



FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY



**Bakalářská práce**

# Aplikace pro podporu zjišťování počtu osob na fotografii

Matěj Sochorec





FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY

## **Bakalářská práce**

# **Aplikace pro podporu zjišťování počtu osob na fotografii**

Matěj Sochorec

### **Vedoucí práce**

Doc. Ing. Dalibor Fiala, Ph.D.

© Matěj Sochorec, 2024.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

**Citace v seznamu literatury:**

SOCHOREC, Matěj. *Aplikace pro podporu zjišťování počtu osob na fotografii*. Plzeň, 2024. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Doc. Ing. Dalibor Fiala, Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd  
Akademický rok: 2023/2024

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Matěj SOCHOREC**  
Osobní číslo: **A22B0145P**  
Studijní program: **B0613A140015 Informatika a výpočetní technika**  
Specializace: **Informatika**  
Téma práce: **Aplikace pro podporu zjišťování počtu osob na fotografii**  
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

## Zásady pro vypracování

- Proveďte průzkum trhu dostupných softwarových aplikací pro zjišťování počtu osob na fotografii v digitálním formátu (např. JPG).
- Prostudujte relevantní techniky a metody zpracování digitálního obrazu, které by bylo možno využít pro stanovení počtu osob na fotografii.
- Navrhněte automatickou nebo poloautomatickou metodu pro zjištění počtu osob na obraze v digitálním formátu.
- Implementujte metodu navrženou v předchozím bodě v libovolném programovacím jazyce spolu se vhodným (grafickým) uživatelským rozhraním.
- Realizovanou aplikaci řádně otestujte na reprezentativní sadě netriviálních dat a zhodnoťte dosažené výsledky.



Rozsah bakalářské práce: **doporuč. 30 s. původního textu**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Doc. Ing. Dalibor Fiala, Ph.D.**  
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **29. srpna 2023**  
Termín odevzdání bakalářské práce: **2. května 2024**

L.S.

---

**Doc. Ing. Miloš Železný, Ph.D.**  
děkan

---

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**  
vedoucí katedry

V Plzni dne 29. srpna 2023

# Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 30. dubna 2024

.....

Matěj Sochorec

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

## Abstrakt

Cílem této bakalářské práce je navrhnout a implementovat aplikaci pro podporu zjišťování počtu osob na fotografii. Metoda rozpoznání počtu osob může být plně automatická nebo jen podpůrná. Tato práce nejprve uvádí výběr různých aplikací nalezených průzkumem trhu, které nějakým způsobem zjišťují počet osob na fotografii. Dále prozkoumává relevantní techniky a metody pro zpracování digitálního obrazu. Práce pokračuje grafickým návrhem aplikace, popisem použitých knihoven a vysvětlením konkrétní implementace aplikace. Nakonec je popsán proces testování a co výsledky z tohoto testování znamenají.

## Abstract

The goal of this bachelor's is to design and implement support application for counting crowds of people in photos. The recognition method for counting people can be fully automatic or just supportive. Firstly selection of different applications that count people in photos found with market research is presented in this work. Next it explores relevant techniques and methods for digital image processing. This work continues with showing graphical design, description of used libraries and explanation of specific implementation details of the application. Lastly the testing process is described and the results of said testing are explained.

## Klíčová slova

image processing • počítání osob • gui • opencv • javafx

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Průzkum trhu</b>	<b>5</b>
2.1	Příklady aplikací . . . . .	5
2.1.1	Counting things . . . . .	5
2.1.2	iScanner . . . . .	6
2.1.3	CountThis . . . . .	6
2.1.4	Object Counter . . . . .	9
2.1.5	VCL crowd counting . . . . .	9
2.1.6	DotDotGoose . . . . .	9
2.1.7	Počítání objektů v obrazu aplikace Photoshop . . . . .	14
2.2	Výčet vlastností . . . . .	15
<b>3</b>	<b>Relevantní techniky a metody zpracování digitálního obrazu</b>	<b>16</b>
3.1	Tradiční metody . . . . .	16
3.1.1	Přímé metody . . . . .	16
3.1.2	Nepřímé metody . . . . .	18
3.1.3	Reprezentace vlastností . . . . .	19
3.2	Umělé Neuronové sítě . . . . .	21
3.2.1	Učení neuronových sítí . . . . .	21
3.2.2	Šíření informací neuronové sítě . . . . .	22
3.2.3	Perceptron . . . . .	23
3.2.4	Konvoluční neuronové sítě . . . . .	24
<b>4</b>	<b>Návrh</b>	<b>27</b>
<b>5</b>	<b>Implementace</b>	<b>28</b>
5.1	Model aplikace . . . . .	29
5.2	Struktura projektu . . . . .	29
5.3	Překlad . . . . .	31
5.4	Formát exportovaného souboru . . . . .	32

---

5.4.1	general . . . . .	32
5.4.2	points . . . . .	32
5.4.3	settings . . . . .	32
5.5	Popis tříd . . . . .	34
5.5.1	Main . . . . .	34
5.5.2	Controller . . . . .	34
5.5.3	window.fxml . . . . .	36
5.5.4	DataHandler . . . . .	37
5.5.5	Point . . . . .	37
5.5.6	PointSettingsController . . . . .	38
5.5.7	pointSettings.fxml . . . . .	38
5.5.8	Shape . . . . .	38
5.5.9	ExportMenuController . . . . .	39
5.5.10	exportMenu.fxml . . . . .	39
5.5.11	ExportedDataWriter . . . . .	39
5.5.12	ExportedReader . . . . .	39
5.5.13	FaceRecognition . . . . .	40
5.5.14	DetectionSettingsController . . . . .	40
5.5.15	detectionSettings.fxml . . . . .	40
5.5.16	ControllsController . . . . .	40
5.5.17	controlls.fxml . . . . .	41
5.5.18	AboutController . . . . .	41
5.5.19	about.fxml . . . . .	42
<b>6</b>	<b>Testování</b>	<b>43</b>
6.1	Struktura testování . . . . .	43
6.2	Demografika testerů . . . . .	45
6.3	Fotografie testovací sady . . . . .	45
6.4	Výsledek testování . . . . .	48
<b>7</b>	<b>Závěr</b>	<b>51</b>
	<b>Bibliografie</b>	<b>57</b>
	<b>Seznam obrázků</b>	<b>59</b>
	<b>Seznam tabulek</b>	<b>61</b>

Znalost počtu přítomných osob na určité akci nebo shromáždění je překvapivě důležitá informace. Tato informace se hodí organizátorům nebo třeba policii při plánování a regulování budoucích akcí [KMH22]. Bohužel zjištění počtu osob na akcích a shromážděních s volným přístupem je složitá záležitost. Počítání osob v reálném čase je nepraktické a náročné na prostředky, nebo v některých případech i nemožné. Použití aplikace, která by dokázala odhadnout počet osob v dané fotografii by velmi pomohlo s tímto problémem. Následný přístup není náročný na prostředky. Analýzu počtu osob lze provést z pořízené fotografie (např.: shromáždění, akcí, koncertů, demonstrací apod.).

Cílem této práce je vytvoření aplikace, která takto dokáže samostatně nebo s pomocí uživatele spočítat počet osob, které se vyskytují na dané fotografii. Způsob spočítání osob může být automatický nebo poloautomatický. Poloautomatický způsob by měl být schopný uživateli asistovat s manuálním vybíráním osob pomocí intuitivního uživatelského rozhraní, označováním vybraných osob nebo udržováním počtu osob. Automatický způsob by měl být schopný přibližně spočítat počet osob na dané fotografii bez většího zásahu uživatele, pomocí různých technik zpracování obrazu.

V kapitole 2 můžeme vidět provedení průzkumu trhu. Tato část je důležitá z důvodu zjištění dostupných aplikací souvisejících s případem počítání osob na fotografii. Průzkum trhu by mohl odhalit žádoucí i nežádoucí vlastnosti aplikace, stejně jako dostupnost aplikací, které by šly pro tento účel použít.

V další kapitole 3 prozkoumáme relevantní techniky a metody zpracování obrazu, které by bylo možné použít pro stanovení počtu osob na fotografii. V této kapitole si projdeme obecné přístupy tradičních metod pro počítání osob na fotografii, principy umělých neuronových sítí, a nakonec si představíme některé konkrétní metody.

V kapitole 4 si popíšeme metodu pro automatické počítání osob na fotografii. Popíšeme si postup, který tato metoda používá k detekci osob, použité parametry a jak zpracujeme výstup této metody.

V kapitole 5 si vytvoříme návrh automatické nebo poloautomatické metody pro zjištění počet osob na obraze v digitálním formátu. Výběr relevantních technik a korektní návržení metod pro počítání osob bude mít velký dopad na efektivitu a přesnost výsledné aplikace. Následně si popíšeme implementaci této navržené metody v libovolném programovacím jazyce a vhodného odpovídajícího grafického uživatelského rozhraní. Správná implementace uživatelského rozhraní bude důležitá pro intuitivnost aplikace.

V kapitole 6 řádně otestujeme realizovanou aplikaci a popíšeme si postup testování. Dále si vyličíme, jak byli vybráni testeři a jak byla sestavena testovací sada. Nakonec zhodnotíme dosažené výsledky.

V závěrečné kapitole 7 si zrekapitulujeme výsledek implementované aplikace a provedeného průzkumu trhu. Zhodnotíme výsledky testování a co naznačují. Nakonec si popíšeme případná rozšíření aplikace.

Tato kapitola prochází různé počítačové a mobilní aplikace a jejich vlastnosti, které nějakým způsobem zjišťují počet osob. Většina dostupných aplikací není zaměřena na pouhé počítání osob, a tudíž jsou zahrnuté i aplikace, které počítají různé objekty, včetně osob. Mezi těmito aplikacemi najdeme jak automatické, tak i poloautomatické. Automatické aplikace počítají předměty samostatně. Poloautomatické aplikace pomáhají vybírat, počítat a kategorizovat předměty.

Průzkum trhu byl prováděn v říjnu 2022, což znamená že popis aplikací se vztahuje na verze z tohoto období.

## 2.1 Příklady aplikací

### 2.1.1 Counting things

Tato aplikace automaticky detekuje počet věcí podle vybrané šablony. Aplikace je mobilní, nebo na počítač. Demo aplikace obsahuje pět šablon - křížky na bílém papíře, klávesnice, peníze, řádky v sešitu, mezi kterými je i manuální počítadlo. Placená verze obsahuje kolem tří set dalších šablon a lze ji zakoupit ve formě licence. Mezi tyto šablony patří materiál z oblasti stavebního průmyslu (např. instalatérské trubky, cihly), z oblasti dřevařského průmyslu (např. kmeny stromů, latě, bambus) dále umí počítat i zvířata (např. krávy, ovce, ryby, ptáci), mikroskopické obrazy z oblasti zdravotnictví (např. léky, buňky, bakterie), z oblasti zemědělství (např. semena, brouky, vajíčka) a další. Žádná z uvedených šablon bohužel nepočítá osoby. Kvůli vyšší pořizovací ceně licence se aplikace bohužel nehodí pro soukromé nebo méně časté užití. Existuje i verze této aplikace pro počítání z kamer, což je ale hlavně použitelné v továrnách.

Aplikace nabídne možnost *Take new photo*, po které si uživatel následně vybere šablonu. Obrázek lze vyfotit, nebo načíst z galerie. Po načtení obrázku nabídne aplikace možnost *Count* (Počítej) rozpoznání předmětů na fotografii, které potom sečte



a nad obrázkem zobrazí počet. Aplikace umožňuje uložení výsledků do excelové tabulky, do PDF nebo jsou data poslána do nějakého back-end systému.

Výrobce této aplikace je Dynamic Ventures, Inc. Tato aplikace je dostupná na stránkách countingthings<sup>1</sup> pro zařízení se systémem Windows, Android nebo iOS.

