem je snížit velikost map příznaků a redukovat výpočetní nároky. Provádí zjednodušení dat zkonvoluovaných v předchozí vrstvě prostřednictvím různých operací, jako je max pooling, average pooling nebo sum pooling, snižuje spojení mezi

vrstvami a nezávisle operuje na každé mapě příznaků. Tyto operace zajišťují, že síť zachovává důležité informace, zatímco se stává invariancí k malým posunům.
V max poolingu se vybírá největší prvek z mapy, zatímco average pooling vypočítá průměr prvků v předdefinované velikosti sekce obrazu. Sum pooling spočítá celkový součet příznaků v předdefinované sekci. Pooling vrstva obvykle slouží jako most mezi konvoluční vrstvou a plně propojenou vrstvou. [11, 13]

3.2.3 Plně propojená vrstva

Plně propojená vrstva, také známá jako dense vrstva nebo Fully Connected (FC) vrstva, se obvykle nachází na konci konvoluční neuronové sítě a slouží k zajištění finálního zpracování a klasifikace výstupu z předchozích vrstev. Tato vrstva se skládá z vah, biasů a neuronů a slouží ke spojení neuronů mezi dvěma různými vrstvami. Plně propojené vrstvy jsou zodpovědné za spojení rysů z předchozích vrstev a rozhodnutí o konečném výsledku klasifikace.
Vstupní obraz z předchozích vrstev se zploští a vloží do vrstvy FC, kde začíná klasifikační proces. Tyto vrstvy jsou obvykle umístěny před výstupní vrstvou a tvoří posledních několik vrstev architektury CNN. Důvodem proč jsou dvě vrstvy propojeny je to, že dvě plně propojené vrstvy provádí operace lépe než jedna propojená vrstva. Plně propojená vrstva přijímá výstupy z předchozích vrstev a kombinuje je do konečného výstupu, který reprezentuje pravděpodobnosti příslušnosti k jednotlivým třídám. [11, 13]

3.2.4 Aktivační funkce

Aktivační funkce představují důležitý prvek konvolučních neuronových sítí, který přidává nelinearitu do sítě a umožňují síti učit se složitější a nelineární vzory v datech. Používají se k učení a aproximaci jakéhokoliv druhu průběžného a komplexního vztahu mezi proměnnými v síti a určují, které informace modelu by měly být aktivovány na začátku a které na konci sítě. Aktivační funkce se používají v konvolučních, pooling a plně propojených vrstvách.
Mezi běžné aktivační funkce patří softmax, sigmoid, tanh (hyperbolický tangent) a ReLU (Rectified Linear Unit). Právě funkce ReLU byla v poslední době velmi široce používána, jelikož je téměř lineární a zachovává vlastnosti, které usnadňují optimalizaci lineárního modelu. Rychlost konvergence je proto ve fázi náhodného gradientního sestupu vyšší. V případě, že je rychlost učení příliš vysoká, mohou být aktualizace váh v modelu nestabilní, což zpomaluje konvergenci trénování a může ovlivnit aktivaci funkce.
Funkce ReLU lze popsat následovně:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & jinak, \end{cases} \tag{3.4}$$

kde x je vstup neuronu. Tato funkce má hodnotu nula pro negativní hodnoty a roste lineárně pro kladné hodnoty, což nemá vliv na velikost objemu. Pooling (subsampling) vrstva vrací maximální aktivační hodnotu v regionu, čímž dochází k redukci

rozměrů vstupu jako je šířka a výška. Výstupní vrstva je plně propojená vrstva, která je podobná finální vrstvě neuronové sítě. [11, 15]

3.3 Detekce pomocí HOG a SVM

Histogram orientovaných gradientů (HOG) je technika analýzy obrazu, kterou představili *N. Dalal* a *B. Triggs* v [16]. HOG se zaměřuje na charakterizaci vzhledu a tvaru objektů v obraze pomocí distribuce lokálních intenzit gradientů nebo směrů hran, aniž by bylo nutné znát jejich přesnou pozici.
Metoda podpůrných vektorů (Support Vector Machines, SVM) je algoritmus strojového učení, který byl vyvinut *V. Vapnikem* [17]. SVM se primárně používá pro klasifikaci dat tím, že identifikuje nadrovinu, která odděluje prostory do dvou tříd. Tato nadrovina se snaží maximalizovat rozestup mezi třídami, aby se dosáhlo co nejpřesnější klasifikace.
Společné použití HOG a SVM při detekci objektů spočívá v tom, že HOG extrahuje charakteristiky z obrazu, které poté SVM využije pro klasifikaci objektů v obraze.

3.3.1 Histogram orientovaných gradientů

Charakterizace vzhledu a tvaru obrazu pomocí distribuce gradientů nebo hran se provádí dělením okna obrazu na menší oblasti (buňky) a pro každou z nich se sestavuje lokální histogram směrů gradientů nebo orientace hran nad pixely buňky. Pixel, jehož magnituda gradientu je v daném směru větší, dává histogramu váhu tomuto směru. Gradient obrazu $\nabla I(x, y)$ je vektorová veličina, která udává změnu intenzity pixelu v prostoru. Výpočet se provede konvolucí obrazu s maskou první derivace. Nejjednodušší masky, jako je jednobodová středová 1D maska, poskytují nejlepší výkon, a proto jsou obecně používány k výpočtu histogramu orientovaných gradientů. Horizontální a vertikální gradienty lze vypočítat následovně:

$$G_x = [-1, 0, 1] \circledast I(x, y) \tag{3.5}$$

$$G_y = [-1, 0, 1]^T \circledast I(x, y) \tag{3.6}$$

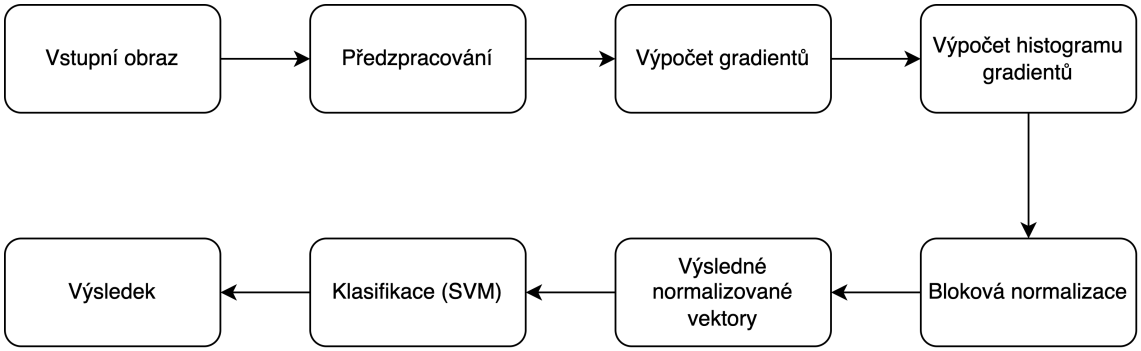
Magnituda G a směr Θ gradientu se pomocí G_x a G_y vypočítá následovně:

$$G = \sqrt{G_x^2 + G_y^2} \tag{3.7}$$

$$\Theta = arctan(\frac{G_y}{G_x}) \tag{3.8}$$

Kombinované složky histogramu reprezentují vstupní obraz. Pro snížení citlivosti na změny světla a stínů jsou buňky seskupeny do bloku pomocí obdélníkové (R-HOG) nebo kruhové (C-HOG) geometrie a následně normalizovány.
Ve své práci Dalal a Triggs [16] vyzkoušeli čtyři metody blokové normalizace:

$$\text{L2-norm} : f = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}} \tag{3.9}$$



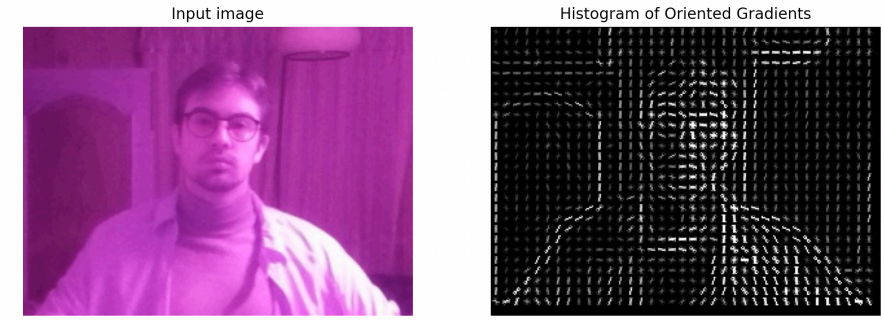
Obrázek 3.7: Popis průběhu detekce objektu pomocí HOG a SVM

L2-hys : L2-norm následovaná oříznutím (omezením maximálních hodnot v na 0,2) a znovu-normalizováním

$$\text{L1-norm} : f = \frac{v}{\|v\|_1 + \varepsilon} \tag{3.10}$$

$$\text{L1-sqrt} : f = \sqrt{\frac{v}{\|v\|_1 + \varepsilon}} \tag{3.11}$$

kde v znázorňuje ne-normalizovaný vektor obsahující všechny histogramy jednoho bloku, $\|v\|_k$ jeho k-tou normu pro $k = 1, 2$ a ε malou konstantu. Z experimentů vyplynulo, že L2-hys, L2-norm a L1-sqrt poskytují podobné výsledky, zatímco výsledky L1-norm jsou méně spolehlivé.
Vysvětlení histogramu orientovaných gradientů bylo převzato z [18].



Obrázek 3.8: Vizualizace HOG.

3.3.2 Metoda podpůrných vektorů

Metoda podpůrných vektorů (SVM) slouží k trénování klasifikátoru, který umí rozlišovat oblasti obsahující obličeje od těch, které obličeje neobsahují. Algoritmus SVM hledá nejlepší separační rovinu mezi oblastmi, která maximalizuje vzdálenost mezi ní a nejbližšími bodům z obou tříd. Tyto body jsou nazývány podpůrné vektory a jsou důležité pro určení hranice mezi oblastmi.

Výsledkem trénování SVM je klasifikátor, který dokáže určit, zda daná oblast obsahuje obličej či nikoliv. Pokud je na vstupu nový obraz, jsou na něm vypočítány HOG deskriptory a pomocí SVM se určí, zda dané oblasti obrazu odpovídá obličej.

3.4 Kódování snímků pomocí neuronové sítě

Pro rozpoznání obličejů je potřeba získat několik základních naměřených prvků každého obličeje. K tomu se natrénuje hluboká konvoluční neuronová síť (sekce 3.2) pro vygenerování 128 naměřených prvků obličeje.
Pro trénování se používá trojice hrubě zarovnaných shodných a odlišných snímků obličeje využívající metodu těžby trojic, tím je zaručena vysoká výkonnost rozpoznávání obličejů pouze s 128 (128-d vektor) bajty na obličej.
Síť je trénuje tak, aby druhé mocniny eukleidovské vzdálenosti mezi 128-d vektory přímo odpovídaly podobnosti obličejů: obličeje stejné osoby mají menší vzdálenosti, zatímco obličeje různých osob mají vzdálenosti delší. Jakmile je síť natrénována, je schopna vygenerovat 128 naměřených prvků u jakéhokoliv obličeje. Samotné rozpoznání se poté řeší k-NN (k-Nearest Neighbors, algoritmus k-nejbližších sousedů) klasifikací.
Z důvodu vysoké časové a hardwarové náročnosti lze na snímky aplikovat již některé existující natrénované konvoluční neuronové sítě.

3.4.1 Algoritmus k-nejbližších sousedů

Algoritmus k-nejbližších sousedů (k-NN) je jednoduchý, ale efektivní klasifikační algoritmus používaný ve strojovém učení. Principem algoritmu je určení třídy neznámého objektu na základě vlastností jeho k nejbližších sousedů v prostoru příznaků. V tomto případě jsou příznaky 128-d vektory, které reprezentují prvky obličeje.
Pro klasifikaci nového obličeje algoritmus k-NN spočítá euklidovské vzdálenosti mezi 128-d vektorem neznámého obličeje a vektory všech známých obličejů. Poté vybere k nejbližších sousedů a určí třídu neznámého obličeje na základě tříd nejbližších sousedů. Pokud je $k = 1$, třída neznámého obličeje bude přiřazena třídě nejbližšího souseda. Pro vyšší hodnoty k je obvykle používána hlasovací metoda, kde třída s největším počtem „hlasů“ mezi k nejbližšími sousedy určuje třídu neznámého obličeje.

4 Návrh systému

Pro úspěšnou implementaci kamerového bezpečnostního systému je důležité, aby byl schopen rychle detekovat obličeje za různých podmínek. Systém by měl být navržen tak, aby umožňoval detekci i za neideálních podmínek, jako jsou špatné osvětlení, rychlé pohyby osoby nebo různé natočení hlavy. Důležitou vlastností systému by měla být možnost oznámení výsledků a jednoduché ovládání s možnostmi zobrazení poslední detekce známého obličeje, neznámého obličeje a osoby nebo objektu, kde nebylo možné obličej nalézt. Systém by měl také umožňovat zobrazení historie detekce a přidání snímků obličejů, které má systém znát.
Důležitou součástí systému by měla být databáze pro uchování pořízených snímků a jiných dat z aplikace. Tato databáze by měla umožňovat ukládání snímků, jmen známých osob a uživatelských logů.
Samotný systém by měl být navržen tak, aby detekce a rozpoznávání obličejů probíhalo bez zásahu uživatele. Systém by také měl být schopen předejít ztrátě dat při výpadku internetového připojení nebo při výpadku proudu elektrické sítě. V případě výpadku proudu by měl systém být schopen samostatného znovu-spouštění.