## 2.1.2 iScanner

Podobnou funkcionalitu jako aplikace Count Things nabízí i *iScanner*. Tato multifunkční mobilní aplikace je primárně určená pro skenování dokumentů je navíc schopná počítat předměty. Režim počítání je ale dostupný pouze v hrazené verzi, která se nabízí opět ve formě licence. Bezplatná verze funkčnost počítání předmětů vůbec neobsahuje. Tato funkce je nabízená pouze v placené verze a je založena na umělé inteligenci. Přes aplikaci vyfotíte předměty a ta je automaticky spočítá.

Aplikace je vhodná jak pro použití např. při zjišťování stavu materiálu nebo zboží ve skladech, načítání množství knih v prodejnách apod. Výrobce aplikace je BPMobile. Aplikace je dostupná pro Android nebo iOS na webových stránkách iscanner<sup>2</sup>.

## 2.1.3 CountThis

V této mobilní aplikaci lze fotit různé předměty, které pak aplikace automaticky spočítá. Předměty může uživatel sám nafotit nebo si může vybrat obrázky z galerie uložených fotografií. K výběru je několik různých typů předmětů, jako například: trubky, bedny, ovoce nebo dřevěné klády, proto se používá především v oblasti průmyslové výroby, stavitelství nebo logistiky. Lze detekovat i předměty, které nejsou ve výběru, ale přesnost potom není moc vysoká. Díky této vlastnosti lze detekovat i osoby, což je velmi relevantní k naší úloze. Aplikace je dostupná zdarma nebo v hrazené verzi. Pomocí zdarma verze lze oskenovat a uložit až 10 fotografií, při větším počtu fotografií je už vyžadované zakoupení plné verze. Placená verze je ve formě licence.

Výrobce této aplikace je AIBY Inc. Aplikace je dostupná pro Android ke stažení na Google Play obchodě<sup>3</sup>, nebo pro iOS na App Store<sup>4</sup>.

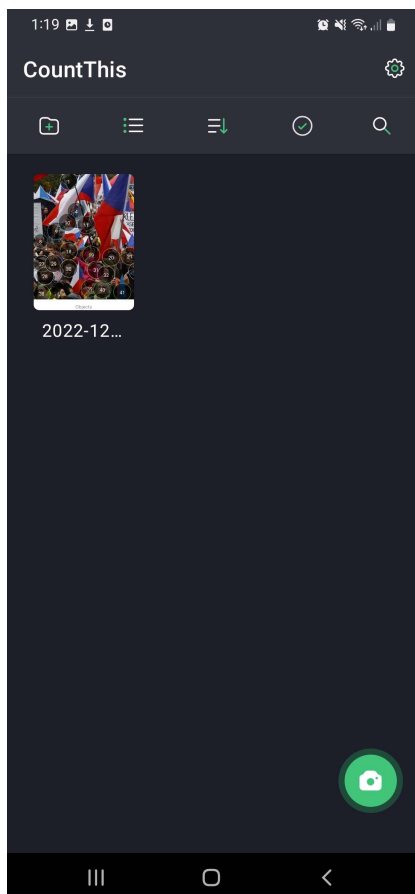
<sup>1</sup><https://countthings.com/>

<sup>2</sup><https://iscanner.com/>

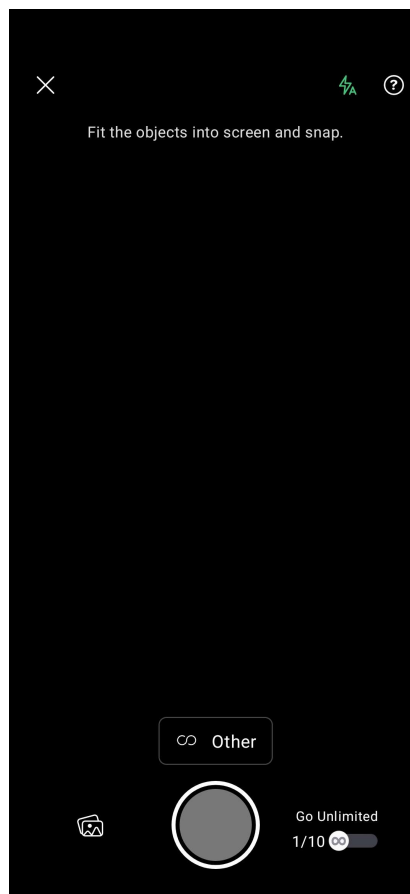
<sup>3</sup><https://play.google.com/store/apps/details?id=com.countthis.count.things.counting.template.counter&hl=en&gl=US>

<sup>4</sup><https://apps.apple.com/us/app/count-this-counting-app/id1553863458>

- Aplikace začíná v hlavním menu, máme několik možností řazení a organizace fotografií, což můžeme vidět na obrázku 2.1(a). Kliknutím na fotoaparát se aplikace dostane na obrazovku fotoaparátu na obrázku 2.1(b)
- Na obrazovce fotoaparátu na obrázku 2.1(b) můžeme vyfotografovat předměty, které chceme spočítat, nebo si ikonou vlevo dole otevřeme galerii fotografií, ze které vybereme snímek.



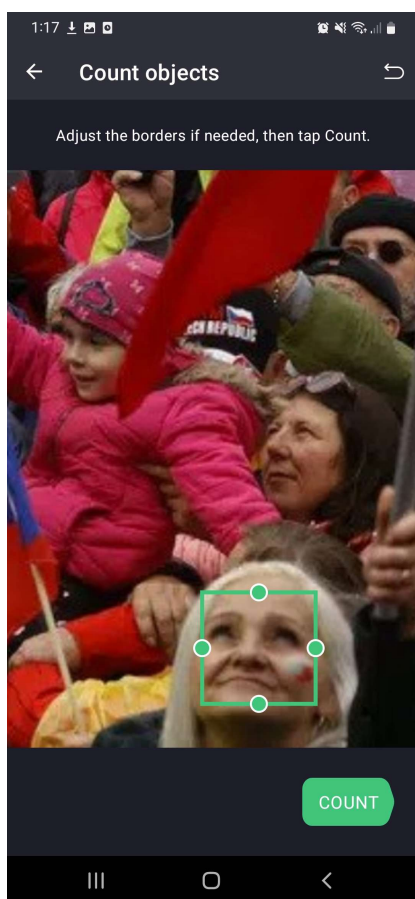
(a) hlavní menu s uloženými počítáními



(b) Fotoaparát s ikonkou galerie fotografií

Obrázek 2.1: Obrazovky fotoaparátu a vybírání předmětu

- V dalším kroku vybereme objekt či subjekt (v našem případě hlavy osob), který slouží jako šablona předmětů, které chceme následně vyhledat. Aplikace nám poté nějaký předmět navrhne, jak můžeme vidět na obrázku 2.2(a).
- Aplikace nám vygeneruje body, které jsou podobné naší šabloně viz Obrázek 2.2(b). Body lze manuálně přidávat, mazat a přemísťovat.



(a) Vybírání předmětu pro počítání



(b) Úprava vygenerovaných bodů

Obrázek 2.2: Obrazovky s úpravou bodů a hlavní menu

- Fotografie s přiřazenými body si pak můžeme uložit k zobrazení v hlavním menu na Obrázku 2.1(a).

## 2.1.4 Object Counter

Další podobná aplikace jako *Count this* nebo *Counting things*. Je to mobilní aplikace, využívá umělé inteligence k počítání objektů. Těchto typů objektů je v aplikaci umístěno cca dvacet (např. dřevěné klády, ptáci, lidské postavy, domy na satelitních snímcích). Tato aplikace je poskytnuta zdarma. Použití aplikace *Object Counter* je časově náročnější z důvodu častého umisťování reklam.

Pro počítání předmětu lze použít fotoaparát, nebo vybrat fotku z galerie. Po načtení fotografie stiskneme tlačítko *Count* a aplikace počet předmětů spočítá. Ukáže se obrazovka s označenými předměty a jejich počtem. Aplikace nabídne možnost uložení počtu do historie.

Výrobce této aplikace je Ocean Float Mobile. Tato aplikace je dostupná na stránkách [oceanmobile<sup>1</sup>](https://oceanfloatmobile.com/) pro mobilní zařízení se systémem Android.

## 2.1.5 VCL crowd counting

Tato webová aplikace dokáže automaticky pomocí konvolučních neuronových sítí odhadnout počet osob na fotografii nebo na video záznamu. Více o konvolučních neuronových sítích popisují v sekci 3.2.4. Aplikace má velmi jednoduché rozhraní, ve kterém je umístěné tlačítko pro nahrání fotografie.

Aplikace je dostupná na webových stránkách [VCL<sup>2</sup>](https://vcl.itl.gr/crowd-counting/).

## 2.1.6 DotDotGoose

Aplikace je určena pro manuální počítání předmětů na fotografiích. Rozhraní je intuitivní, dovoluje libovolné vytváření a upravování tříd předmětů - rozdělení předmětů do barevně kódovaných skupin (např. hlavy osob, kachny na řece, spadané stromy v lese atd.). Rozhraní umožňuje snadno vytvářet a upravovat třídy předmětů. Pomocí nástrojů *pan* (táhnutí myši pro posunutí přiblíženého obrázku) a *zoom* (přiblížení obrázku) umožňuje přesně pokládat body pro označení individuálních předmětů. S uvedenou aplikací lze exportovat metadata použitelná v tabulkách a softwarech pro zpracování statistik.

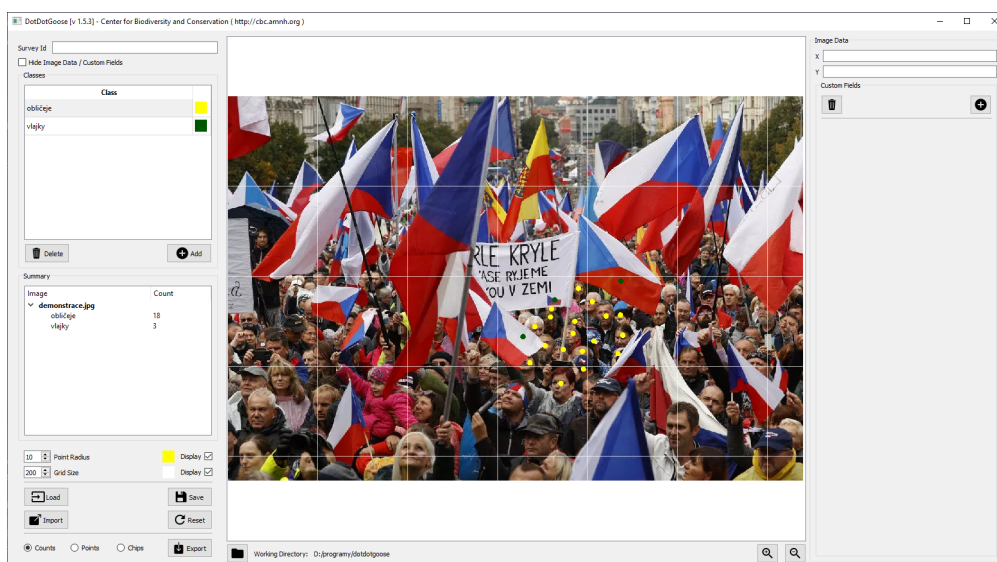
---

<sup>1</sup><https://objectcounter.oceanfloatmobile.com/>

<sup>2</sup><https://vcl.itl.gr/crowd-counting/>

Aplikace je dostupná zdarma ke stažení na webových stránkách muzea historie - *American Museum of Natural History*<sup>1</sup>. Aplikace je také open source (aplikace s volně dostupným zdrojovým kódem), zdrojový kód je zdarma ke stažení na github<sup>2</sup>.

- Aplikace DotDotGoose obsahuje čtyři sekce uživatelského rozhraní, což vidíme na obrázku 2.3. V hlavní části je umístěna fotografie, na kterou se kladou body. V další části jsou umístěny třídy předmětů vlevo nahoře, pod kterou je umístěna část se souhrnem předmětů a možnostmi exportu. V pravé části obrazovky je umístěna část, která umožňuje popsání obrázku metadaty a poznámkami.
- Fotografií můžeme nahrát buď přetažením do okna, nebo výběrem pracovní složky, ze které se nahrají všechny obrázky.



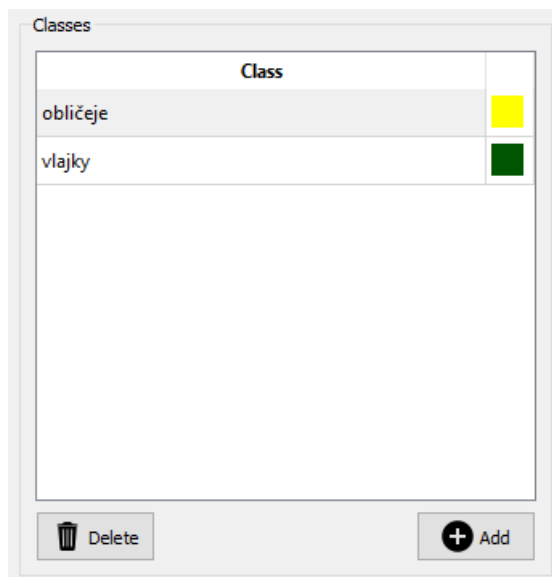
Obrázek 2.3: Okno aplikace DotDotGoose

- Táhnutím myši s držením levého tlačítka lze fotografii posunout, otáčením kolečka na myši nebo použitím tlačítek s lupou ji lze přiblížit.

<sup>1</sup><https://www.amnh.org/research/center-for-biodiversity-conservation/capacity-development/biodiversity-informatics/software-counting-images-open-source>

<sup>2</sup><https://github.com/persts/DotDotGoose>

- Na obrázku 2.4 můžeme vidět seznam vytvořených tříd předmětů (obličej, vlajky a další) kliknutím na *add* tlačítko v horní levé části obrazovky, u jednotlivých tříd lze následně měnit barvu označení.



Obrázek 2.4: Okno s třídami předmětů

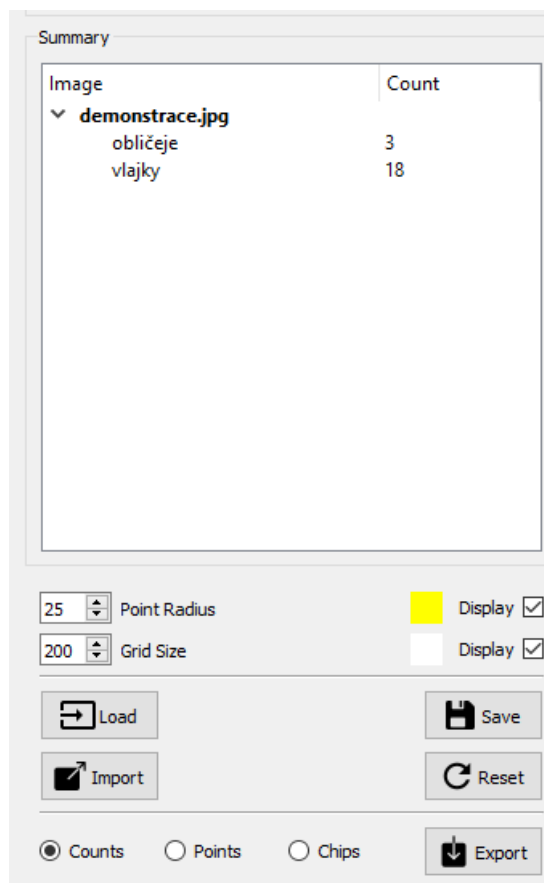
- Aktivní třídu vybereme kliknutím levého tlačítka myši na název této třídy. Všechny přidané body budou patřit vybrané třídě.

- Bod přidáme do obrázku držením klávesy *Ctrl* a kliknutím levého tlačítka myši na předmět, který chceme označit jako na obrázku 2.5.
- Podržením klávesy *Shift* a tažením myši můžeme body označit, stisknutím klávesy *Delete*, pak můžeme označené body vymazat.



Obrázek 2.5: Označování předmětů pomocí bodů

- V okně na obrázku 2.6 můžeme vidět všechny třídy, které jsme v předchozích krocích vytvořili.
- Pod oknem se souhrnem tříd objektů jsou zobrazena tlačítka pro export, načtení a uložení vybraných bodů nebo počtů předmětů.



Obrázek 2.6: Souhrn předmětů a možnosti exportu



## 2.1.7 Počítání objektů v obrazu aplikace Photoshop

*Photoshop* je program pro úpravu bitmapové grafiky s velkým množstvím nástrojů. Jeden z těchto nástrojů nám umožňuje manuální počítání položek na fotografii. Tato funkce je užitečná, ale uživatel musí být zkušený v práci s aplikací *Photoshop*, aby jí mohl efektivně využít.

Pro manuální počítání položek v obrazu musí uživatel vybrat nástroj *Počítání*. Dále může uživatel upravit volby tohoto nástroje, mezi které patří: skupina počítání, barva, velikost značky a velikost popisu. Může se vytvořit více skupin počítání, kde ke každé skupině lze nastavit barvu, velikost značky a popisu. Uživatel může vytvořit a změnit skupiny počítání. Při držení klávesy *Alt* a kliknutím na značku jí uživatel odstraní. Dále lze skrýt nebo zobrazit čísla počtu. Pro značku nebo popis přidá kliknutím do obrazu. Výsledný součet položek lze uložit do souboru.

V aplikaci *Photoshop* je k dispozici také automatické počítání, které se provádí pomocí nástroje *Kouzelná hůlka*. Vytvoříme výběr, ve kterém jsou zahrnuty objekty v obraze, které budeme počítat. Tento nástroj funguje na principu kontrastu barev, je doporučen kontrast s pozadím pro dosažení co nejlepších výsledků. Při zvolení *Vybrat datové body* v záložce *Analýza* z výběru vytvoříme datový bod počítání.

Výrobce tohoto programu je Adobe. Program je dostupný na webové stránce Adobe<sup>1</sup> pro systémy Windows a macOS. Instrukce pro používání nástrojů pro počítání jsou též uvedeny na stránkách Adobe<sup>2</sup>.

---

<sup>1</sup><https://www.adobe.com/products/photoshop.html>

<sup>2</sup><https://helpx.adobe.com/cz/photoshop/using/counting-objects-image.html>

## 2.2 Výčet vlastností

Vlastnosti programů jsou zkompileované do následující tabulky 2.1, kde sloupce reprezentují jméno aplikace a řádky reprezentují konkrétní vlastnosti. Tyto uvedené údaje byly pořízeny v říjnu 2022.

	Counting things	iScanner	Object Counter	CountThis	VCL Crowd counting	DotDotGoose	Photoshop
funkčnost zdarma verze	5 šablon	nedostupno	plná funkčnost	10 fotografií	plná funkčnost	plná funkčnost	nedostupno
cena prémiové verze	2800Kč/měsíc	250Kč/měsíc	---	109Kč/týden	---	---	460Kč/měsíc
počítání z fotografie	ano	ne	ano	ano	ano	ano	ano
počítání z fotoaparátu	ano	ano	ano	ano	ne	ne	ne
manuální počítání	ano	ano	ne	ano	ne	ano	ano
automatické počítání	ano	ano	ano	ano	ano	ne	omezené
open source	ne	ne	ne	ne	ne	ano	ne

Tabulka 2.1: výčet vlastností jednotlivých aplikací

*Funkčnost zdarma verze* udává konkrétní limitace aplikací ve zdarma verzi. *Cena prémiové verze* popisuje cenu uvedenou pro získání licence umožňující přístup k plné verzi programu. *Počítání z fotografie* označuje možnost nahrání obrázku do aplikace, ve které pak aplikace počítá předměty. *Počítání z fotoaparátu* uvádí možnost vyfočení obrázku přímo v aplikaci. *Manuální počítání* vyjadřuje možnost manuálního vybírání/kategorizace předmětů uživatelem. *Automatické počítání* popisuje možnost počítání vybraného předmětu aplikací pomocí algoritmů nebo umělé inteligence. *Open source* označuje, zda má aplikace volně dostupný kód.

Porovnáním nalezených aplikací (Counting things, iScanner, Count This, Object Counter, VCL Crowd counting, DotDotGoose, a nástroj Počítání objektů programu Photoshop) můžeme odvodit vlastnosti, které budou důležité při návrhu naší aplikace. Jednou z těchto vlastností je že výsledná aplikace bude *open-source*, neboli zdrojové soubory budou volně dostupné. I samotná aplikace bude volně ke stažení a zdarma. Na rozdíl od většiny nalezených aplikací by mohla být aplikace zaměřená na počítání osob, tudíž by byla preciznější. Aplikace nebude využívat výpočetního výkonu serveru, což sníží její výkon a přesnost, ale bude vždy k dispozici.

# Relevantní techniky a metody zpracování digitálního obrazu

## 3

Počítání osob na fotografii je problém úzce spjatý s oblastí zpracování digitálního obrazu. V této kapitole si popíšeme některé techniky a metody, které jsou relevantní k této oblasti. Nejprve si v sekci 3.1 vysvětlíme tradiční metody používané k počítání osob v obraze, do nichž spadají přímé (sekce 3.1.1) a nepřímé (sekce 3.1.2) metody. Následně si v sekci 3.1.3 popíšeme reprezentaci vlastností při počítání regresí. Sekci 3.2 si popíšeme základní popis a původ umělých neuronových sítí. V sekci 3.2.1 si popíšeme, jak se tyto neuronové sítě rozdělují podle stylu učení, v sekci 3.2.2 je uvedeno, jak se v nich šíří informace. Dále v 3.2.3 si popíšeme jednoduchý model umělé neuronové sítě *Perceptron*. Nakonec si v sekci 3.2.4 vysvětlíme základy konvoluce a jak by mohla pomoci v naší problematice.

## 3.1 Tradiční metody

Tradiční metody pro počítání osob v davu se dají rozdělit na přímé nebo nepřímé metody. Přímé metody spočívají v rozpoznání lidských vlastností. Nepřímé metody berou dav jako celek a zkoumají jeho vlastnosti, které pak mapují počet lidí v davu.

### 3.1.1 Přímé metody

Přímé metody pro rozpoznání dále dělíme na několik skupin. Nejzřejmější rozdělení je, zda detekujeme celou postavu osob jako *Monolitická detekce* nebo jen části těla. Části těla také můžeme reprezentovat jako elipsy (Porovnávání tvarů). Mezi přímé metody také náleží detekce s více senzory a pro videa můžeme použít počítání shlukováním.

### 3.1.1.1 Monolitická detekce

První skupina se zaměřuje na rozpoznání obecných vlastností člověka. Tento přístup je nejvíc intuitivní, jeho princip vychází z očíslování osob ve scéně detekcí [Loy+13]. Na monolitické detekci je založený běžný systém na detekci chodců. Takový klasifikační systém se trénuje pomocí trénovacích fotografií s celými postavami osob. Natrénovaný klasifikační systém postupně oskenuje celý obrázek pro detekci kandidátů osob. Podle *A survey of crowd counting and density estimation based on convolutional neural network* [Fan+22] metody využívající přístup detekování člověka jako celku, mají dobré výsledky, pokud jde o menší skupinu nebo relativně řídký dav. V případě, že je hustota davu vysoká, výkon výrazně klesne. Důvodem je časté překrývání jednotlivých postav na fotografii.

### 3.1.1.2 Detekce částí těla

Druhá skupina metod zaměřujících se na tento problém počítá osoby pomocí rozpoznávání určitých částí těla, prioritně hlavy a ramena. Samotná oblast hlavy není dostačující kvůli svému tvaru a vzhledu ke spolehlivé detekci osob. Tato metoda díky svému přístupu detekce jen částí těla tolik netrpí na překrývání oproti monolitické detekci.