4.1 Návrh řešení

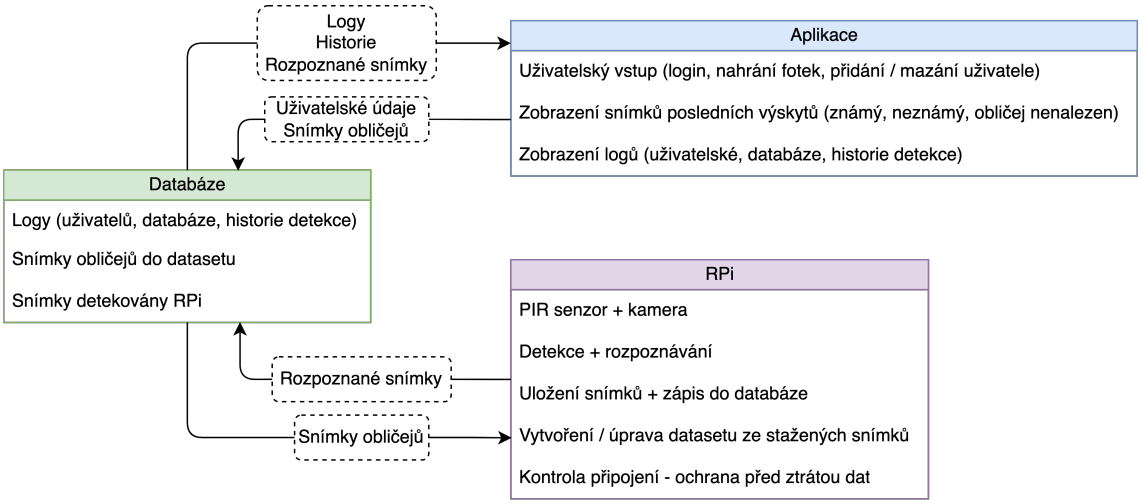
Systém navržený v této práci je složen ze tří hlavních částí, které spolupracují, aby poskytly ucelené řešení pro detekci a rozpoznávání obličejů:

- Minipočítač Raspberry Pi: Tato část systému je základem pro provádění detekce a rozpoznávání obličejů. Raspberry Pi je vybaveno kamerou a pohybovým senzorem, které zajišťují snímání obličejů v reálném čase. Minipočítač také provádí nezbytné výpočty pro rozpoznávání obličejů a umožňuje přidávání nových obličejů do datasetu známých osob. Tímto způsobem je Raspberry Pi jádrem systému, které zpracovává veškerá data získaná z okolí.

- Databáze: Databáze je nezbytná pro uchování informací o detekovaných a rozpoznaných obličejích. Slouží k ukládání snímků, jmen známých osob a uživatelských logů. Databáze také zajišťuje komunikaci mezi Raspberry Pi a webovou aplikací, což umožňuje synchronizaci dat a zajišťuje, že obě části systému mají přístup k aktuálním informacím.

- Webová aplikace: Webová aplikace je zodpovědná za poskytnutí uživatelského rozhraní, které umožňuje jednoduché ovládání celého systému. Aplikace poskytuje funkce, jako je nahrávání snímků pro přidání do datasetu, prohlížení

historie detekcí a rozpoznávání obličejů či správu známých osob. Díky webovému rozhraní mohou uživatelé snadno interagovat se systémem a sledovat jeho výsledky.



Obrázek 4.1: Schéma navrženého systému ilustruje, jak jednotlivé části spolupracují na detekci a rozpoznávání obličejů. Systém se skládá ze tří částí: Raspberry Pi s kamerou a senzorem pro detekci a rozpoznávání obličejů, databáze pro ukládání informací a komunikaci mezi jednotlivými částmi systému, a webové aplikace, která poskytuje uživatelské rozhraní pro ovládání a sledování celého systému.

4.2 Raspberry Pi

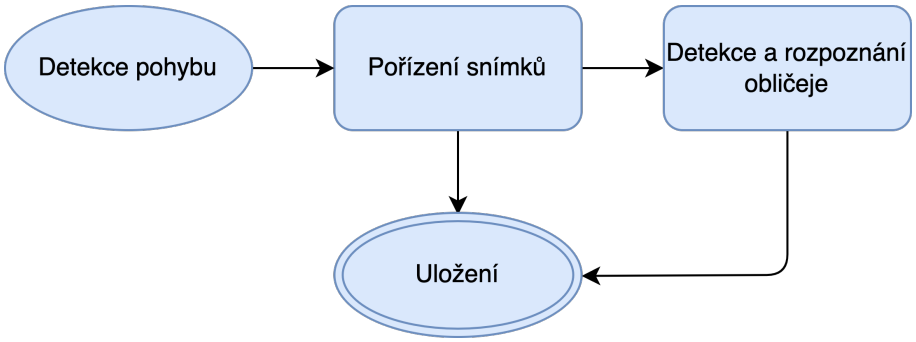
Jako základní prvek kamerového bezpečnostního systému je v této práci navrženo použití Raspberry Pi s kamerou a pohybovým senzorem. Raspberry Pi je miniaturní počítačová deska, která disponuje dostatečným výkonem pro detekci a rozpoznávání obličejů, a zároveň je dostatečně malá a energeticky nenáročná pro instalaci v kamerovém systému. Kamera Raspberry Pi umožňuje snímání videa a fotografií v kvalitě až 1080p. Pohybový senzor detekuje pohyb v dané oblasti a spouští kameru pro pořízení snímků. Raspberry Pi je vybaveno operačním systémem Raspberry Pi OS, který je založen na Linuxu. Výhodou tohoto systému je, že je založen na open-source softwaru, což umožňuje snadnou úpravu a rozšiřování systému.
Nejdůležitějším prvkem Raspberry Pi je však aplikace složená ze dvou částí, která byla vytvořena v jazyce Python s využitím knihoven jako OpenCV, dlib a face recognition, piexif, pickle a Numpy.
První část zahrnuje snímání obrazu pomocí kamery a jeho zpracování detekčním a rozpoznávacím algoritmem. Druhá část aplikace se zaměřuje na komunikaci s databází, získávání nových obličejů pro rozšíření datasetu známých obličejů, odesílání výsledků z detekovaných a rozpoznaných snímků, monitorování připojení k databázi a obnovovaní spojení v případě výpadku.

Celá aplikace také zajišťuje ukládání pořízených dat, aby nedošlo ke ztrátě informací v případě výpadku napájení.

4.2.1 Detekce a rozpoznávání

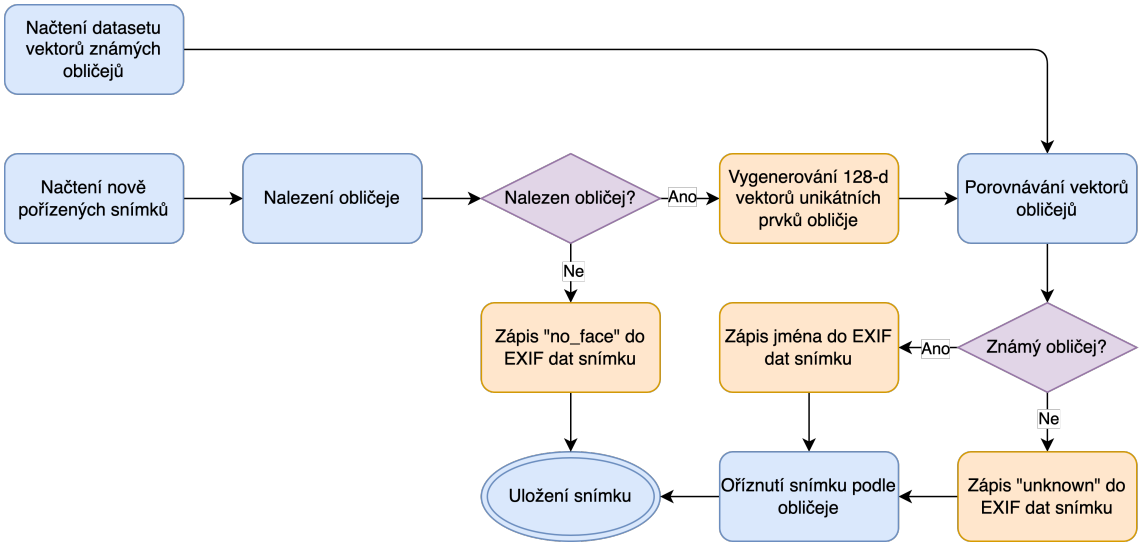
Tento proces začíná detekcí pohybu, na kterou reaguje pořízením snímku. Snímek je následně zpracován rozpoznávacím algoritmem, který identifikuje obličeje na snímku. V případě, že nebyl nalezen žádný obličej na snímku, je snímek zaznamenán do systému se stavem "no face". Pokud byl obličej na snímku nalezen, aplikace se pokusí jej rozpoznat z datasetu známých obličejů. Pokud se nepodaří obličej identifikovat, je do systému zaznamenán snímek se stavem "unknown". V opačném případě je k obličeji přiřazeno jméno známého a výsledek je uložen do systému.

Při výběru vhodného algoritmu pro detekci obličejů byla hlavním kritériem přesnost detekce. Algoritmy byly nejprve podrobeny testování, ještě než byly začleněny do systému. Výsledky testů (viz sekce 6.2) ukazují, že detektor Viola-Jones neposkytuje dostatečnou přesnost a nesplňuje tak požadavky tohoto projektu. Navržený systém pracuje s pořízenými snímky uloženými v databázi, a pokud by byl použit detektor Viola-Jones, databáze by obsahovala velké množství anomálií, tedy snímků, na kterých byl obličej nesprávně detekován, i když ve skutečnosti na snímku nebyl.



Obrázek 4.2: Postup první části aplikace

Zohlednění těchto zjištění vedlo k výběru metod HOG a CNN pro detekci obličejů, jelikož tyto metody nabízejí vyšší přesnost a stabilitu v rámci navrženého systému. Důvodem pro zamítnutí Viola-Jones detektoru byla také uspokojivá rychlost detekce obličejů pomocí metody HOG. Při porovnání doby detekce obličejů na snímcích byly rozdíly mezi Viola-Jones a HOG minimální, což zdůraznilo důležitost dosažení vyšší přesnosti detekčních metod.



Obrázek 4.3: Detailní schéma detekce a rozpoznávání obličeje

Popis postupu detekce a rozpoznávání obličeje je uveden v obrázku 4.3.:

1. RPi detekuje pohyb za pomoci PIR senzoru, pokud je pohyb detekován, RPi spustí kameru.
2. Kamera se spustí a pomocí knihovny OpenCV pořídí 5 snímků v jednosekundových intervalech. EXIF data snímků jsou upravena pomocí knihovny piexif pro zaznamenávání data a času pořízení a snímky jsou uloženy v systému.
3. Detekční algoritmus HOG nebo CNN z knihovny face recognition je aplikován na snímky.

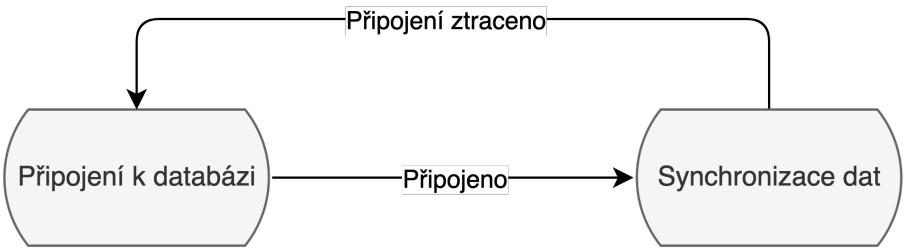
4.a Pokud nebyl nalezen obličej, do EXIF dat snímku se zapíše „no face“.

4.b Pokud byl obličej nalezen, pomocí již natrénované neuronové sítě z knihovny face recognition se získá 128-d vektor naměřených prvků obličeje.

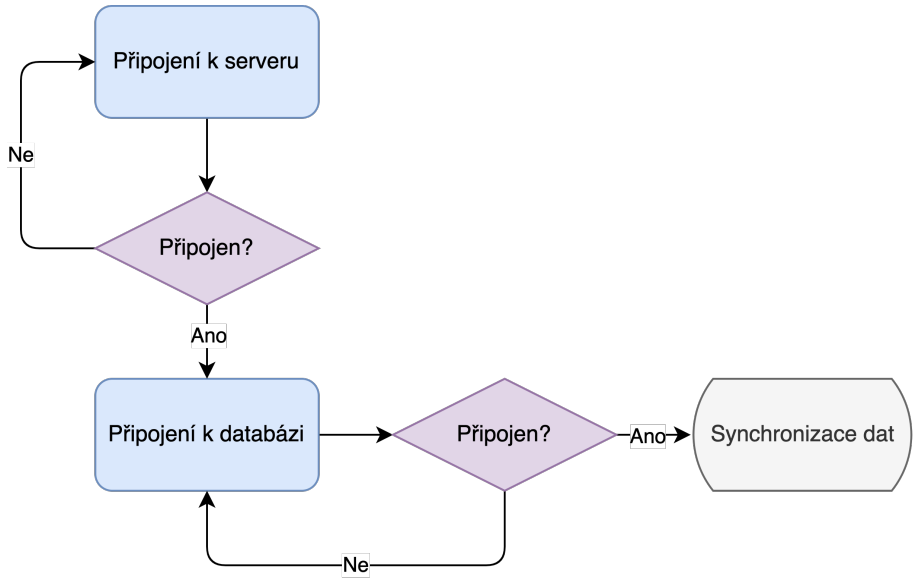
5. Nově získané vektory se porovnají s vektory již známých obličejů v datasetu pomocí Eukleidovské vzdálenosti. Z vektoru známého obličeje nejblíže nově získanému vektoru se získá jméno osoby.
6. Výsledné jméno nebo „unknown“ se zapíše do EXIF dat snímku.