### 3.1.1.3 Detekce s více senzory

Překrývání osob na fotografii je možné vyřešit začleněním více kamer z různých úhlů. Nevýhodou této metody je potřeba instalace více kamer a takové rozložení není možné v mnoha případech zařídit. Tato metoda má lepší přesnost a často je i rychlejší [Loy+13].

### 3.1.1.4 Porovnávání tvarů

Zhao tuto metodu definoval v *Segmentation and Tracking of Multiple Humans in Crowded Environments* [ZNW08]. Definuje set částí těla parametrizovaných jako složenina elips a stochastickým procesem (proces, který náhodnou pravděpodobností dobře popisuje rozdělení) odhadne konfiguraci elips, která nejlépe popíše fotografii.

### 3.1.1.5 Počítání shlukováním

Počítání shlukováním je založené na tom, že jednotlivé vizuální vlastnosti jsou jednotné, díky čemuž je lze seskupit, aby tvořili prvky, které se pohybují. Tato metoda používá nízkoúrovňové vlastnosti, jejichž trajektorie shlukuje pro odhadnutí počtu osob na scéně. Metoda vyžaduje souvislý pohyb, jinak může nastat nepřesný odhad, když jsou osoby v obraze statické. Počítání shlukováním funguje, jen když

máme k dispozici posloupné snímky. Nevýhoda tohoto přístupu je, že nefunguje se statickými fotografiemi.

## 3.1.2 Nepřímé metody

Největším problémem u přímých metod je překrývání osob na fotografii. Tento problém nepřímých metod je možné řešit pomocí učení mapování z obrázku na počet osob či na mapu hustoty. Prvním krokem je vytažení nízkoúrovňových vlastností určení příznaků, poté se použitím různých regresivních modelů algoritmus učí mapovat příznaky na počet lidí v davu.

### 3.1.2.1 Počítání regresí

Modely založené na regresii se nezabývají prostorovou informací obrazu. Regresní model je takový model, který poskytuje funkci, popisující vztah jednoho či více nezávislých proměnných a výsledné proměnné. Počítání s regresí se vyhýbá dělení jednotlivců a hledání vlastností, odhaduje hustotu davu z příznaků a popisu davu jako celku. Regresní metody jsou schůdné v případech, kdy jsou velmi husté davy a detekce jednotlivých osob je zřejmě omezena. Tento problém je možné řešit mapami hustoty (density maps), které mohou poukazovat na přetížení davu. Mapy hustoty popisují hustotu osob davu v různých částech fotografie. Přestože tento přístup dává dobrý výsledek, metody založené na CNN (konvoluční neuronová síť) je dnes z velké části nahradily [LSF19].



Obrázek 3.1: Variace hustoty davů, vedoucí k různým mapám hustoty [Yua+22]

### 3.1.3 Reprezentace vlastností

Celá další sekce je shrnutím sekce *Counting by Regression* v *Crowd Counting and Profiling: Methodology and Evaluation* [Loy+13].

Pro provedení regrese musíme vyřešit, jak reprezentovat vlastnosti davu na fotografii. Reprezentace vlastností se zabývá vyjmutím, výběrem a přeměnou nízkoúrovňových vlastností na vstup regresní funkce.

#### 3.1.3.1 Vlastnosti popředí

Vlastnosti popředí jsou jednou z nejběžnějších reprezentací pro odhad hustoty davu. Odečtem pozadí získáme segment popředí, ze kterého můžeme odvodit několik vlastností např. celkovou plochu, obvod nebo počet jednotlivých komponent segmentu (oddělené části davu). Složitost tvaru segmentu můžeme změřit rozdílem mezi obvodem segmentu a obsahem.

#### 3.1.3.2 Vlastnosti hran

Segmenty odvozené z davů s nízkou hustotou vykazují hrubé hrany, zatímco segmenty s vysokou hustotou vykazují hladké hrany. Jedním z těchto detektorů hran je *Canny* [Can86]. Z hran segmentu můžeme odvodit tyto vlastnosti: celkový počet pixelů hran, histogram orientací hran segmentu nebo Minowskiho dimenze (fraktální dimenze, dimenze počítání čtverců na hraně), která počítá kolik prvků slouží pro vyplnění hran.

#### 3.1.3.3 Vlastnosti textury

Textura davů nese silnou informaci o počtu osob na fotografii. Davy s vysokou hustotou ukazují silnější reakci, ukazují rozdílnou lokální strukturu s porovnáním s řídkým davem. Vzhled a tvar hlavy nebo ramen může být objeven gradientní mapou intensity, což nese hodně informace o hustotě davu.

Mezi gradientní vlastnosti, používané k odhadu hustoty davu, patří: matice společného výskytu stupňů šedi (GLCM - grey-level co-occurrence matrix), lokální binární vzor (LBP - local binary pattern), gradientní histogram (HOG - histogram of gradients) nebo matice společného výskytu orientace gradientu (GOCM - gradient orientation co-occurrence matrix)[HL10].

GCLM (Matice společného výskytu stupňů šedi) se používá v mnoha studiích počítání davu. Pro získání GCLM rozdělíme obrázek do osmi stupňů šedě a zamaskujeme segmentem popředí. Dále odhadneme pravděpodobnost, že sousední hodnoty

pixelů jsou stejné  $p(i, j|\theta)$ . Tato pravděpodobnost se vypočítá pro čtyři různé orientace ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). Z matice společného výskytu stupňů šedi můžeme následně odvodit pro každé  $\theta$ : stejnorodost (hladkost textury), energie (suma hodnot matic na druhou) a entropie (náhodnost textury).

LBP (lokální binární vzor) je další vlastností k popisu hustoty davů. LBP se často používá k rozpoznávání obličejů a obličejové mimiky. Operace LBP se řídí podle definice lokálních sousedů, což je poloměr a počet blízkých bodů sousedících s cílovým bodem (např. poloměr 1 s osmi sousedními body). Výsledek LBP tvoří binární kód. Tento proces probíhá po celém obrázku.

## 3.2 Umělé Neuronové sítě

Před tím, než si ukážeme konkrétní způsoby relevantní k rozpoznání lidí v davu, které používají umělé neuronové sítě, tak si popíšeme obecné fungování a principy neuronových sítí. Nejběžnější přístup je kombinace několika vlastností, ze kterých vytvoříme bázi těchto vlastností.

Umělá neuronová síť je výpočetní model používaný v oblasti umělé inteligence. Je založený na biologickém chování mozku [Her23]. Neuron je tvořený tělem (soma) a výběžky z něj, které se dělí na dva typy: axony (dlouhé) a dendrity (krátké). V těle neuronu je uloženo jádro. Neuron přijímá informace z ostatních neuronů pomocí dendritů a odesílá informace pomocí axonů. Konec těchto výběžků se nazývá synapse. Efektivnost těchto synapsí může být ovlivněna procházejícími signály tak, že se učí z aktivit, kterých jsou součástí. Tento efekt slouží jako lidská paměť, vlastnosti můžeme vidět napodobené u neuronů umělých neuronových sítí.

Každá neuronová síť je složena z neuronů, které jsou propojeny tak, že výstup jednoho neuronu je vstupem do jednoho či více dalších neuronů. Neurony jsou spojené takzvanými synaptickými vahami, které se upravují při učení. Neurony jsou zorganizované do vrstev, kde je první vrstva vstupní a poslední výstupní, mezi nimi mohou být také další skryté vrstvy. Podle konfigurace se neuronová síť v průběhu času vyvíjí, mění stav svých neuronů a upravuje odpovídající váhy.

### 3.2.1 Učení neuronových sítí

Aby síť vytvářela správnou odezvu na vstupní signál, musí být váhy neuronové třídy správně nastavené. Proces nastavení těchto vah se nazývá učení. V problematice umělých neuronových sítí učení znamená, že se síť přizpůsobuje, aby mohla lépe vyřešit úlohu. Síť se přizpůsobuje tím, že se pokouší eliminovat objevené chyby, čímž se zvýší přesnost sítě. Umělé neuronové sítě se dělí podle způsobu učení (trénování neuronové sítě).

#### 3.2.1.1 Učení s učitelem

Neuronová síť se učí srovnáním aktuálního výstupu s výstupem požadovaným. Tento typ potřebuje znalost požadované odpovědi pro daný vstup [Pri+16]. Neuronovým sítím jsou přiřazeny dva sety dat: trénovací (určený k učení neuronové sítě) a testovací (určený k testování přesnosti neuronové sítě). Trénovací data musí být správně označena, což vytvoří vztah mezi vlastnostmi trénovacího příkladu a daného popisku. Algoritmus učení s učitelem analyzuje trénovací data a odvodí funkci použitelnou k označení nových příkladů. Ideální algoritmus by správně rozhodl o



třídě popisku neznámých příkladů. Nastavuje synapse tak, aby se snížil rozdíl mezi skutečným a požadovaným výstupem. Provádí se tolik průchodů, dokud rozdíl skutečných a požadovaných výstupů spadá do určité odchylky.

V sítích typu učení s učitelem je často používaný algoritmus zpětného šíření chyby. Je založený na metodě gradientního sestupu. Vypočítá gradient účelové funkce s ohledem na váhy sítě. Iteruje zpětně z poslední vrstvy, aby se vyhnul zbytečnému počítání.

### 3.2.1.2 Učení bez učitele

Tento typ neuronových sítí se používá v případě, pokud neznáme požadovaný výstup předem. Modely, které se učí bez učitele, se mohou naučit jednoduché představitele vstupních dat, která mohou být použita ke generování nových a hledání existujících dat. Váhy synapsí se nastavují tak, aby byl výstup sítě konzistentní, tj. aby síť poskytovala stejný výstup při stejném nebo podobném vstupu. Pokud jsou dva spojené neurony aktivovány zároveň, tak zvýší váhu spojení mezi těmito dvěma neurony. Při učení síť používá chyby ve svém napodobeném výstupu k tomu, aby se přiblížila k požadovanému výsledku.

## 3.2.2 Šíření informací neuronové sítě

Umělé neuronové sítě se dají dělit podle způsobů propojení neuronů a druhu aktivačních funkcí neuronů. Neuronové sítě lze dále dělit podle toho, zda se informace v síti šíří acyklicky nebo cyklicky.

### 3.2.2.1 Dopředné neuronové sítě (feedforward neuronové sítě)

Informace v těchto neuronových sítích jdou jen dopředu. Topologie této neuronové sítě je tudíž acyklická, tzn. vazba nikdy nesměruje zpět. Nejjednodušší model dopředné neuronové sítě je třeba Perceptron. Jedná se o jednovrstvou síť s dopředným šířením. Perceptron má několik vstupů ohodnocenými váhami, jeden výstup a aktivační funkci.

### 3.2.2.2 Zpětnovazebné neuronové sítě (recurrent neuronové sítě)

Topologie této sítě je cyklická, tzn. všechny vazby směřují dopředu a některé mohou směřovat zpět. Informace se mohou šířit jak do dalších vrstev, tak do předchozích nebo dokonce z neuronu do sebe samého. Zpětnovazebné sítě používají vnitřní paměť pro zpracování posloupnosti vstupů, díky čemuž jsou použitelné pro úlohy

např. rozpoznávání řeči nebo rukopisu. Zpětnovazebné neuronové sítě ukazují Turingovskou úplnost, což znamená, že je možné spustit (s určitou obtížností) libovolný program s libovolným vstupem.

### 3.2.3 Perceptron

Perceptron je jeden z běžně nejpoužívanějších modelů umělých neuronových sítí. Perceptron se skládá ze vstupních hodnot, skalární a aktivační funkce, která generuje binární hodnoty (0 nebo 1). Považuje se za jednovrstvou dopřednou síť. Používá aktivační funkci, která mapuje vstup  $x$ , který násobí s naučenou hodnotou váhy  $w$  a vygeneruje hodnotu funkce  $f(x)$ :

$$f(x) = 1 \text{ pokud } w * x + b > 0$$
$$f(x) = 0 \text{ jinak}$$

kde  $w * x$  je skalární součin:

$$\sum_{i=1}^m w_i x_i$$

kde  $m$  je počet vstupů do perceptronu a  $b$  je bias, prvek, který upraví hranici od počátku bez jakékoli závislosti na vstupní hodnotě.

Existuje také vícevrstvý perceptron, model neuronové sítě podobný jednovrstvému, který má více skrytých vrstev a informace se v síti šíří i zpětnovazebně [JMM96]. Tyto vícevrstvé perceptrony mohou vyřešit složitější nelineární problémy a funguje dobře s velkými i malými daty. Nevýhodou je, že výpočet bude časově náročnější a složitější.

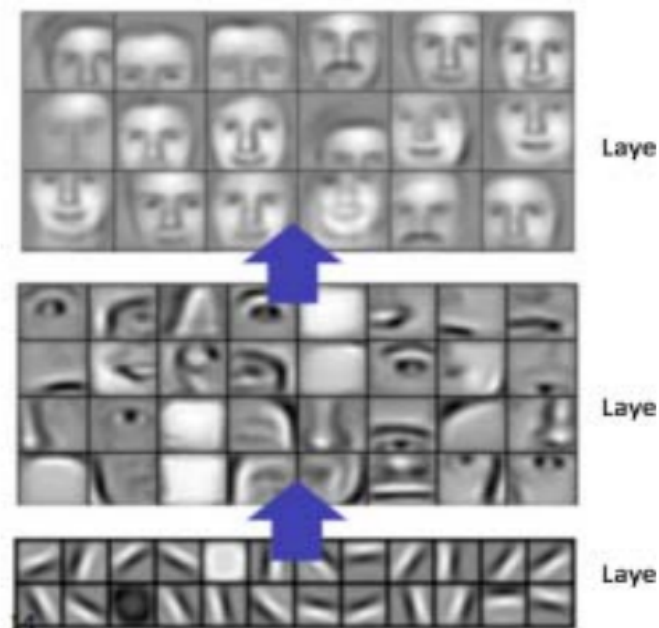
### 3.2.4 Konvoluční neuronové sítě

Celá další sekce čerpá a je shrnutím článku *Understanding of a convolutional neural network* [AMA22].

Konvoluční neuronové sítě (zkratkově CNN - convolutional neural network) mají výborný výkon obzvláště v problémech, které se zabývají zpracováním obrazu. CNN používají matematickou operaci nazývanou se *konvoluce* (lineární operace mezi maticemi). CNN jsou upravené vícevrstvé perceptrony. CNN mají několik vrstev: konvoluční (convolutional) vrstva, nelineární (non-linearity) vrstva, sdružovací (pooling) vrstva a plně připojená (fully connected) vrstva. Konvoluční a plně připojená vrstva mají parametry, zatímco nelineární a sdružovací vrstva nemají parametry.

CNN jsou použitelné v problémech jako například: rozpoznání videa a obrazu, klasifikace a segmentace obrazu, zpracování jazyka nebo analýza obrazu z lékařského obrazu. CNN by neměl mít vlastnosti, které jsou závislé na prostoru. Například by měl rozpoznávat obličej bez ohledu na to, kde a jak jsou na fotografii umístěny viz obrázek 3.2.

CNN jsou vhodné k naší problematice z důvodu, že může přesně modelovat nepřímý vztah mezi fotografiemi davu a počtem osob v něm [Kon+23]. Bohužel problémy, jako různá měřítka nebo rušné pozadí (nejednotné pozadí fotografie, které se nesnadně zpracovává), jsou ale i přes to pro CNN obtížné [Pri+16].

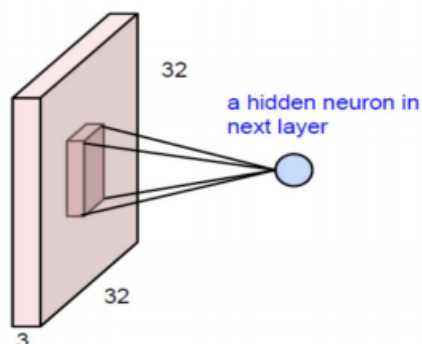


Obrázek 3.2: Naučené vlastnosti konvoluční neuronové sítě [AMA22]

CNN se skládá z několika prvků:

### 3.2.4.1 Konvoluce (Convolution)

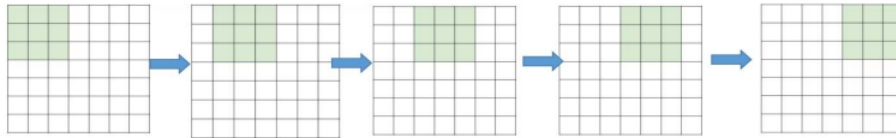
Pokud bychom připojovali každou informaci fotografie, bylo by třeba použít příliš velké množství vážených spojení. Tudíž použijeme mnohem efektivnější metodu, ve které místo celého obrazu spojíme neurony jen s částmi obrazovky, jak můžeme vidět na obrázku 3.3. Parametry této vrstvy jsou naučitelné filtry, které mají malou plochu. Tento způsob drasticky snižuje počet spojení a umožní detekovat vlastnosti bez ohledu na jejich přesnou lokaci ve fotografii.



Obrázek 3.3: Konvoluce jako alternativa pro úplně připojenou síť [AMA22]

### 3.2.4.2 Krok (Stride)

Pomocí kroku můžeme změnit překrytí neuronu dalšími vrstvami se svými sousedy. Při zvýšení kroku dále snížíme počet výstupních spojení. Na obrázku 3.4 můžeme vidět, jak se filtr hýbe o 1 pixel s použitím kroku 1.



Obrázek 3.4: Filtr se hýbe jen o jeden pixel, pro krok 1. [AMA22]

### 3.2.4.3 Vyplnění (Padding)

Jedna z nevýhod konvoluce je, že dochází ke ztrátě informace na okraji obrázku, která je způsobena filtrem. Jednoduchým a efektivním řešením je vyplnění okraje nulami (zero-padding) - pixely s nulovou hodnotou.

Tato kapitola popisuje návrh metody pro detekci osob na fotografii.

Pro detekci osob na fotografii můžeme využít knihovny OpenCV<sup>1</sup>. Využijeme předtrénovaný HOG (histogram orientovaných gradientů) a lineární SVM (metoda podpurných vektorů) model.

Metoda používá posuvné okénko, ve kterém při každém kroku vytáhne HOG vlastnosti, které pak pošle do lineárního SVM pro klasifikaci.

Před použitím samotné metody zmenšíme velikost fotografie. Zmenšením fotografie zajišťuje že musí být porovnáno méně posuvných okének. Dále zmenšení zvyšuje přesnost detekce a snížení detekce falešných pozitiv.

Algoritmus má několik parametrů, které jsou velmi důležité a mají vliv na přesnost a rychlost detekce. Tyto parametry jsou: *winStride* a *padding*. *winStride* je parametr popisující velikost kroku posuvného okénka. Čím menší je tento parametr tím více okének musí být zhodnoceno, což velmi snižuje rychlost detekce.

Další důležitý parametr je *padding*. Tento parametr indikuje počet pixelů které jsou vyplněny u posuvného okénka. Přidání trochu vyplnění před vytažení vlastností HOG zvětšuje přesnost detektoru.

Po využití automatické metody dostaneme souřadnice detekcí lidí, které předáme do aplikace pro manuální úpravy uživateli.

---

<sup>1</sup><https://opencv.org/>

# Implementace

## 5

V této kapitole jsou popsány knihovny použité při implementaci aplikace. Dále je popsána adresářová struktura projektu a popis, v jakém prostředí byl projekt postaven. Následně je vysvětlen překlad tohoto projektu. Dále je popsán formát souboru, který aplikace exportuje, a nakonec si popíšeme konkrétní implementace jednotlivých tříd.

Tato aplikace byla vytvořena v jazyce Java verze 17 s využitím knihovny JavaFX<sup>1</sup>, což je open source knihovna, ve které se staví aplikace s uživatelským rozhraním. Uživatelská rozhraní jsou definována pomocí FXML, jde o značkový jazyk založený na XML.

Pomocí knihovny OpenCV<sup>3</sup> byla vytvořena automatická metoda pro detekci a počítání lidí na fotografii.

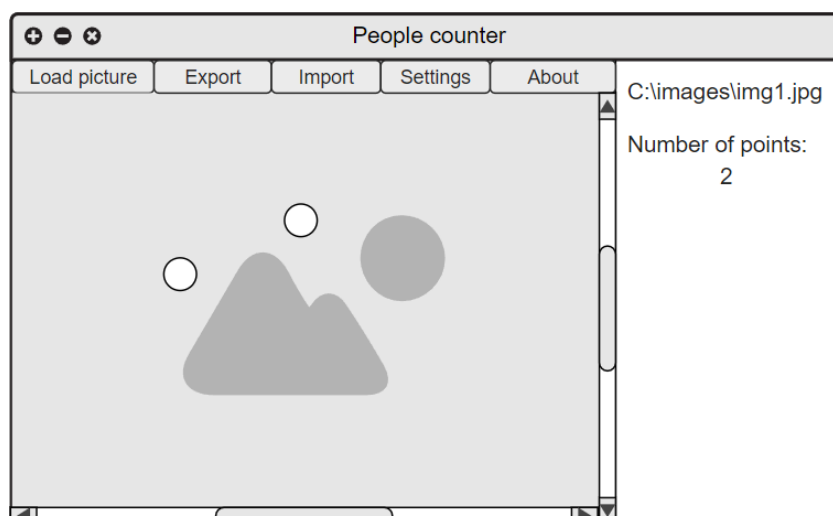
Na obrázku 5.1 je zobrazen návrh hlavního okna naší aplikace. V horní části obrázku můžeme vidět tlačítka pro samotné ovládání aplikace. Nalevo je fotografie, na kterou budeme moci pokládat body a posouvat fotografii táhnutím myši. V pravé části jsou uvedeny informace k načtené fotografii a počet bodů, které jsme položili. Návrh grafického rozhraní byl vytvořen v aplikaci moqups<sup>2</sup>.

---

<sup>1</sup><https://openjfx.io/>

<sup>2</sup><https://moqups.com/>

<sup>3</sup><https://opencv.org/>



Obrázek 5.1: Návrh hlavního okna

## 5.1 Model aplikace

Aplikace je navržena podle MVC (model-view-controller) modelu. MVC je designový vzor, který rozdělí logiku aplikace do tří spolu propojených částí. *Model* je centrální část aplikace, spravuje data, logiku a ovládá aplikaci. Model může reprezentovat buď data přenášená mezi *View* a *Controller* nebo nějaká jiná data související s logikou. *View* představuje reprezentaci a logiku všech grafických prvků, které uživatel vidí nebo s nimi určitým způsobem interaguje (např. tlačítko, rozevírací seznam, zaškrtnuté políčko). *Controller* pracuje jako interface mezi *Model* a *View*. Zpracovává logiku a příchozí požadavky, upravuje data pomocí *Model* a komunikuje s *View*, aby vykreslil konečný výsledek.

## 5.2 Struktura projektu

Projekt byl sestaven v prostředí Maven<sup>1</sup>. Na obrázku 5.2 můžeme vidět adresářovou strukturu projektu aplikace. Ve složce *javadoc* můžeme otevřením souboru *index.html* otevřít javadoc dokumentaci projektu. Ve složce *src/main* najdeme dvě složky *java* a *resources*. Ve složce *java* jsou controller třídy a vedlejší pomocné třídy. Složka *resources* obsahuje *fxml* soubory, které definují *view* z MVC modelu. Ve složce *tools* je umístěn nástroj *Warp*<sup>2</sup>. Tento nástroj slouží k vytvoření soběstačné binární aplikace a je využitý při překlada programu. *pom.xml* (Project object model) je soubor, který obsahuje informace o projektu a o jeho konfiguraci pro překlad programu.