4.2.2 Komunikace s databází

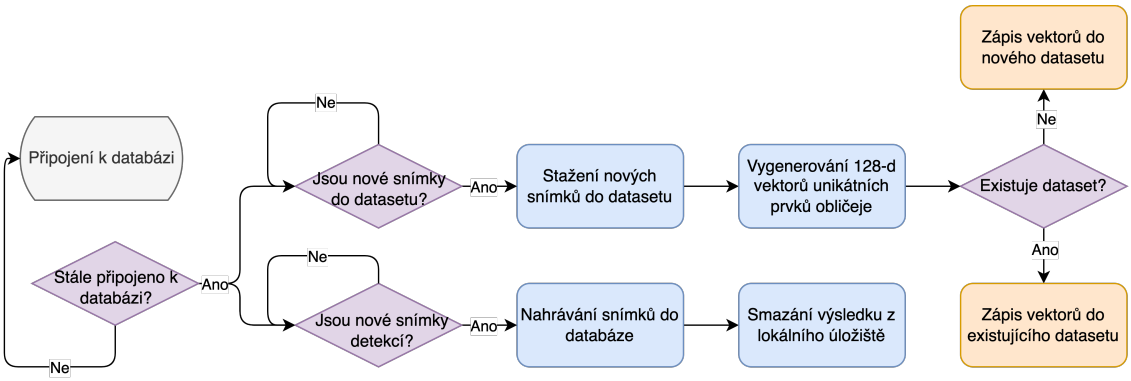
Tato část aplikace se zaměřuje na komunikaci s databází, získávání nových obličejů pro rozšíření datasetu známých obličejů, odesílání výsledků z detekovaných a rozpoznaných snímků, monitorování připojení k databázi a obnovovaní spojení v případě výpadku.



Obrázek 4.4: Postup druhé části aplikace



Obrázek 4.5: Diagram cyklu připojení k databázi.



Obrázek 4.6: Diagram cyklu synchronizace dat.

Popis diagramu na obrázcích 4.5 a 4.6:

1. Aplikace začína v cyklu, ve kterém se pokouší o připojení k databázi. Pokud se připojení nezdaří, aplikace se pokusí znovu po určitém časovém intervalu.

2. Po úspěšném připojení k databázi, aplikace vstoupí do druhého cyklu, který se zaměřuje na kontrolu připojení, nahrávání a stahování dat.

3. Aplikace kontroluje připojení k databázi. Pokud dojde k výpadku připojení, aplikace opustí druhý cyklus a vrátí se do prvního cyklu, kde se znovu pokusí o připojení k databázi.

4. Pokud je připojení stále aktivní, aplikace nahrává do databáze pořízené, detekované a rozpoznané snímky s výsledky, pokud jsou dostupné v systému.

5. Aplikace kontroluje databázi pro nové snímky obličejů. Pokus jsou v databázi nové snímky obličejů, aplikace je stáhne.

6. Pro každý stažený snímek obličeje aplikace získá 128-d vektor naměřených prvků obličeje pomocí již natrénované neuronové sítě z knihovny face recognition.

7. Získané 128-d vektory se přidají do datasetu známých obličejů, čímž se rozšiřuje dataset a zlepšuje schopnost aplikace rozpoznávat obličeje.

8. Aplikace cyklicky opakuje kroky 3 až 7 v druhém cyklu, dokud nedojde k výpadku připojení, poté se vrátí do prvního cyklu.

4.3 Databáze

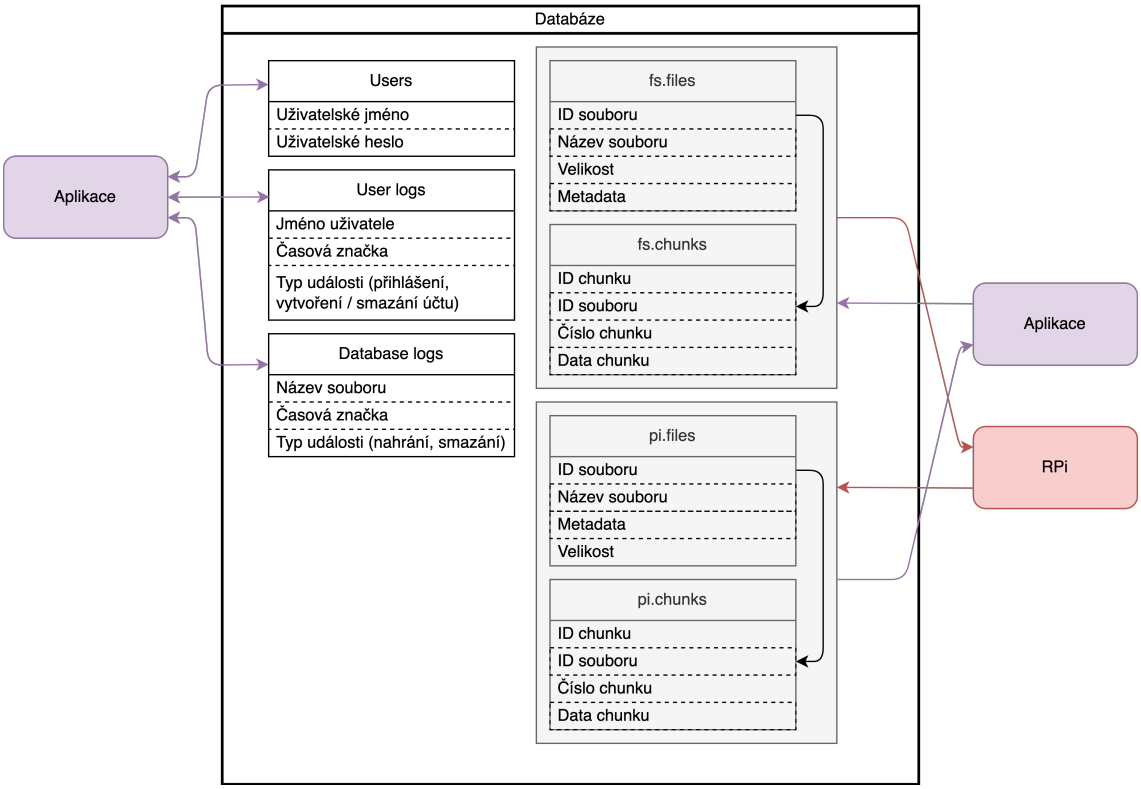
Databáze je další zásadní součást celého systému pro uchovávání, zpracování a sdílení informací mezi jednotlivými komponenty. Jako úložiště pro data slouží NoSQL databáze MongoDB, která je široce používaná. MongoDB byla zvolena díky její schopnosti ukládat velké množství dat ve formátu BSON, což je binární reprezentace formátu JSON. Tento formát umožňuje efektivní ukládání a načítání dat, zatímco poskytuje snadnou manipulaci a dotazování.

Databáze obsahuje následující kolekce:

- Uživatelská data: Tato kolekce obsahuje informace o uživatelích webové aplikace, jako jsou uživatelská jména a hesla. Umožňuje přístup k aplikaci a zajišťuje, že pouze oprávněné osoby mohou manipulovat s daty a provádět akce v systému.

- Uživatelské logy: Kolekce uživatelských logů uchovává informace o událostech souvisejících s uživateli, jako je přihlášení do aplikace, vytvoření nebo smazání účtů. Tyto záznamy pomáhají sledovat aktivity uživatelů a zajišťují bezpečnost a integritu systému.

- Databázové logy: V této kolekci jsou uloženy záznamy o akcích prováděných v databázi, jako je nahrávání nebo mazání souborů uživateli. Tyto logy umožňují monitorování a audit databázových operací a zajišťují transparentnost systému.



Obrázek 4.7: Popis databáze

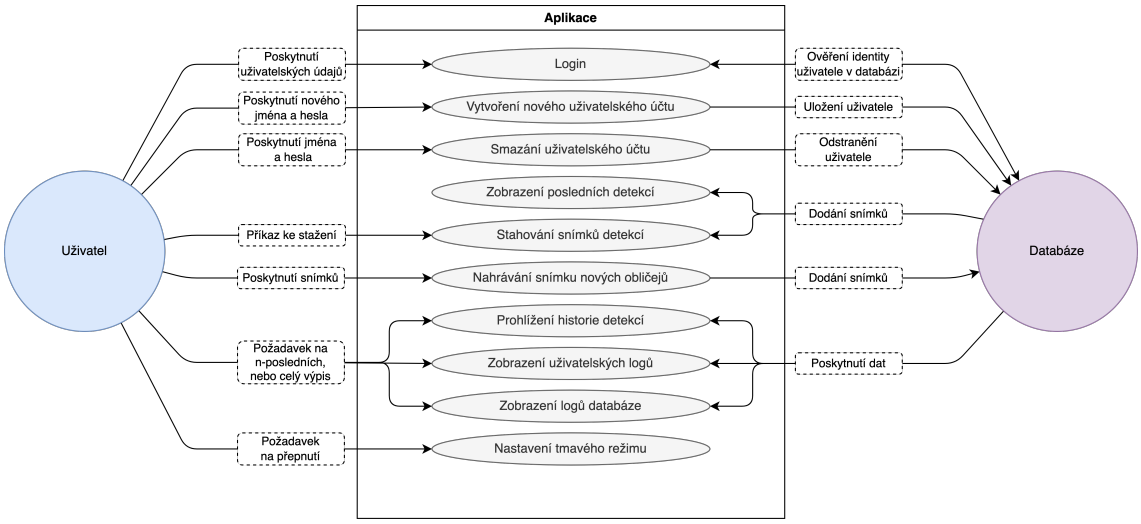
- 1. kolekce GridFS: Tyto kolekce slouží k ukládání souborů nahraných z webové aplikace, jako jsou nové obličeje do datasetu. MongoDB používá GridFS pro ukládání velkých souborů, které jsou rozděleny do menších částí nazývaných „chunks“. Kolekce *fs.files* obsahuje metadata o souborech, zatímco kolekce *fs.chunks* uchovává samotné části souborů. Raspberry Pi čerpá data z těchto kolekcí, což umožňuje průběžné rozšiřování datasetu známých obličejů.

- 2. kolekce GridFS: Tyto kolekce slouží k ukládání dat nahraných z Raspberry Pi, jako jsou detekované a rozpoznané snímky obličejů s výsledky. Stejně jako v předchozím případě, MongoDB využívá GridFS pro ukládání těchto souborů, které jsou následovně dostupné pro webovou aplikaci.

4.4 Aplikace a webserver

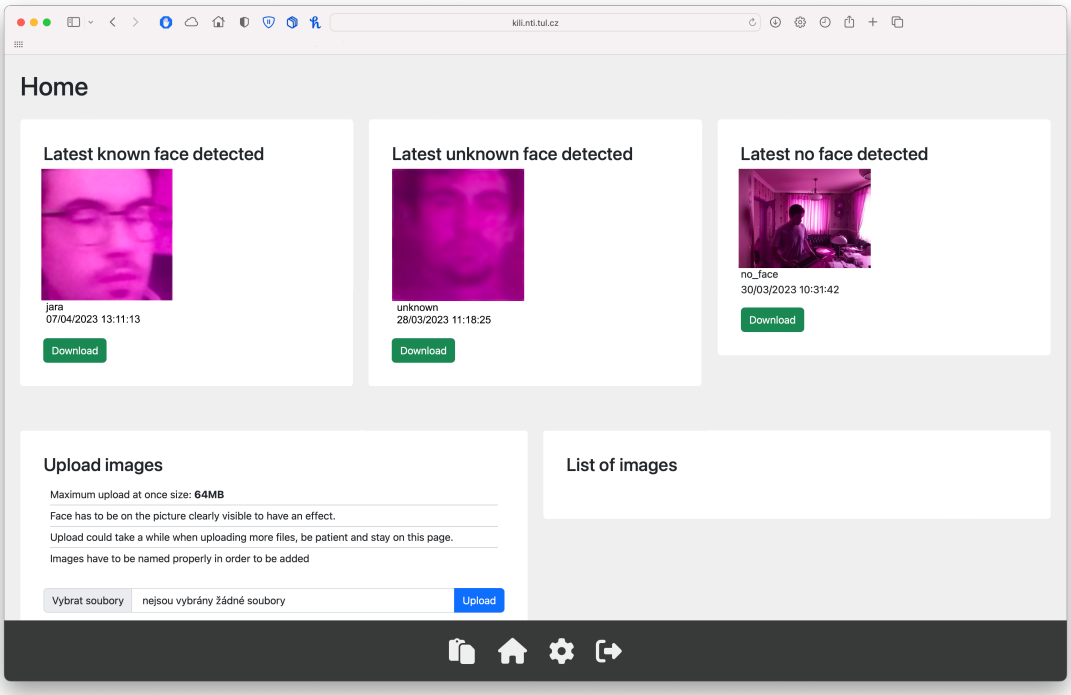
Webová aplikace je třetí klíčovou částí navrženého sytému pro detekci a rozpoznávání obličejů. Je vyvinuta v jazyce Python s použitím frameworku Flask. Aplikace slouží jako uživatelské rozhraní, které umožňuje snadnou interakci s celým systémem a přehledné sledování jeho výsledků. Poskytuje řadu funkcí, které usnadňují správu systému, přidávání nových obličejů do datasetu, prohlížení historie detekcí a rozpoznávání obličejů či správu známých osob.
Aplikace je obsluhována pomocí webserveru Nginx, který je zároveň proxy serverem

pro aplikaci. Nginx je používán pro zajištění bezpečného přístupu k webové aplikaci a zajišťuje, že uživatelé mohou přistupovat k aplikaci pouze přes HTTPS.
Samotný fyzický server používá operační systém CentOS 8, který je založen na distribuci Red Hat Enterprise Linux. Tento operační systém byl zvolen pro svoji stabilitu a bezpečnost.
Webová aplikace poskytuje uživatelům širokou škálu funkcí pro snadnou interakci se systémem. Jednou z funkcí je přihlašování, které umožňuje uživatelům přístup. Aplikace také umožňuje vytvářet nové uživatelské účty pro další osoby, které potřebují přístup k systému, stejně jako mazání existujících uživatelských účtů.
Aplikace nabízí různé možnosti pro zobrazování posledních detekcí, včetně rozpoznaných osob, neznámých osob a objektů, u kterých nebyl nalezen obličej. Uživatelé mají možnost stahovat jednotlivé snímky výsledných detekcí a rozpoznávání pro další analýzu nebo archivaci.