<sup>1</sup><https://maven.apache.org/>

<sup>2</sup><https://github.com/dgiagio/warp>





Obrázek 5.2: Adresářová struktura projektu

## 5.3 Překlad

Pro překlad musí být nainstalovaná Java JDK (java development kit) verze 17 (nemusí obsahovat *JavaFX* nebo *OpenCV*, o ten se postará automatický překlad) a musí být nainstalovaný *Maven*. Překlad je proveden automaticky konfiguračním souborem *pom.xml*. V *CMD* (command-line) zadejte v hlavní složce projektu *hellofx* příkaz:

```
mvn package
```

*Maven* pak automaticky do složky *hellofx/target* vytvoří *JLink* prostředí s odpovídajícím souborem *hellofx-1.0-SNAPSHOT.jar* a poté *Warp* vytvoří soběstačnou binární aplikaci *PeopleCounter.exe*. Pro spuštění této aplikace nejsou potřeba žádné závislé knihovny.

## 5.4 Formát exportovaného souboru

Exportovaný soubor se uloží na vybranou cestu uživatele ve formátu s koncovkou *txt* v čistém textu. Soubor se ukládá v kódování *UTF-8*. Formát tohoto souboru můžeme vidět na obrázku 5.3, kde každá řádka má pár záznamů ve formátu *proměnná=hodnota*. Soubor je rozdělen na tři části: *general*, *points* a *settings*.

### 5.4.1 general

Tato část obsahuje informaci o absolutní cestě k fotografii *picture\_path*, dále celkový počet bodů označených uživatelem *number\_of\_points* a nakonec informaci *finished*. Tato proměnná má hodnotu *true*, pokud bylo dokončeno počítání osob, nebo *false*, pokud chce uživatel ještě pokračovat. Tyto informace slouží k importu do programu, tak jako jednoduchý přehled uživateli bez nutnosti importu do aplikace.

### 5.4.2 points

V této sekci jsou vypsané souřadnice všech položených bodů, které jsou označeny jako *p* s hodnotami nejprve *X* souřadnice bodu, poté *Y* souřadnice bodu. Těchto záznamů je *N* podle počtu položených bodů. Část s body není povinná a pokud není označené zaškrtačkové pole v *ExportMenu*, tak se do exportovaného souboru nezapiše.

### 5.4.3 settings

V poslední části jsou uvedena všechna nastavení, která jsou k dispozici v *pointSettings* menu. Proměnná *color* je barva bodů v hexadecimální podobě, *shape* je tvar bodů ze třídy *Shape* (např. kruh, čtverec, plus a trojúhelník), *size* je velikost přidávaných bodů (od 1 do 50) a *opacity* je neprůhlednost bodů (od 0 do 100 v procentech, při 0 % jsou body skryté a při 100 % jsou body plně neprůhledné).

```
--general--  
picture_path='CESTA FOTOGRAFIE'  
number_of_points='POČET POLOŽENÝCH BODŮ'  
finished='TRUE nebo FALSE'  
--points--  
p='X SOUŘADNICE BODU 1' 'Y SOUŘADNICE BODU 1'  
p='X SOUŘADNICE BODU 2' 'Y SOUŘADNICE BODU 2'  
...  
...  
p='X SOUŘADNICE BODU N' 'Y SOUŘADNICE BODU N'  
--settings--  
color='BARVA BODŮ V HEXADECIMÁLNÍ PODOBĚ'  
shape='TVAR BODŮ'  
size='VELIKOST BODŮ'  
opacity='PRŮHLEDNOST BODŮ'
```

Obrázek 5.3: Formát exportovaného souboru

## 5.5 Popis tříd

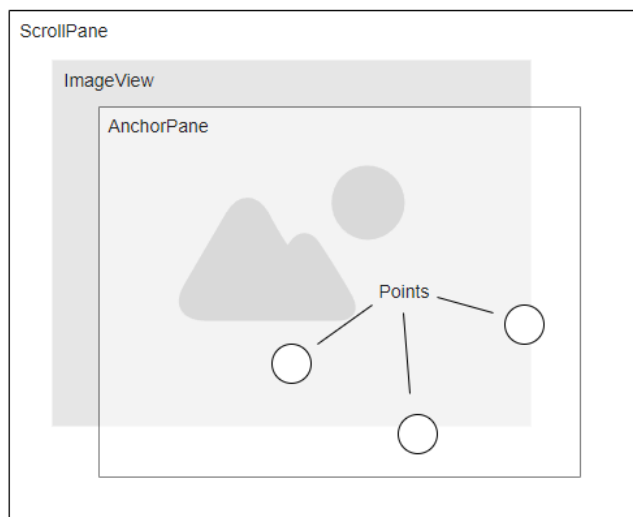
### 5.5.1 Main

Třída Main načte scénu hlavního okna ovládanou třídou *Controller*. Je to vstupní bod do programu.

### 5.5.2 Controller

*Controller* je třída hlavního okna, která patří k view *window.fxml*. Tato třída se stará jak o funkčnost tlačítek v menu, tak o funkčnost ovládání samotné manipulace s fotografií a pokládání bodů. Třída zajišťuje obsluhu událostí - tzn. kliknutí myši na fotografii pro položení bodu, posunutí kolečka pro přiblížení a oddálení fotografie nebo táhnutí myši na posunutí fotografie. Dále obstarává načtení fotografie vybrané uživatelem. Při načtení nové fotografie se z plochy odstraní všechny položené body a obnoví se výchozí stav. Tato fotografie může být načtena přímo otevřením fotografie nebo importována přes vybraný exportovaný soubor. V případě načtení z exportovaného souboru, jsou položeny související body na fotografii a načteno nastavení. Navíc třída přidává body na fotografii a kontroluje, aby některé nepřesahovaly její okraj. Body vykresluje na fotografii podle nastavení stylů (barva, tvar, velikost a neprůhlednost). Tyto body jsou grafické, logicky jsou definované třídou *Point*. Dále třída spouští automatickou detekci ve třídě *FaceRecognition*, posadí a spočítá detekované osoby.

Body se na fotografii přidávají způsobem, který je popsán na obrázku 5.4. Podokno *ScrollPane* má jako potomky podokna *ImageView* a *AnchorPane*. Do *ImageView* je při načtení přidána fotografie, zatímco do *AnchorPane* jsou přidávány body při kliknutí na *ImageView*.



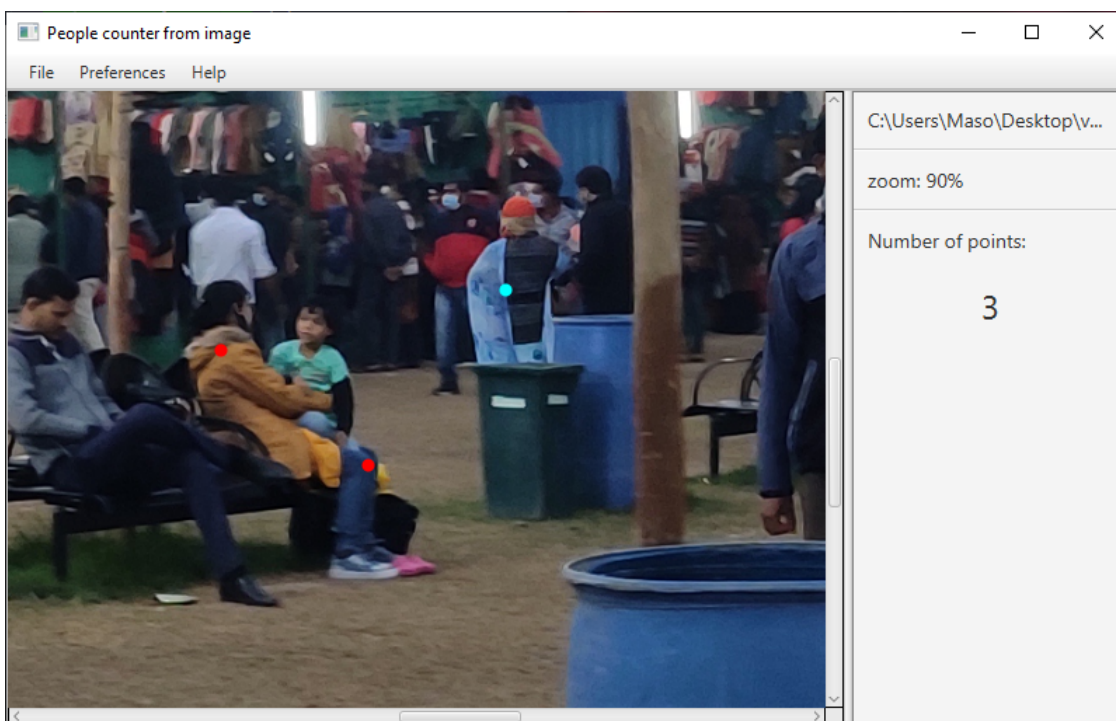
Obrázek 5.4: Rozložení panelů levé části hlavního okna

Nakonec třída *Controller* přidává funkčnost odebrání bodů (kliknutím levým tlačítkem na cílový bod) a označení posledního položeného bodu inverzní barvou pro odlišení od ostatních bodů.

### 5.5.3 window.fxml

*window.fxml* reprezentuje view hlavního okna s controller třídou *Controller*. Hlavní okno je tvořené *BorderPane* panelem, což je panel, který je tvořený potomky: *Center* (vyplňuje prostředek obrazovky, velikost se odvozuje od ostatních potomků) *Left* (zarovnaný doleva), *Top* (zarovnaný nahoru), *Right* (zarovnaný doprava) a *Bottom* (zarovnaný dolů). V *top* části panelu je umístěn *MenuBar*. V této liště jsou tlačítka pro načtení nové fotografie *export* a *import* dat, nastavení stylu bodů, ovládání, *about* sekci a ukončovací tlačítko, které uzavře aplikaci.

V *Center* je umístěno *SplitPane* rozdělující okno na levou část s fotografií a pravou se souvisejícími statistikami. Levá část je tvořena pomocí *ScrollPane* pro posouvání přiblížené fotografie a *AnchorPane*, do kterého jsou při kliknutí přidány body. Pravá část je tvořena *Vbox* s popisky cesty k načtené fotografii, popisem aktuálního přiblížení fotografie v procentech, počet položených bodů a zda je počítání osob dokončeno. Na obrázku 5.5 můžeme vidět snímek hlavního okna s otevřenou vzorovou fotografií se třemi položenými body.



Obrázek 5.5: Snímek hlavního okna

## 5.5.4 **DataHandler**

Třída *DataHandler* zajišťuje poskytování a ukládání důležitých dat a nastavení, které dále poskytuje k dispozici jednotlivým controllerům. V této třídě jsou uloženy následující informace: souřadnice označovacích bodů, nastavení stylů bodů, cesta k fotografii a zda bylo počítání lidí dokončeno. Tato třída je vytvořena podle vzoru *Singleton*. Singleton třída je třída, které může existovat jen jedna instance a poskytuje jednoduchý globální přístup k této instanci.

## 5.5.5 **Point**

Třída *Point* reprezentuje jeden položený bod. Parametry tohoto bodu jsou X a Y souřadnice na načtené fotografii. Tato třída reprezentuje jen logický bod o vykreslení tohoto bodu se stará třída *Controller*.