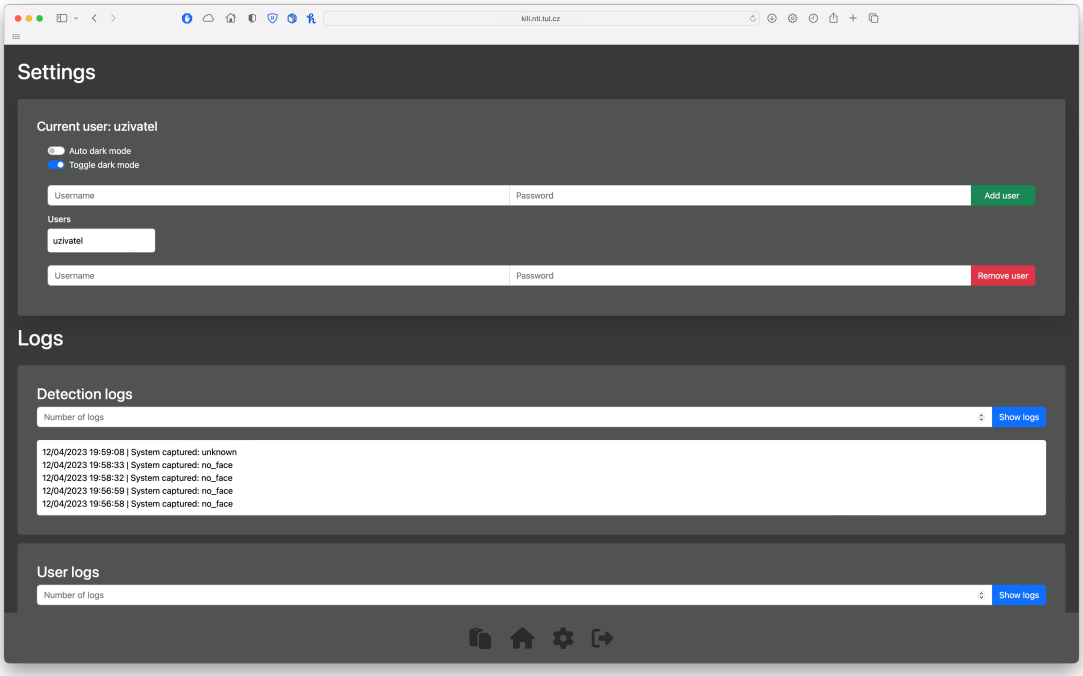


Obrázek 4.8: Popis webové aplikace

Jednou z klíčových funkcí aplikace je nahrávání snímků obličejů pro přidávání do datasetu. Uživatelé mohou nahrávat až 10 snímků najednou, což umožňuje rychle a efektivně rozšiřovat dataset známých obličejů a zlepšovat schopnost systému rozpoznávat obličeje.
Webová aplikace také poskytuje podrobný výpis celkové historie detekcí a rozpoznávání obličejů. Uživatelé mohou zobrazit tuto historii ve formátu JSON, což umožňuje snadné další zpracování dat. Kromě toho aplikace obsahuje logy uživatelů, které zaznamenávají události jako přihlášení, odhlášení, vytvoření a mazání účtů. Dále obsahuje logy databáze, které sledují nahrávání a mazání souborů a který uživatel tyto akce provedl. Tyto logy jsou také dostupné ve formátu JSON pro možnost dalšího zpracování.
Pro pohodlí uživatele aplikace nabízí možnost tmavého režimu, který lze nastavit napevno nebo podle nastavení zařízení.



Obrázek 4.9: Ukázka domovské obrazovky aplikace



Obrázek 4.10: Ukázka logovací obrazovky, otevřeného nastavení, tmavého režimu

5 Zabezpečení

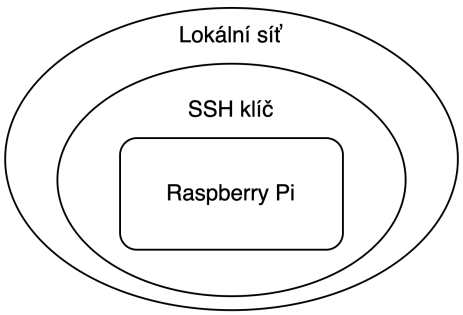
Zabezpečení systému je velmi důležitým faktorem pro jeho funkčnost. Vzhledem k tomu, že systém umožňuje přístup k datům a informacím, jako jsou obličeje, jména, metadata a další, je nutné zajistit, aby byl chráněn před neoprávněným přístupem.
V rámci navrženého systému pro detekci a rozpoznávání obličejů bylo nezbytné zohlednit různé aspekty zabezpečení na úrovni Raspberry Pi, databáze, aplikace a komunikace mezi jednotlivými částmi systému, aby byla zajištěna ochrana těchto dat a soukromí uživatelů.

5.1 Raspberry Pi

V navrženém systému byla v Raspberry Pi provedena řada bezpečnostních opatření, aby byla zajištěna ochrana dat a soukromí uživatelů.

Jedním z takových opatření byly ruční aktualizace operačního systému a nainstalovaného softwaru, což umožňovalo přezkoumat změny v nových verzích a případné dopady na bezpečnost a funkčnost systému před jejich instalací.

Dále bylo změněno výchozí heslo pro uživatele „pi“ na silné a jedinečné heslo, které je těžké uhodnout. Přístup k Raspberry Pi byl zabezpečen prostřednictvím klíčové autentizace místo hesel a bylo zakázáno přihlašování jako root. Přístup k SSH byl omezen pouze na stejnou lokální síť.



Obrázek 5.1: Vizualizace bezpečnostních vrstev Raspberry Pi

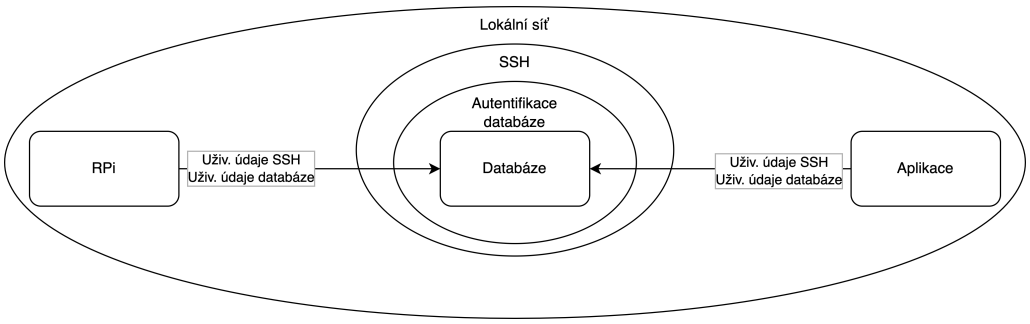
Na Raspberry Pi byl nainstalován pouze nezbytně nutný software a služby, čímž byl snížen povrch pro potenciální útoky. Kromě toho bylo prováděno pravidelné zálohování důležitých dat a konfigurací ve dvou verzích: obraz SD karty systému a fyzická SD karta se systémem. Toto zálohování umožňuje rychlé obnovení systému v případě úspěšného útoku nebo selhání hardwaru. Tato opatření společně přispívají k celkovému zabezpečení Raspberry Pi v rámci detekčního a rozpoznávacího systému.

Aplikace běžící na Raspberry Pi byla navržena tak, aby zamezila ztrátě dat prostřednictvím opatření, jako je průběžné ukládání v mezikrocích, což zajišťuje, že data nebudou ztracena v případě neočekávaného výpadku nebo chyby.

5.2 Databáze

V rámci zabezpečení databáze byla provedena řada opatření. Přístup k databázi byl omezen pouze na lokální síť, Raspberry Pi a aplikace tedy musely být připojené ke stejné síti pro správnou funkčnost systému. Tímto způsobem bylo sníženo riziko neoprávněného přístupu z vnějších zdrojů.

Dále byla pro připojení k databázi vyžadována autentizace pomocí uživatelského jména a hesla. Toto nastavení zajišťovalo, že pouze oprávněné osoby (a aplikace) měly přístup k databázovým zdrojům. Pro připojení mimo server, na kterém byla databáze provozována, bylo nutné použít SSH přístup, který také vyžadoval uživatelské jméno a heslo. Tím byla poskytnuta další vrstva zabezpečení a umožněno sledování a kontrola přístupu k databázi.



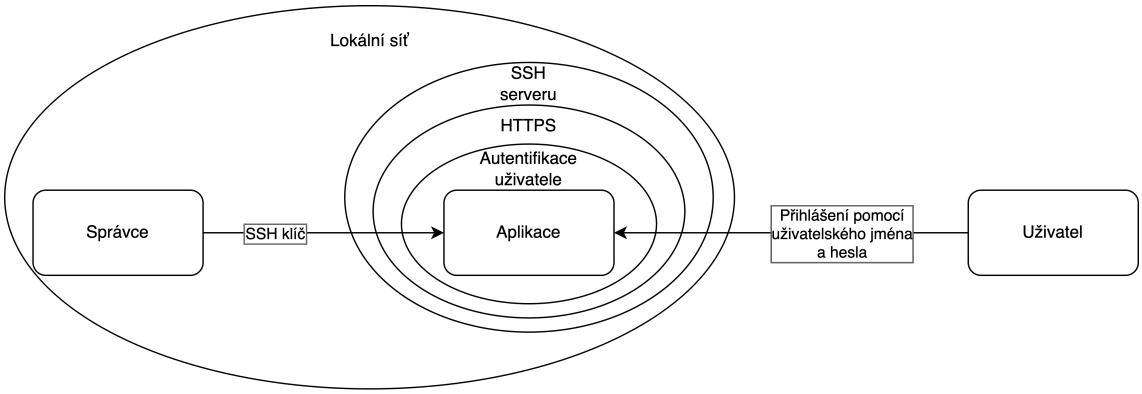
Obrázek 5.2: Vizualizace bezpečnostních vrstev databáze a připojení RPi a aplikace splňující potřebné podmínky

Aby byla zajištěna stabilita a ochrana dat v případě selhání nebo útoku, pravidelně byly prováděny aktualizace a zálohy databáze. Docker kontejner, ve kterém byla databáze provozována, byl také pravidelně aktualizován, což minimalizovalo riziko kompromitace prostředí běhu databáze.

Díky kombinaci těchto opatření byla zabezpečena databáze MongoDB a ochráněna citlivá data.

5.3 Aplikace a webserver

Na fyzickém serveru byla provedena bezpečností opatření zahrnující nastavení uživatelských účtů, SELinuxu, firewallu a nastavení SSH přihlášení pouze pomocí klíče dle [19].
V rámci zabezpečení aplikace bylo použito několik opatření. Aplikace byla provozována na stejném serveru jako databáze a obsluhována webserverem Nginx.



Obrázek 5.3: Vizualizace bezpečnostních vrstev aplikace

Byla uplatněna následující opatření:

- HTTPS a certifikát: Pro zabezpečení komunikace mezi aplikací a uživateli byl zaveden protokol HTTPS a SSL/TLS certifikát na serveru Nginx. Šifrování dat přenášených mezi serverem a uživatelem zajišťuje ochranu citlivých údajů.

- Aktualizace závislostí: Závislosti aplikace byly pravidelně kontrolovány a aktualizovány, aby byly odstraněny potenciální zranitelnosti a zajištěna kompatibilita s nejnovějšími verzemi knihoven.

- Ověřování uživatelů: Byl zaveden systém ověřování uživatelů, který vyžadoval jedinečné uživatelské jméno a silné heslo pro přístup k aplikaci, čímž byla zajištěna ochrana před neoprávněným přístupem.

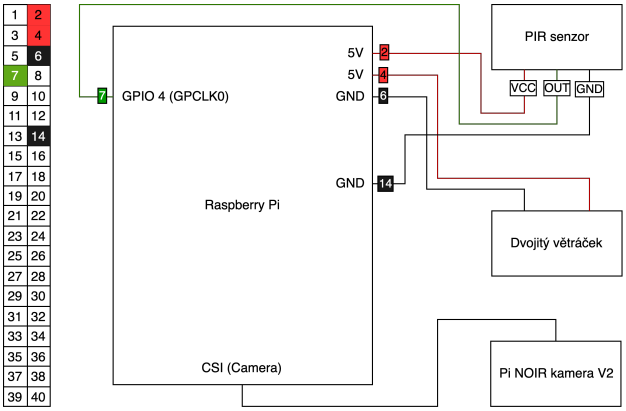
- Nastavení Nginx: Nginx byl nakonfigurován tak, aby poskytl další vrstvu zabezpečení, například omezováním přístupu k citlivým souborům, blokováním nežádoucích požadavků, omezením velikosti souborů nahrávaných uživateli, kontrola vstupních dat, aby se předešlo možným útokům jako SQL injection nebo XSS.

6 Kalibrace a testování

6.1 Zprovoznění Raspberry Pi

Jako základní prvek kamerového bezpečnostního systému je v této práci navrženo použití Raspberry Pi. Při jeho zprovozňování bylo čerpáno z knihy *Simona Monka* [20]. Nejprve byl nainstalován operační systém Raspberry Pi OS (64-bit). Pro bezdrátové připojení k síti se zabezpečovacím protokolem IEEE 802.1X byla použita služba Network Manager. Dále byl na RPi zprovozněn software Samba (reimplementace protokolu SMB) pro pohodlnou vzdálenou manipulaci se soubory. Pro vzdálené ovládání procesů na RPi byl zprovozněn protokol SSH pro správu přes terminál a nainstalován program VNC pro správu přes grafické rozhraní.
Pro sestavení zabezpečovacího systému byly použity následující komponenty:

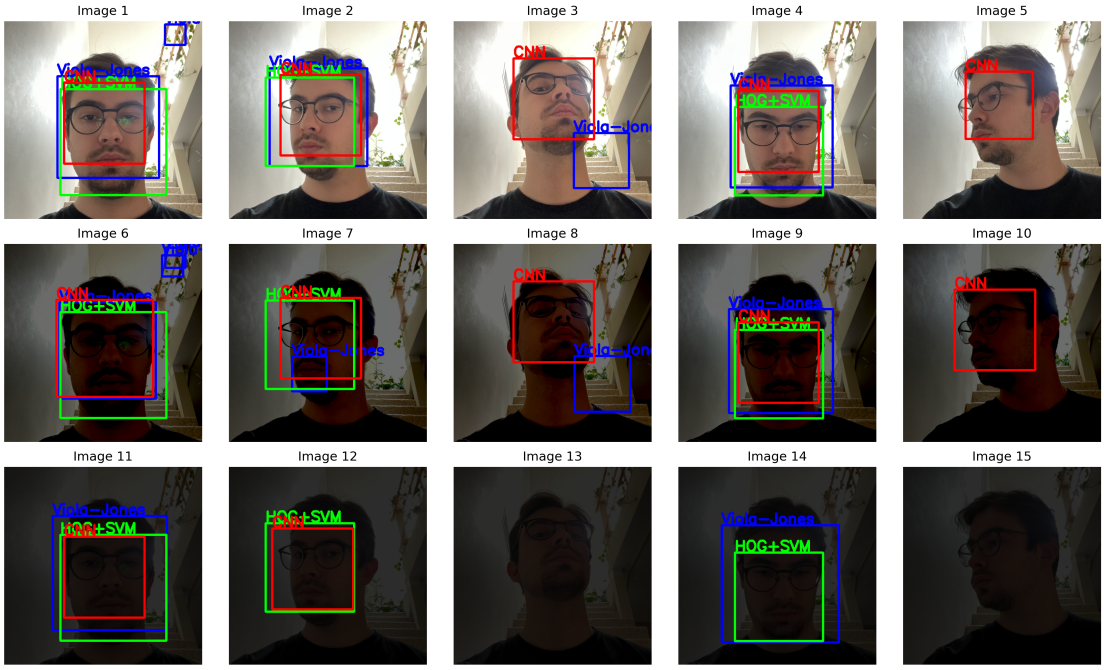
- Raspberry Pi 4 Model B+ (4GB RAM),
- Raspberry Pi NOIR kamera V2,
- PIR senzor,
- MicroSD karta 64GB,
- Heatsink a 5V dvojitý větráček pro chlazení,
- Na míru vymodelovaný 3D tiskem kryt pro RPi, senzor a kameru s kloubem pro jednoduché natočení.



Obrázek 6.1: K Raspberry Pi je připojena Pi NOIR kamera V2 přes CSI port a PIR senzor přes GPIO 4, 5V a GND.

6.2 Porovnání detekčních algoritmů

Před implementací detekčních algoritmů do navrženého systému bylo nezbytné provést jejich testování a porovnání, aby bylo možné zjistit, který z nich poskytuje nejlepší výsledky pro detekci a rozpoznávání obličejů v rámci projektu. Porovnání algoritmů bylo provedeno na základě několika kritérií, která zahrnovala přesnost detekce, rychlost zpracování, robustnost vůči různým světelným podmínkám a náročnost na výpočetní zdroje.
Pro testování a porovnání detekčních algoritmů byl vytvořen testovací dataset, který obsahoval snímky obličejů s pěti různými úhly natočení a třemi světelnými hladinami.



Obrázek 6.2: Ukázka testování tří detekčních algoritmů. Viola-Jones detektor (modrý), HOG (zelený) a CNN (červený).

Z výsledků těchto testů vyplývá, že Viola-Jones detektor bezchybně detekoval obličej pouze v 5 případech z 15 uvedených. Tato úspěšnost je velice nízká pro požadavky systému. Proto byl tento algoritmus vyřazen z finální implementace do systému.

6.3 Testování provozu

Pro testování navrženého sytému byly vytvořeny dvě modelové situace: využití systému pro kontrolu prezence na pracovním místě a kontrola vstupu do místnosti. V obou případech byly otestovány oba detekční algoritmy (HOG a CNN) v kombinaci se stejným rozpoznávacím algoritmem na bázi neuronové sítě. V obou případech

bylo provedeno pro každý detekční algoritmus 10 testů. Při každém testu bylo pořízeno 5 snímků pro zvýšení přesnosti detekce. V situaci *kontrola vstupu do místnosti* byl systém testován ve dvou rozlišení (640×480 a 1920×1080), pro kontrolu prezence na pracovním místě bylo nižší rozlišení (640×480) dostatečné, proto nebylo v tomto případě vyšší rozlišení testováno.
Na pořízených snímcích byla vyhodnocována přesnost detekce a rozpoznání, čas vykonání detekce a rozpoznání, a zároveň bylo kontrolováno správné fungování webové aplikace a databáze, konkrétně zobrazení logů z detekcí a poslední výsledné snímky.

6.3.1 Kontrola prezence na pracovním místě

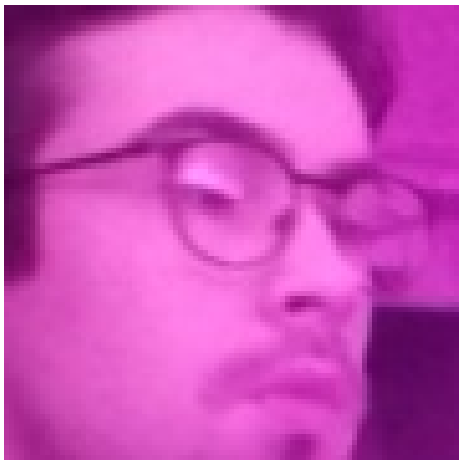
První modelová situace je určena pro kontrolu prezence konkrétní osoby na pracovišti, pracovištěm je myšlen pracovní stůl s počítačem nebo podobné kancelářské zázemí.
Kamera je umístěna na stole, nebo ve výši očí naproti pracující osobě tak, aby byl obličej pracující osoby v záběru PIR senzoru a kamery. Předpokládaná vzdálenost kamery od osoby je přibližně 1 metr, předpokládá se standardní kancelářské stropní osvětlení.
V případě usednutí / opuštění pracovní židle, nebo v případě výraznějšího pohybu (např. protažení se) PIR senzor zaznamená pohyb a systém pořídí 5 snímků. Tyto snímky jsou zpracovány a je zkontrolována přítomnost osoby na pracovním místě. Pokud je osoba na pracovním místě, systém zaznamená čas přítomnosti a zobrazí příslušný log ve webové aplikaci. Pokud osoba opouští pracovní místo, systém zaznamená čas nepřítomnosti a zobrazí příslušný log. Pokud se osoba vůbec nenachází na pracovním místě a současně není na pracovišti žádný pohyb, systém nic nezaznamená.
Situace byla testována pro rozlišení 640×480px. Bylo provedeno 10 příchodů na pracovní místo, během nichž se aktivoval senzor a bylo pořízeno vždy 5 snímků. Na snímky byly aplikovány oba detekční algoritmy (HOG a CNN).

V tabulce 6.1 jsou zobrazeny výsledky detekcí. Úspěšnou detekcí je myšlena správná identifikace známé osoby v případě její přítomnosti a identifikace osoby jako nepřítomné, při její nepřítomnosti.

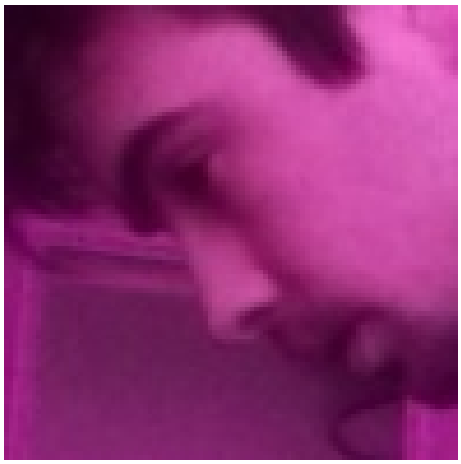
	HOG	CNN
Test 1	5 / 5	5 / 5
Test 2	5 / 5	5 / 5
Test 3	5 / 5	5 / 5
Test 4	5 / 5	5 / 5
Test 5	5 / 5	5 / 5
Test 6	5 / 5	5 / 5
Test 7	5 / 5	5 / 5
Test 8	5 / 5	5 / 5
Test 9	5 / 5	5 / 5
Test 10	0 / 5	1 / 5
Celkem	45 / 50	46 / 50

Tabulka 6.1: Úspěšnost detekce na pracovišti.

Detekčně přesnější metodou je CNN, která dokázala ve všech případech správně vyhodnotit přítomnost osoby na pracovišti a její totožnost. Přesnost metody HOG je stále velmi uspokojivá, nicméně se u ní projevily problémy s nalezením obličeje v případě natočení obličeje o velký úhel.



(a) Obličej nalezen metodou HOG i CNN.



(b) Obličej nalezen pouze metodou CNN.

Obrázek 6.3: Ukázka výsledků detekce metody HOG a CNN.

Růžové zabarvení snímků je způsobeno knihovnou OpenCV pro Python, která nepodporuje ovladač kamery PiCamera NOIR V2 při použití rozlišení 640×480px, pro rozlišení 1920×1080px se tento efekt neprojevuje.
Toto zbarvení však nijak neovlivňuje výslednou detekci obličeje, jelikož se metody aplikují na snímek ve stupni šedi.

6.3.2 Kontrola u vstupu do místnosti

Druhá modelová situace je zaměřena na kontrolu přístupu osob do místnosti, např. kanceláře, laboratoře, vstupní chodby panelového domu nebo jiného uzavřeného prostoru.
V této situaci je kamera se senzorem umístěna v rozmezí výšky očí a 50 cm nad ní. Předpokládaná vzdálenost detekované osoby od kamery je 2 až 3 metry, předpokládá se alespoň minimální osvětlení, nutné pro běžné lidské vidění, zároveň se předpokládá, že obličej osoby je světlejší než pozadí (je nasvětlen zepředu).

(a) Ukázka špatného osvětlení

(b) Ukázka dobrého osvětlení

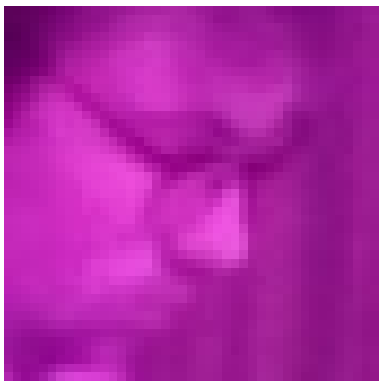
Obrázek 6.4: Ukázka osvětlení.

V případě vstupu osoby do místnosti zaznamená PIR senzor pohyb a systém pořídí 5 snímků. Tyto snímky jsou zpracovány a je zkontrolována přítomnost osoby. Pokud se osoba nachází na snímku, událost je zaznamenána do webové aplikace ve formě logu ve formátu: *čas, datum, jméno osoby* (pokud je osoba známá, jinak „unknown“). Pokud se osoba na snímku nenachází, událost je také zaznamenána do webové aplikace ve formě logu ve formátu: *čas, datum, „no face“*.
Situace byla testována pro rozlišení 640×480px a 1920×1080px. Pro každé rozlišení bylo provedeno 10 vstupů do místnosti, během nichž se aktivoval senzor a bylo pořízeno vždy 5 snímků. Na snímky byly aplikovány oba detekční algoritmy (HOG a CNN).
V tabulce 6.2 jsou zobrazeny výsledky detekcí. Úspěšnou detekcí je myšleno zaznamenání osoby při vstupu do místnosti a její správná identifikace, pokud se nachází mezi známými osobami, nebo její označení jako „unknown“. V případě, že se u vstupu žádná osoba nenacházela, musí být tato situace korektně vyhodnocena a označena kódem „no face“.
Kromě detekcí bylo také testováno rozpoznávání konkrétních osob. Systém byl otestován 2 osobami. Jedna z osob byla uložena v datasetu známých osob (50 snímků), druhá osoba v datasetu uložena nebyla.