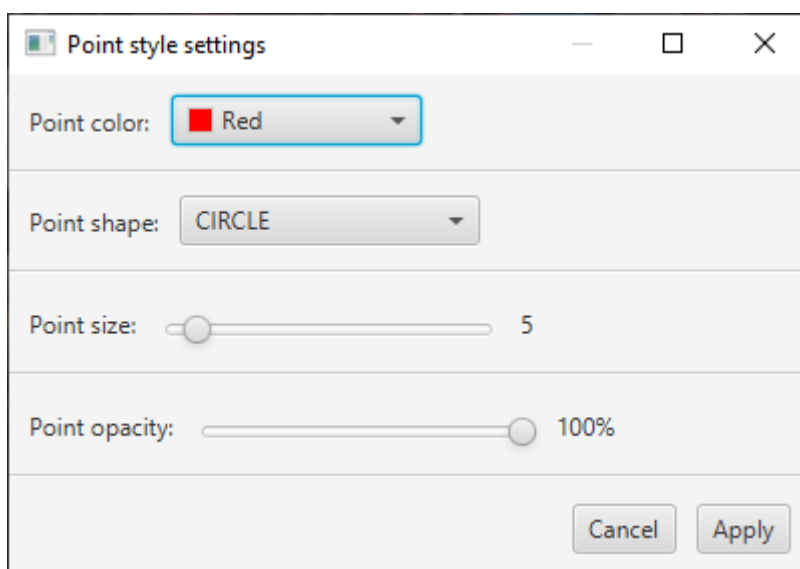


## 5.5.6 **PointSettingsController**

*PointSettingsController* je třída okna s nastavením stylu bodů patřící k view *pointSettings.fxml*. Třída se stará o navázání hodnot posuvníků a jejich popisků. Dále zajišťuje funkčnost tlačítek *Apply* (změny uloží a zavře okno) a *Cancel* (zavře okno bez uložení změn v nastavení).

## 5.5.7 **pointSettings.fxml**

*pointSettings.fxml* reprezentuje view okna s nastavením stylu bodů s controller třídou *PointSettingsController*. Toto okno je tvořené pomocí *VBox*, ve kterém jsou popisky s různými prvky nastavení. *Vbox* je panel, který své prvky organizuje vertikálně pod sebou. Tyto prvky nastavují velikost, tvar, barvu a průhlednost bodů. Na obrázku 5.6 můžeme vidět snímek nastavení položených bodů.



Obrázek 5.6: Snímek okna s nastavením bodů

## 5.5.8 **Shape**

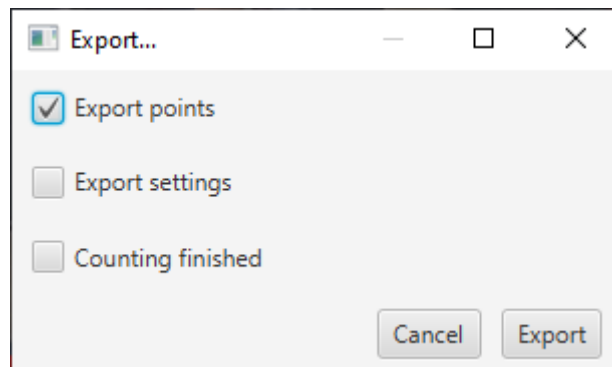
*Shape* je *enum* třída pro možné tvary bodů. Mezi tyto tvary patří: kruh (*CIRCLE*), čtverec (*SQUARE*), plus (*PLUS*) a trojúhelník (*TRIANGLE*).

## 5.5.9 ExportMenuController

*ExportMenuController* je třída okna s exportem dat, patří k view *pointSettings.fxml*. Třída zajišťuje správný export dat do vytvořeného souboru tlačítkem *Export* pomocí třídy *ExportedDataWriter*. Dále přidává funkčnost zrušení akce exportu tlačítku *Cancel* (zruší akci exportování).

### 5.5.10 exportMenu.fxml

*exportMenu.fxml* reprezentuje view okna s exportem stavových dat s controller třídou *ExportMenuController*. V tomto okně je panel *VBox*, ve kterém najdeme zaškrťovací pole pro export bodů, stylu bodů, nastavení a zda bylo označování bodů dokončeno. Na obrázku 5.7 můžeme vidět okno s výše popsaným exportem.



Obrázek 5.7: Snímek export okna

### 5.5.11 ExportedDataWriter

Třída *ExportedDataWriter* provádí napsání dat do souboru při jejich exportu. Tento formát je popsán v sekci 5.4.

### 5.5.12 ExportedReader

Tato třída se stará o načtení souboru s exportovanými daty dle daného formátu a o jejich následné uložení.

### 5.5.13 FaceRecognition

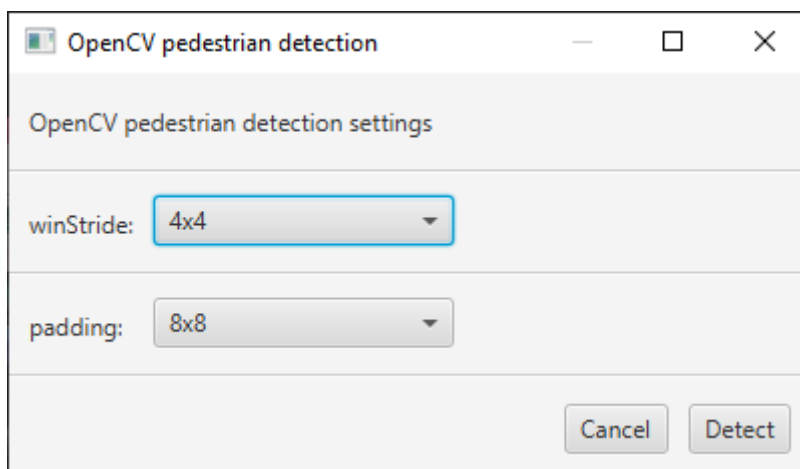
Tato třída popisuje zpracování a detekci osob ve fotografii. Třída se stará o načtení fotografie do *OpenCV*, zmenšení této fotografie pro zrychlení detekce. Dále obsahuje nastavení parametrů potřebných k samotné detekci. Po detekci se třída stará o zpětné zvětšení poměru získaných souřadnic.

### 5.5.14 DetectionSettingsController

*DetectionSettingsController* je třída okna s nastavením stylu bodů patřící k view *detectionSettings.fxml*. Třída se stará o předání vybraných hodnot metodě pro detekci osob. Dále zajišťuje funkčnost tlačítek *Detect* (provede detekci osob) a *Cancel* (zavře okno bez provedení detekce).

### 5.5.15 detectionSettings.fxml

*detectionSettings.fxml* reprezentuje view okna s nastavením metody pro detekci osob s controller třídou *PointSettingsController*. Toto okno je tvořené pomocí *VBox*, ve kterém jsou popisky s různými prvky nastavení metody detekce. U obou popisek jsou uvedeny možnosti s hodnotami parametrů. Na obrázku 5.8 můžeme vidět snímek nastavení metody pro detekci osob.



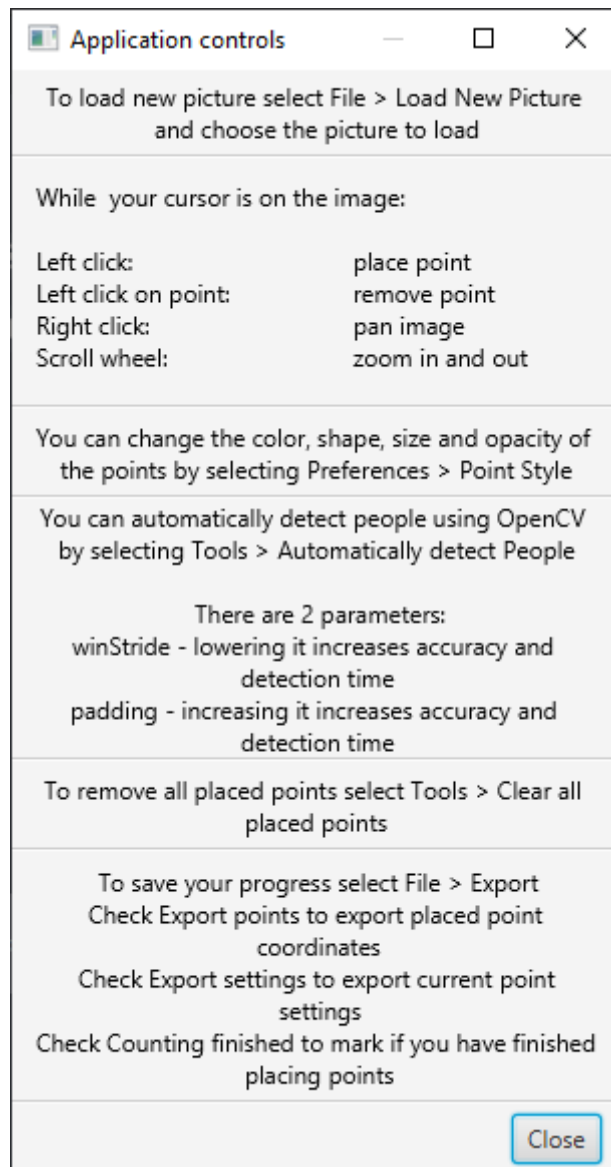
Obrázek 5.8: Snímek okna s nastavením detekce osob

### 5.5.16 ControllsController

*ControllsController* je třída okna s vysvětlivkami k ovládání aplikace, patří k view *pointSettings.fxml*. Třída přidává funkčnost uzavření okna tlačítkem *Close*.

## 5.5.17 **controlls.fxml**

*controlls.fxml* reprezentuje view okna s vysvětlivkami ovládacích prvků s controller třídou *ControllsController*. Toho okno je sestaveno pomocí *Vbox*, ve kterém jsou oddělené sekce popisující detailní ovládání vytvořené aplikace. Na obrázku 5.9 můžeme vidět okno s vysvětlivkami jednotlivých ovládacích prvků.



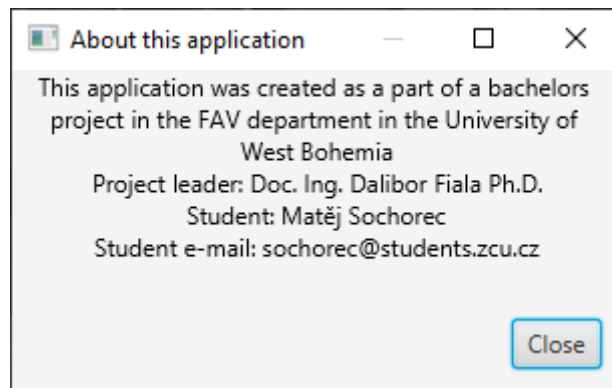
Obrázek 5.9: Snímek okna s vysvětlivkami ovládání

## 5.5.18 **AboutController**

*AboutController* je třída okna s informacemi o aplikaci, patří k view *pointSettings.fxml*. Tato třída také přidává funkčnost zavření okna tlačítkem *Close*.

## 5.5.19 **about.fxml**

Třída *about.fxml* reprezentuje view okna s informacemi o aplikaci s controller třídou *AboutController*. V tomto okně je panel *VBox*, ve kterém jsou informace o aplikaci (autor, vedoucí projektu, kontakt atd.). Na obrázku 5.10 můžeme vidět okno s informacemi o aplikaci.



Obrázek 5.10: Snímek okna s informacemi o aplikaci

V této kapitole je popsáno, jak probíhal postup testování, a jak vypadala testovací sada s popisem samotných souborů v testovací sadě. Dále jsou výsledky testování sestaveny do tabulky a grafů, u kterých jsou prozkoumána možná vysvětlení výsledků.

## 6.1 Struktura testování

Testovací sada byla zaslána deseti testerům. Tato sada má adresářovou strukturu, kterou můžeme vidět na obrázku 6.1.

```
testovaci_sada
├── test_foto
│   └── 8 testovacích fotografií
├── postup_testovani.txt 6.2
└── PoepCounter.exe
```

Obrázek 6.1: Adresářová struktura testové sady

Ve složce `test_foto` jsou fotografie, které testeři označili. Tyto fotografie jsou součástí datasetu popsaného v sekci 6.3. V souboru `postup_testovani.txt` na obrázku 6.2 je popsán postup, který měli testeři následovat. `PeepCounter.exe` je samotná aplikace.