	640×480		1920×1080	
	HOG	CNN	HOG	CNN
Test 1	0 / 5	0 / 5	5 / 5	5 / 5
Test 2	4 / 5	3 / 5	3 / 5	5 / 5
Test 3	2 / 5	4 / 5	5 / 5	5 / 5
Test 4	3 / 5	3 / 5	1 / 5	5 / 5
Test 5	3 / 5	3 / 5	4 / 5	5 / 5
Test 6	4 / 5	4 / 5	5 / 5	5 / 5
Test 7	5 / 5	4 / 5	0 / 5	5 / 5
Test 8	4 / 5	3 / 5	5 / 5	5 / 5
Test 9	3 / 5	5 / 5	1 / 5	5 / 5
Test 10	3 / 5	5 / 5	4 / 5	5 / 5
Celkem	31 / 50	34 / 50	33 / 50	50 / 50

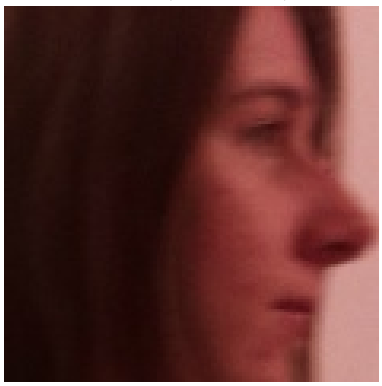
Tabulka 6.2: Úspěšnost detekce u vstupu do místnosti. Šedě zbarvené hodnoty jsou výsledky získané z výpočtu na výkonnějším počítači.

(a) Obličej nalezen metodou HOG i CNN (640×480)



(b) Obličej nalezen pouze metodou CNN (640×480)

(c) Obličej nalezen metodou HOG i CNN (1920×1080)



(d) Obličej nalezen pouze metodou CNN (1920×1080)

Obrázek 6.5: Ukázka výsledků detekce metody HOG a CNN.

6.3.3 Časová náročnost

	640x480px		1920x1080px	
	HOG	CNN	HOG	CNN
Kontrola prezence	5.22 s	3 m 50.48 s	-	-
	5.45 s	3 m 33.29 s	-	-
	4.95 s	3 m 24.81 s	-	-
	5.01 s	3 m 43.83 s	-	-
	5.28 s	3 m 21.71 s	-	-
	5.03 s	3 m 34.12 s	-	-
	5.39 s	3 m 47.95 s	-	-
	5.59 s	3 m 37.09 s	-	-
	5.13 s	3 m 43.00 s	-	-
	5.04 s	3 m 30.98 s	-	-
Kontrola u vstupu	5.40 s	3 m 50.50 s	23.92 s	-
	5.64 s	3 m 38.92 s	23.73 s	-
	5.12 s	3 m 41.19 s	22.98 s	-
	5.19 s	3 m 49.76 s	23.56 s	-
	5.46 s	3 m 28.56 s	24.03 s	-
	5.21 s	3 m 30.58 s	23.47 s	-
	5.58 s	3 m 55.23 s	23.70 s	-
	5.78 s	3 m 44.62 s	22.99 s	-
	5.31 s	3 m 57.49 s	23.61 s	-
	5.22 s	3 m 37.80 s	24.12 s	-

Tabulka 6.3: Časová náročnost celého procesu (od detekce PIR senzorem až po zobrazení výsledků v aplikaci). Chybějící hodnoty nejsou uvedeny z důvodu, že testování nebylo v této konfiguraci na RPi provedeno.

Během testování byla zaznamenávána časová náročnost procesů, vždy od času aktivace PIR senzoru přes detekci, rozpoznávání až po zobrazení výsledků ve webové aplikaci. Odlišnosti v naměřené časové náročnosti byly způsobeny zejména: rozlišením pořízených snímků, fázi synchronizačního cyklu, jehož provedení trvá cca 10 s, a aktuální rychlosti připojení k internetu RPi i serveru.

6.3.4 Zhodnocení provozu

V rámci testování provozu navrženého systému byly vytvořeny dvě modelové situace, na kterých byly otestovány oba detekční algoritmy (HOG a CNN) se stejným rozpoznávacím algoritmem na bázi neuronové sítě. Testování ukázalo, že oba algoritmy jsou schopny detekovat a rozpoznávat osoby s přijatelnou úspěšností, nicméně metoda CNN byla detekčně přesnější.

Reálné výsledky byly téměř totožné s výsledky očekávanými. Webová aplikace se osvědčila při zobrazení výsledků, potřebných informací a možností jednoduchých nastavení jako je přidávání nových obličejů pro rozpoznávání.
Při testování bylo zjištěno, že CNN nebylo možné použít při rozlišení 1920x1080, protože výkon Raspberry Pi nestačil na zpracování tak velkých snímků tímto algoritmem.

7 Alternativy a vylepšení

7.1 Možná vylepšení

V této podkapitole jsou popsána potenciální vylepšení navrženého systému, která by mohla přispět ke zlepšení celkového výkonu a efektivity systému pro detekci a rozpoznávání obličejů. Tato vylepšení se zaměřují na optimalizaci různých aspektů systému, jako jsou rychlost zpracování, přesnost detekce a zvýšení funkcionality.

1. Paralelizace detekce a rozpoznání: Aktuálně systém zpracovává snímky sekvenčně, což může vést k významným prodlevám, zejména při použití algoritmu CNN. Paralelizace detekčního a rozpoznávacího procesu by mohla redukovat čas čekání na zpracování snímků a zlepšit rychlost celého systému.

2. Přechod na C++: I když je Python snadno použitelný a vývojářsky přívětivý, jeho výkon může být v některých případech omezen. Přechod na jazyk C++ by mohl přinést vyšší rychlost a efektivitu systému, zvláště při implementaci paralelního zpracování.

3. Přidání zdroje světla: Pro zlepšení detekce obličejů za špatných světelných podmínek, například v absolutní tmě, by mohl být do systému integrován malý zdroj světla, který by se automaticky zapnul v případě potřeby.

4. Rozšíření funkcí webové aplikace: Aktuálně webová aplikace poskytuje pouze snímky poslední detekce známého obličeje, neznámého obličeje a objektu bez nalezeného obličeje. V případě historie se zobrazí pouze logy s časovými údaji a jménem. Pro zlepšení uživatelského pohodlí by mohla být aplikace rozšířena o možnost zobrazit všechny pořízené snímky, což by umožnilo uživatelům lépe sledovat a kontrolovat činnost systému.

5. Mazání známých obličejů: V současnosti je dataset známých obličejů pouze rozšiřován o nové obličeje. Pro zlepšení správy datasetu a zajištění ochrany osobních údajů by mohla být do aplikace implementována funkce pro mazání dříve přidaných známých obličejů.

7.2 Alternativní řešení

V této podkapitole jsou popsány další možnosti, jak přistupovat k detekci a rozpoznávání obličejů, které by mohly nabídnout lepší výkon nebo různé výhody oproti navrženému systému. Níže jsou uvedeny některé návrhy a doplňující řešení, které by mohly být zváženy:

1. Použití Viola-Jones algoritmu pro real-time video rozpoznávání: Místo čidla a snímání jednotlivých snímků by systém mohl využívat Viola-Jones algoritmu pro detekci obličejů v reálném čase na videu. Tento přístup by mohl zvýšit rychlost a plynulost detekce obličejů, i když s možným kompromisem v přesnosti rozpoznávání.

2. Použití zařízení s lepším hardwarem pro CNN: Pro dosažení lepšího výkonu při použití konvolučních neuronových sítí (CNN) pro rozpoznávání obličejů by mohl být systém implementován na zařízení s výkonnějším hardwarem. To by mohlo zlepšit rychlost a přesnost CNN a umožnit zpracování většího množství snímků za kratší čas. Příkladem takového zařízení by mohl být kit od NVIDIA z rodiny Jetson.

3. Využití externího zařízení pro zpracování dat: Místo lokálního zpracování na Raspberry Pi by snímání mohlo být řízeno například pomocí Arduina nebo jiného podobného zařízení. Snímky by pak mohly být odesílány do výkonnějšího počítače nebo serveru (případně přes cloud) pro další zpracování. Tento přístup by mohl zlepšit rychlost a přesnost rozpoznávání obličejů, avšak může také zvýšit nároky na připojení k internetu a závislost na dostupnosti externího zařízení.

4. Využití cloudových služeb pro detekci a rozpoznávání obličejů: Místo vlastní implementace algoritmů pro detekci a rozpoznávání obličejů by bylo možné využít dostupné cloudové služby, jako je Amazon Rekognition, Google Cloud Vision nebo Microsoft Azure Face API. Tyto služby poskytují předtrénované modely strojového učení pro detekci a rozpoznávání obličejů a často nabízejí vysokou úroveň přesnosti a rychlosti. V tomto případě by Raspberry Pi (nebo Arduino nebo jiné podobné zařízení) pouze získávalo snímky a odesílalo je do cloudové služby pro zpracování. Tento přístup by mohl snížit nároky na výpočetní výkon lokálního zařízení a zjednodušit celkovou implementaci, avšak zároveň zvyšuje závislost na internetovém připojení a může mít dopad na provozní náklady a soukromí uživatelů.

8 Závěr

Předložená bakalářská práce se věnuje návrhu kamerového bezpečnostního systému s rozpoznáváním obličeje. Na základě provedené rešerše komerčních bezpečnostních systémů a projektů využívající detekci a rozpoznávání obličejů byly pro použití v navrhovaném systému zvoleny tři detekční algoritmy: Viola-Jones detektor, histogram orientovaných gradientů (HOG) a konvoluční neuronová síť (CNN). Na základě primárního testování přesnosti detekce byl Viola-Jones detektor vyloučen z finální implementace z důvodu jeho nepostačující přesnosti.
Navržený systém se skládá z minipočítače Raspberry Pi 4 doplněného o PIR senzor pohybu a kameru. RPi je propojeno s webovou aplikací přes databázi, které běží na za tímto účelem vytvořeném serveru. Návrh systému je popsán v kapitole 4. Aplikace umožňuje pohodlné ovládání systému, správu známých osob, které mohou být rozpoznány a přístup k historii detekcí (podrobněji v 4.4).
Systém byl testován v poloreálném provozu ve dvou modelových situacích (6.3). První situací byla kontrola přítomnosti na pracovním místě, ve které byla testována detekce algoritmy HOG a CNN v rozlišení 640×480px. Algoritmus HOG se ukázal jako méně přesný s úspěšností 90 % a časovou náročností asi 5,1 s. Konvoluční neuronová síť dosáhla přesnosti 92 % s nároky na čas zhruba 3 m 37 s. Z testování této situace vyplývá, že algoritmus HOG je pro tuto aplikaci vhodnější. Druhou modelovou situací byla kontrola vstupu do místnosti, při které byly testovány oba detekční algoritmy. Testovací snímky byly pořizovány v rozlišení 640×480px a 1920×1080px. Při nižším rozlišení byl algoritmus CNN (s úspěšností 68 % a časovou náročností cca 3 m 44 s) opět přesnější než algoritmus HOG (s úspěšností 62 % a časovou náročností 5,39 s). V případě rozlišení 1920×1080px se na RPi podařilo úspěšně zprovoznit pouze detekci pomocí algoritmu HOG (s úspěšností 66 % a časovou náročností cca 23,5 s). Ukázalo se, že pro zpracování snímků pomocí algoritmu CNN nemá RPi dostatečný výkon. Snímky s vyšším rozlišením musely být pomocí CNN zpracovány externě na výkonnějším počítači s úspěšností 100 %. Z testování vyplývá, že algoritmus HOG je pro tuto sestavu a aplikaci vhodnější.
Jako důležité vylepšení na základě testování byla autorem navržena paralelizace detekce a rozpoznání, která by snížila časovou náročnost procesu. Dále by mohly být také rozšířeny funkce webové aplikace o odstraňování známých osob nebo o další funkce požadované uživatelem.
V rámci této práce byl úspěšně navržen a implementován systém pro monitorování sledovaného prostoru a rozpoznávání přítomných osob. Výsledný systém může být vhodný například pro použití v malých a středních firmách, kde může sloužit jako levnější alternativa kamerových bezpečnostních systémů.

Seznam obrázků

Seznam tabulek

Bibliografie

[1] KAINZ, Ondrej et al. RASPBERRY PI-BASED ACCESS CONTROL USING FACE RECOGNITION. *Acta Electrotechnica et Informatica*. 2019, roč. 19, č. 4.

[2] ROSENBROCK, Adrian. *Raspberry Pi Face Recognition*. 2018-06-25. Dostupné také z: https://pyimagesearch.com/2018/06/25/raspberry-pi-face-recognition/.

[3] ROSENBROCK, Adrian. *Face recognition with OpenCV, Python, and deep learning*. 2018-06-18. Dostupné také z: https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/.

[4] ROSENBROCK, Adrian. *Face detection with dlib (HOG and CNN)*. 2021-04-19. Dostupné také z: https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/.

[5] RAJ, Aswinth. *Real Time Face Recognition with Raspberry Pi and OpenCV*. 2019-03-29. Dostupné také z: https://circuitdigest.com/microcontroller-projects/raspberry-pi-and-opencv-based-face-recognition-system.

[6] AMOS, Brandon. *OpenFace, Free and open source face recognition with deep neural networks*. Dostupné také z: https://cmusatyalab.github.io/openface/.

[7] VIOLA, Paul a Michael JONES. Rapid Object Detection using a Boosted Cascade of Simple Features. In: SOC, IEEE Comput. (ed.). *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001.

[8] VIOLA, Paul a Michael JONES. Robust Real-Time Face Detection. *International Journal of Computer Vision*. 2003, roč. 57, č. 2.

[9] MAŠEK, Bc. Jan. *Detekce objektů v obraze s pomocí Haarových příznaků*. 2012. Dipl. pr. Vysoké Učení Technické v Brně, Fakulta elektrotechniky a komunikačních technologií.

[10] RAZALI, Haziq. *Haar*. 2015-06-01. Dostupné také z: https://www.researchgate.net/post/Can-someone-answer-a-few-questions-on-Haar-Features-and-Adaboost.

[11] GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[12] GALVEZ, Reagan L. et al. Object Detection Using Convolutional Neural Networks. In: *TENCON 2018 - 2018 IEEE Region 10 Conference*. 2018, s. 2023–2027. Dostupné z DOI: 10.1109/TENCON.2018.8650517.

[13] GURUCHARAN, MK. *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*. 2022-07-28. Dostupné také z: https://www.upgrad.com/blog/basic-cnn-architecture/.

[14] APHEX34. *Typical_cnn*. 2015-12-16. Dostupné také z: https://commons.wikimedia.org/wiki/File:Typical_cnn.png.

[15] LUO, Mingzhu, Yewei XIAO a Yan ZHOU. Multi-scale face detection based on convolutional neural network. In: SOC, IEEE Comput. (ed.). *2018 Chinese Automation Congress (CAC)*. 2018.

[16] DALAL, Navneet a Bill TRIGGS. Histograms of Oriented Gradients for Human Detection. In: SOC, IEEE Comput. (ed.). *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005.

[17] VAPNIK, V.N. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*. 1999, roč. 10, č. 5, s. 988–999. Dostupné z DOI: 10.1109/72.788640.

[18] AHMED, Abdourahman Houssein, Kidiyo KPALMA a Abdoulkader Osman GUEDI. Human Detection Using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction. In: *2017 13th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*. 2017, s. 334–338. Dostupné z DOI: 10.1109/SITIS.2017.62.

[19] TEVAULT, Donald A. *Mastering Linux security and hardening: secure your Linux server and protect it from intruders, malware attacks, and other external threats*. Packt, 2018.

[20] MONK, Simon. *Raspberry Pi cookbook*. O'Reilly, 2016.

9 Příloha: Návod k použití aplikace

Dokumentace - aplikace

Obsah

1. První spuštění

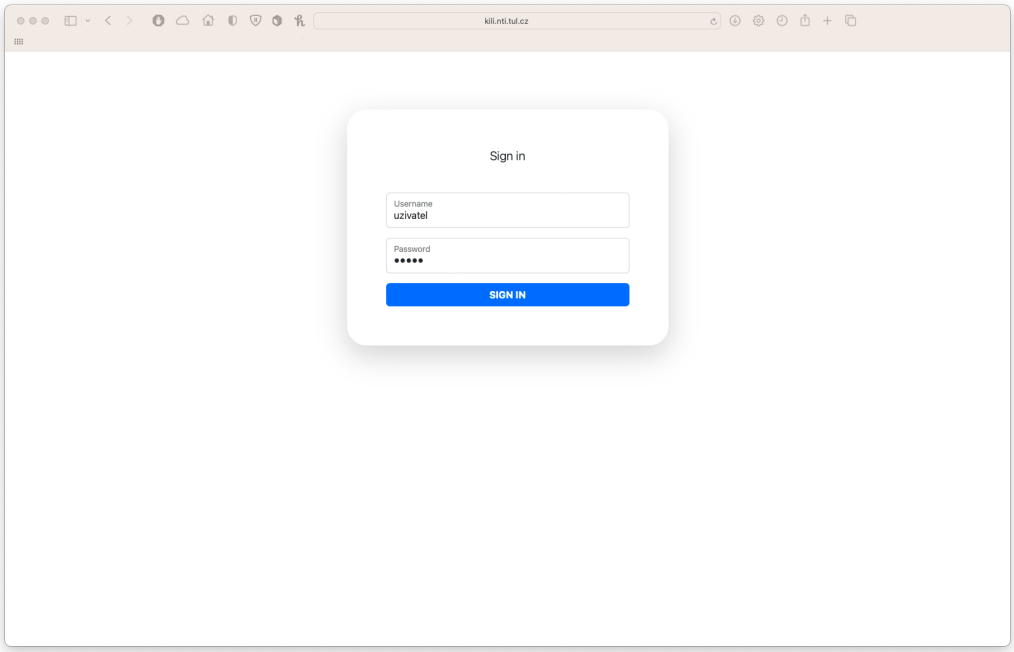
1.1 Přihlášení

Abyste mohli začít používat webovou aplikaci, budete potřebovat přístup k internetu a moderní webový prohlížeč (např. Google Chrome v. 110.0.5464.58 a novější, Mozilla Firefox v. 103.0.5 a novější, Microsoft Edge v. 110.0.5464.58 a novější, Apple Safari v. 15.3.1 a novější).

Postupujte podle následujících kroků:

1. Otevřete svůj webový prohlížeč a v adresním řádku zadejte odkaz „kili.nti.tul.cz" (bez uvozovek) a stiskněte klávesu Enter.

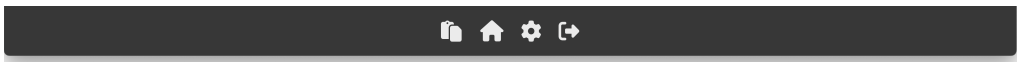
2. Na přihlašovací obrazovce zadejte přednastavené uživatelské jméno a heslo.
   a) Do pole „Username“ zadejte uživatelské jméno „uzivatel“ (bez uvozovek).
   b) Do pole „Password zadejte uživatelské heslo „heslo“ (bez uvozovek).
   c) Klikněte na tlačítko „SIGN IN“ nebo stiskněte klávesu Enter.



3. Aplikace vás přesměruje na domovskou obrazovku. V případě chybného uživatelského jména nebo hesla se zobrazí chybová zpráva a budete muset přihlašovací údaje zadávat znovu. Pečlivě si zkontrolujte, zda v zadaném jméně nebo hesle není chyba.

2. Uživatelské rozhraní

Aplikace obsahuje 2 obrazovky: domovskou a logovací. K přepínání mezi obrazovkami slouží navigační lišta.



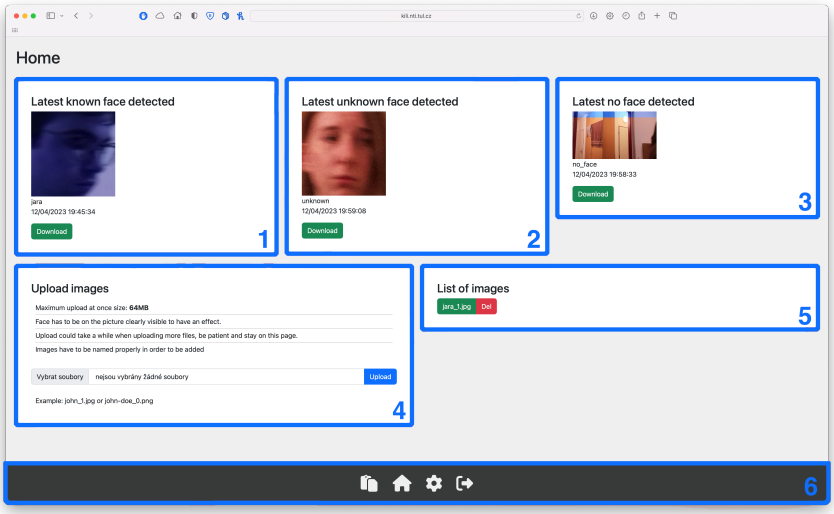
Ikona slouží k přepnutí na domovskou obrazovku.
Ikona slouží k přepnutí na logovací obrazovku.
Ikona slouží k zobrazení nastavení.
Ikona slouží k odhlášení z aplikace.

2.1 Domovská obrazovka

Domovská obrazovka se skládá z 5 dlaždic a navigační lišty.

Na snímku je popsáno:

(1) Dlaždice zobrazení poslední detekce známé osoby,
(2) Dlaždice zobrazení poslední detekce neznámé osoby,
(3) Dlaždice zobrazení poslední detekce s nenalezeným obličejem,
(4) Dlaždice pro nahrávání snímků do aplikace,
(5) Dlaždice zobrazující stav nahrávání snímků
(6) Navigační lišta

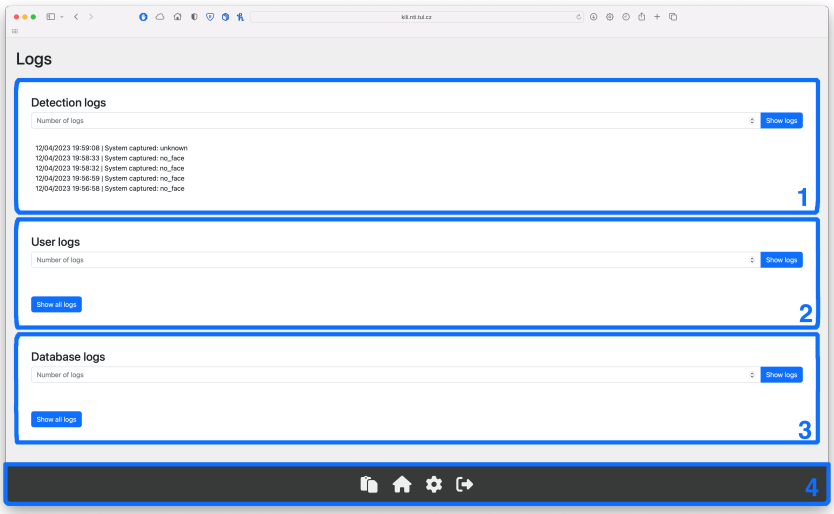


2.2 Logovací obrazovka

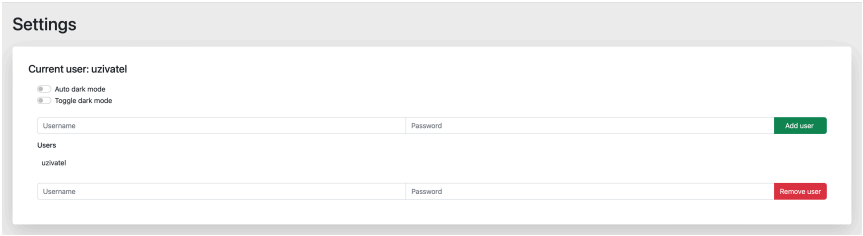
Logovací obrazovka se skládá ze 3 dlaždic a navigační lišty.

Na snímku je popsáno:

(1) Dlaždice pro výpis historie všech detekcí,
(2) Dlaždice pro výpis uživatelských logů,
(3) Dlaždice pro výpis logů databáze,
(4) Navigační lišta



2.3 Nastavení

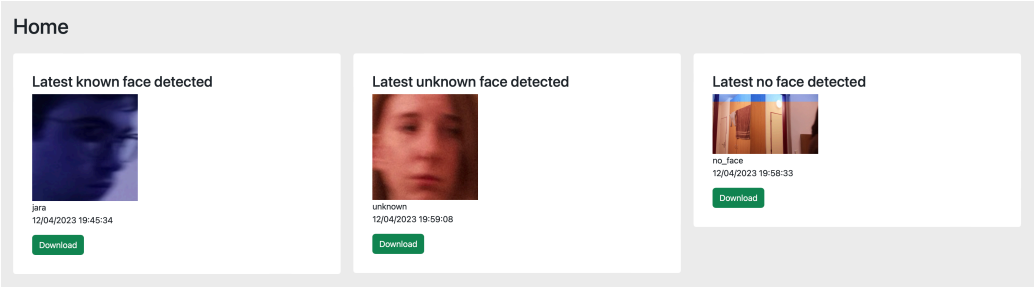


3. Ovládání aplikace

3.1 Zobrazení detekcí

Pro zobrazení poslední detekce, postupujte podle následujících kroků:

1. Přejděte na domovskou obrazovku.
2. V horní části naleznete tři dlaždice. Každá zobrazuje poslední detekci:
   a) Známý obličej,
   b) Neznámý obličej,
   c) Nenalezený obličej.
3. Pod každým snímkem naleznete jméno (nebo „unknown“ či „no_face“), čas a datum detekce.
4. Pro stažení snímků klikněte na tlačítko „Download“ pod příslušným snímkem.

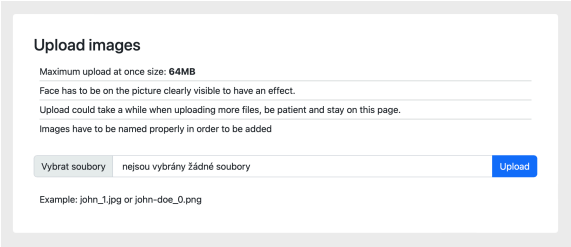


3.2 Přidávání obličejů pro rozpoznávání

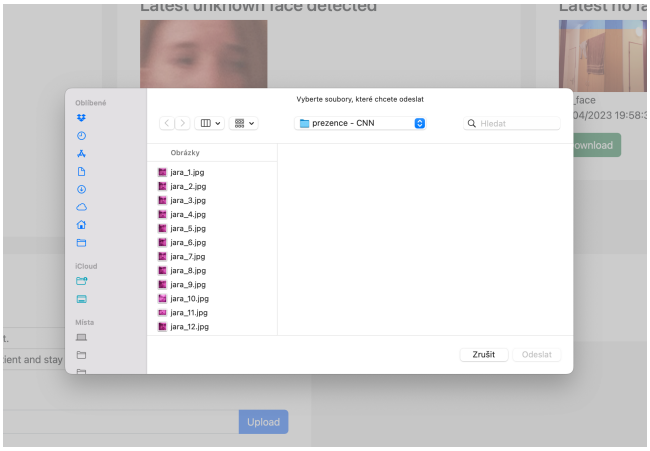
3.2.1 Nahrávání snímků

Pro nahrání snímků postupujte podle následujících kroků:

1. Přejděte na domovskou obrazovku a v levé dolní části na dlaždici „Upload images“ klikněte na pole pro nahrání snímků



2. Otevře se okno prohlížeče souborů. Vyberte snímek nebo snímky, které chcete nahrát (celková velikost nesmí přesáhnout 64 MB).



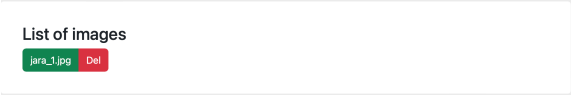
3. Potvrďte výběr ve svém prohlížeči souborů, okno se zavře.
4. Klikněte na tlačítko „Upload“.
5. Dodržujte následující pravidla při nahrávání snímků:
   a) Obličej na snímku musí být dobře viditelný.
   b) Snímky musí být správně pojmenovány ve formátu jméno_číslo.koncovka (např. john_1.jpg nebo john-doe_0.png).

Při nahrávání více snímků najednou vyčkejte, může to chvíli trvat.

3.2.2 Kontrola stavu nahrávání

Pro kontrolu stavu nahrávání snímků postupujte podle následujících kroků:

1. Přejděte na domovskou obrazovku a v pravé dolní části naleznete dlaždici „List of images“.
2. Na dlaždici uvidíte současný stav nahrávání snímků.
   a) Pokud uvidíte seznam nahrávaných snímků, znamená to, že snímky jsou stále v databázi a ještě nebyly přidány do datasetu.
   b) Pokud snímky již ze seznamu zmizely, znamená to, že snímky byly úspěšně přidány do datasetu obličejů pro rozpoznávání.
3. Pokud jsou snímky ještě v databázi (tj. nejsou ještě nahrány datasetu), je možné je snímky smazat kliknutím na tlačítko „Del“ u názvu příslušného snímku.

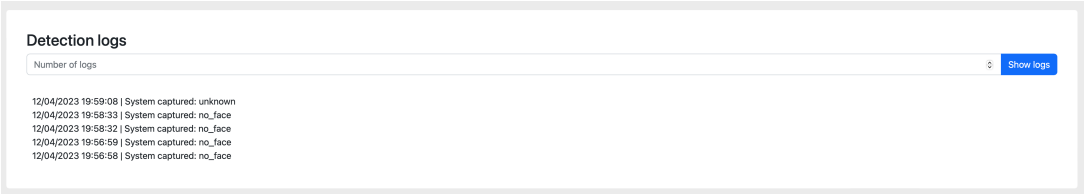


3.3 Zobrazení logů

3.3.1 Historie detekcí

Pro zobrazení historie detekcí postupujte podle následujících kroků:

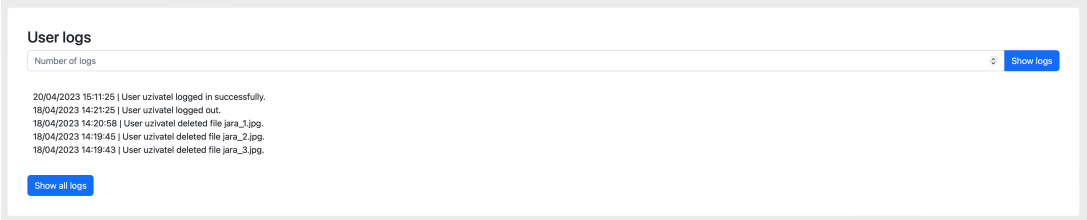
1. Přejděte na logovací obrazovku.
2. Nalezněte v horní části první dlaždici „Detection logs“.
3. Do číselného pole v dlaždici zadejte počet výpisů výsledků detekcí, které chcete zobrazit. Výpisy výsledků detekcí jsou seřazeny časově vzestupně (tj. od nejnovější po nejstarší).
4. Klikněte na tlačítko „Show logs“.
5. Pod číselným polem ze zobrazí výpis výsledků detekcí ve zvoleném počtu. Pokud bude výpis obsahovat méně výsledků, než bylo zadáno, znamená to, že byla vypsána celá historie detekcí.



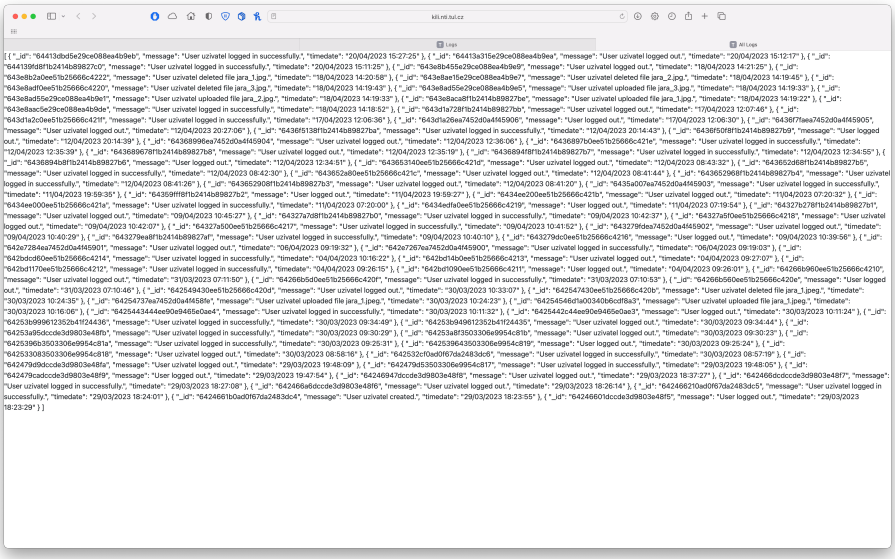
3.3.2 Logy uživatelů

Pro zobrazení logů uživatelů postupujte podle následujících kroků:

1. Přejděte na domovskou obrazovku.
2. V prostřední části najdete druhou dlaždici „User logs“.
3. Do číselného pole v dlaždici zadejte počet logů, které chcete zobrazit. Logy jsou seřazeny časově vzestupně (tj. od nejnovějšího po nejstarší).
4. Klikněte na tlačítko „Show logs“.
5. Pod číselným polem se zobrazí výpis logů ve zvoleném počtu. Pokud bude výpis obsahovat méně logů, než bylo zvoleno, znamená to, že byly vypsány všechny logy.

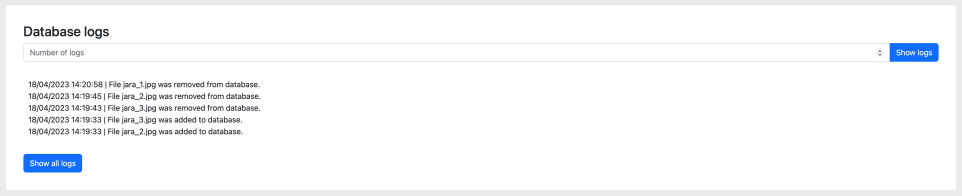


Pro zobrazení všech logů uživatelů:
1. Klikněte na tlačítko „Show all logs“.
2. Otevře se vám nová karta v prohlížeči.
3. V nové kartě se vám zobrazí všechny logy ve formátu JSON.

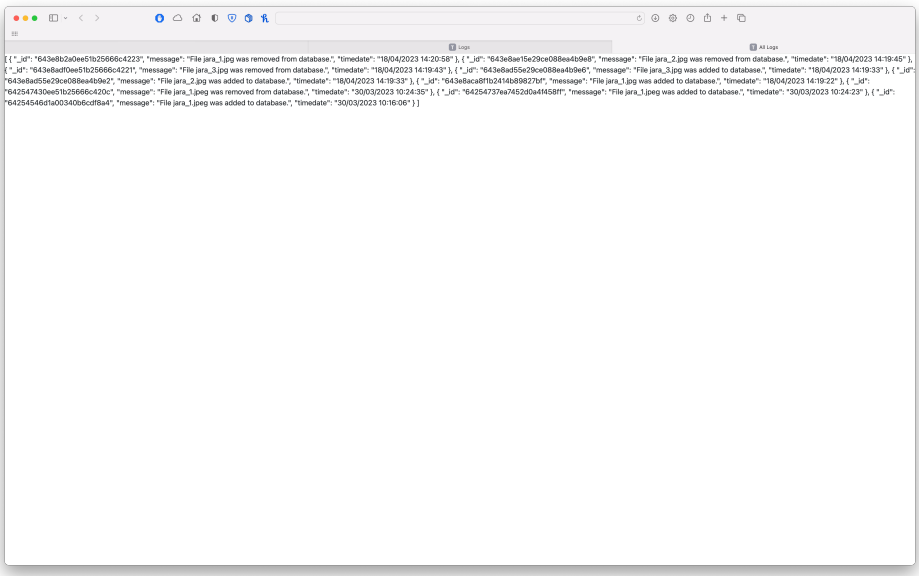


3.3.3 Logy databáze

Pro zobrazení logů databáze postupujte podle následujících kroků:
1. Přejděte na domovskou obrazovku.
2. Ve spodní najdete druhou dlaždici „Database logs“.
3. Do číselného pole v dlaždici zadejte počet logů, které chcete zobrazit. Logy jsou seřazeny časově vzestupně (tj. od nejnovějšího po nejstarší).
4. Klikněte na tlačítko „Show logs“.
5. Pod číselným polem se zobrazí výpis logů ve zvoleném počtu. Pokud bude výpis obsahovat méně logů, než bylo zvoleno, znamená to, že byly vypsány všechny logy.



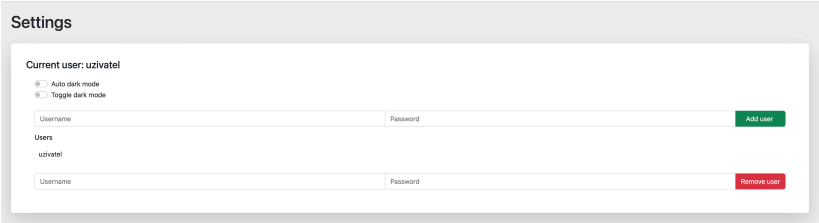
Pro zobrazení všech logů databáze:
1. Klikněte na tlačítko „Show all logs“.
2. Otevře se vám nová karta v prohlížeči.
3. V nové kartě se vám zobrazí všechny logy ve formátu JSON.



3.4 Přidání / odebrání uživatele

Pro přidání uživatele postupujte podle následujících kroků:

1. Otevřete kartu nastavení. Na kartě zvolte horní pár textových polí.
2. Do pole „Username“ zadejte jméno nového uživatele.
3. Do pole „Password“ zadejte heslo nového uživatele.
4. Klikněte na tlačítko „Add user“.
5. Po znovunačtení stránky se uživatel zobrazí na seznamu uživatelů „Users“.



Pro odstranění uživatele postupujte podle následujících kroků:

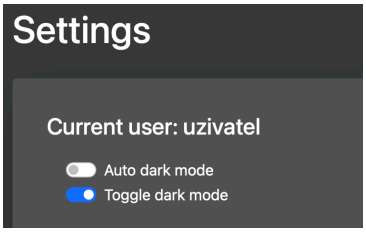
1. Otevřete kartu nastavení. Na kartě zvolte spodní pár textových polí.
2. Do pole „Username“ zadejte jméno uživatele, kterého chcete smazat (nelze smazat momentálně přihlášeného uživatele).
3. Do pole „Password“ zadejte heslo uživatele, kterého chcete smazat.
4. Klikněte na tlačítko „Remove user“.
5. Po znovunačtení stránky se uživatel smaže ze seznamu uživatelů „Users“.

3.5 Změna barevného motivu

Pro nastavení stálého tmavého motivu postupujte podle následujících kroků:

1. Otevřete kartu nastavení.
2. V horní části karty klikněte na spodní přepínač značený „Toggle dark mode“.
3. Aplikace se přepne do stálého tmavého motivu.

Pro nastavení automatického motivu postupujte podle následujících kroků:



1. Otevřete kartu nastavení.
2. V horní části karty klikněte na horní přepínač značený „Auto dark mode“.
3. Aplikace se přepne do automatického motivu podle zařízení (tj. pokud je na zařízení nastaven světlý motiv, aplikace bude používat světlý motiv, pokud je na zařízení nastaven tmavý motiv, aplikace bude používat tmavý motiv).