Po splnění instrukcí v souboru `postup_testovani.txt` odeslali zpět testeři exportované soubory obsahující všechny důležité informace.

Tento dokument slouží k popisu postupu testování aplikace pro podporu zjišťování počtu osob na fotografii.

postup testování:

1. Spusťte program PeopleCounter.exe.
2. Načtěte nový obrázek kliknutím na: File > Load new picture. Vyberte jednu fotografii ze složky test\_foto přibalenou v testovacím archivu.
3. Po načtení obrázku označte všechny osoby na fotografii umístěním bodu na hlavu.

Při ukazování kurzoru na fotografii:

- body se pokládají levým kliknutím myši
- levým kliknutím na bod se tento bod vymaže
- posunutí kolečkem nahoru/dolů se fotografie přiblíží/oddálí
- pravým tlačítkem se posune fotografie

Tvar/barva/velikost/průhlednost bodů se dá upravit podle potřeby kliknutím na: Preferences > Point style.

4. Po dokončení označování exportujte záznam bodů kliknutím na: File > Export. Zaškrtněte políčko Export Points a Counting finished a klikněte na tlačítko Export.
5. Opakujte 2-4 pro všechny obrázky ve složce test\_foto

Pokud narazíte na nějakou chybu, nebo na nějaký nedostatek/připomínku k fungování aplikace, napište jí zvlášť do textového souboru.

Výsledné exportované soubory a možné připomínky pošlete zpět preferovaným komunikačním kanálem.

Obrázek 6.2: Obsah souboru postup\_testovani.txt

## 6.2 Demografika testerů

Abychom mohli výzkum kvalitně zrealizovat musíme si správně definovat cílovou skupinu respondentů. Existuje několik způsobů výběru: *prostý náhodný výběr* (testeri jsou vybráni náhodně), *systematický náhodný výběr* (sestaven z výběru každého n-tého případu, např. každého desátého), *stratifikovaný náhodný výběr* (vytvoření úzce zaměřené skupiny - straty) a *vícestupňový shlukový výběr* (není úzce zaměřený, nevytvářejí se straty, ale jen přirozená seskupení lidí - klastry). Pro náš výzkum byl nejvhodnější způsob výběru skupiny testerů *vícestupňový shlukový výběr*. Tento způsob byl zvolen z důvodu, aby obsáhl co nejširší věkové rozpětí a technickou zdatnost testerů.

V rámci testování bylo osloveno 10 testerů, z nichž zhruba polovina byla ve věkové kategorii 20-25 let a druhá polovina 45-55 let. Technická schopnost těchto testerů je podobně pestrá, pohybuje se od schopnosti průměrného IT studenta po schopnost běžného uživatele. Takto různorodá demografie testerů byla vybrána, aby se ověřilo, že je aplikace intuitivní pro uživatele technicky zdatného i pro nezkušeného uživatele.

## 6.3 Fotografie testovací sady

Pro testovací sadu bylo vybráno osm barevných fotografií z datasetu *Crowd in a rally* [Dat17], uvedený na webové stránce *Kaggle*. Je to obtížný dataset s 3000+ fotografiemi, ve kterém byly fotografie pořízeny pomocí více než čtyřmi sty dobrovolníky v letech 2021-2022. Fotografie jsou v HD (high definition - vysoké rozlišení) kvalitě a byly opatřeny mobilními telefony v různých světelných kondicích. Fotografie byly pořízeny v různých městech Indie. Tyto fotografie můžeme vidět na obrázku 6.3. Fotografie byly vybrány podle několika vlastností. Jednou z hlavních vlastností je počet lidí, kterých je v řádu desítek až stovek, což je ideální počet pro naši metodu. Další významnou vlastností je kvalita fotografie, což umožňuje přesné označení osob.



Při zpracování fotografií je třeba vždy mít na mysli, že existují různé režimy nakládání se záznamy a jejich následné posuzování. Musíme počítat s tím, že k pořizování, dalšímu používání, šíření a zveřejňování fotografií a videí fyzických osob může docházet v režimu zpracování osobních údajů. Pokud je navržená aplikace využita výhradně pro soukromé účely (na soukromé párty, na svatbě apod.) a není později nikde sdílána, není třeba se problematikou zpracování a ochrany osobních údajů zabývat. Je-li aplikace využita s cílem dalšího šíření a zveřejňování, je třeba se řídit platnou legislativou. Z výše uvedených důvodů byly vybrány fotografie z volně dostupného datasetu, aby nedošlo k porušení nařízení.

Některé z vlastností fotografií testovací sady jsou zajímavé, protože by se mohly ukázat ve výsledcích testování. Na foto7 můžeme vidět největší počet osob (ve stovkách), kvůli kterému můžeme čekat větší variaci mezi výsledky testerů, než na ostatních fotografiích. Další fotografie s větší variací může být foto4, což je fotografie s velkým počtem lidí, kteří se překrývají, tím pádem může vést k nejasnosti při označování. Na druhou stranu na fotografiích foto2 a foto3 jsou počty osob v desítkách, jsou od sebe lépe rozpoznatelné, takže můžeme předpokládat, že bude variace výsledků menší.



(a) foto1



(b) foto2



(c) foto3



(d) foto4



(e) foto5



(f) foto6



(g) foto7



(h) foto8

Obrázek 6.3: Vybrané fotografie pro testování

## 6.4 Výsledek testování

Počet označených lidí z exportovaných souborů testerů byl zapsán do tabulky 6.1. Řádky *tester a-j* označují počet označených osob pro jednotlivé fotografie. V řádku *aritmetický průměr* můžeme vidět průměrný počet lidí označených na jednotlivých fotografiích. *Směrodatná odchylka* určuje míru variability. Pro náš případ malá hodnota směrodatné odchylky znamená, že všechny počty lidí na fotografiích si jsou podobné. Vyšší hodnota znamená, že byly zřejmé velké rozdíly v počtech označených lidí mezi jednotlivými testery. Směrodatná odchylka je určena vztahem:

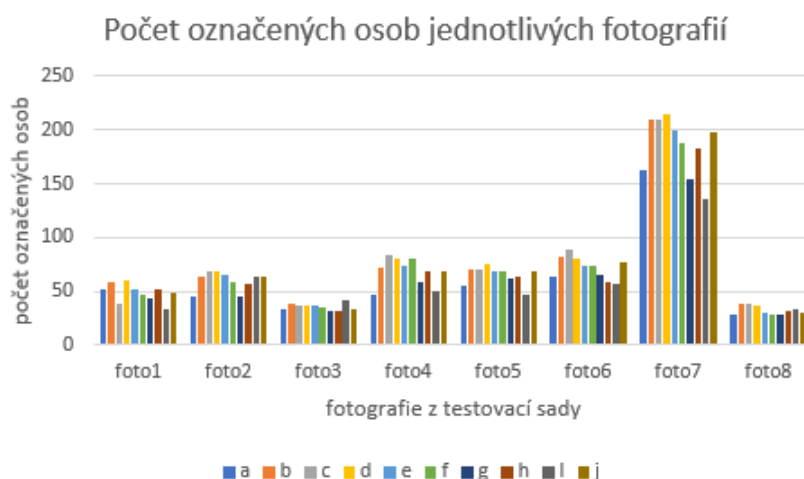
$$\sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$$

kde  $x$  jsou naměřené hodnoty,  $\bar{x}$  je jejich aritmetický průměr a  $n$  je počet měření. *Variační koeficient* určuje standardizovaný poměr odchylky a průměru, který je důležitý z důvodu různých počtů lidí na fotografiích. Variační koeficient je určený vztahem:  $\frac{S_x}{\bar{x}}$  kde  $S_x$  je směrodatná odchylka a  $\bar{x}$  je aritmetický průměr.

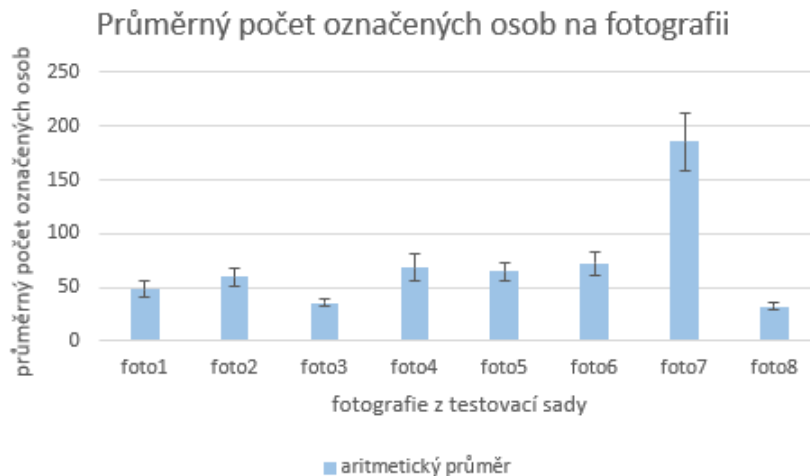
	foto1	foto2	foto3	foto4	foto5	foto6	foto7	foto8
tester a	51	45	34	47	55	63	162	29
tester b	58	63	38	72	70	82	210	38
tester c	39	68	36	83	71	88	210	39
tester d	60	69	36	80	76	80	214	36
tester e	52	66	36	74	69	74	200	30
tester f	46	58	35	81	69	73	188	29
tester g	43	45	31	59	61	66	154	29
tester h	51	57	31	68	63	59	183	31
tester i	34	64	42	50	47	57	135	33
tester j	49	63	34	69	69	77	198	30
aritmetický průměr	48,3	59,8	35,3	68,3	64,9	71,9	185,4	32,4
směrodatná odchylka	8,06	8,68	3,23	12,63	8,61	10,33	26,85	3,89
variační koeficient	0,167	0,145	0,092	0,185	0,133	0,144	0,145	0,12

Tabulka 6.1: počet označených osob na jednotlivých fotografiích testery

Graf 6.4 znázorňuje jednotlivé počty označených osob pro každou fotografii zúčastněných testerů. Dále v grafu 6.5 vidíme průměrné počty osob pro každou fotografii, kde chybové úsečky označují směrodatnou odchylku. Oba grafy byly vytvořeny v aplikaci *Microsoft Excel*<sup>1</sup>.



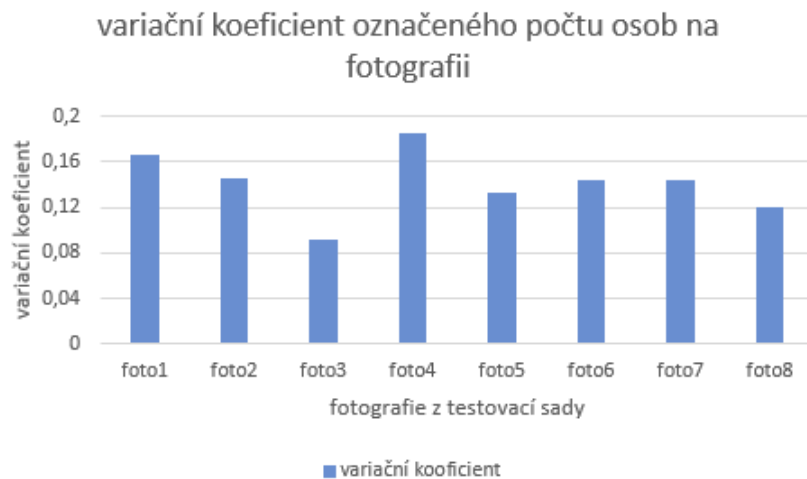
Obrázek 6.4: Počet označených osob na jednotlivých fotografiích



Obrázek 6.5: Průměrný počet označených osob na fotografii

Graf 6.6 udává variační koeficient označeného počtu osob na fotografiích všech testerů.

<sup>1</sup><https://www.microsoft.com/en-us/microsoft-365/excel>



Obrázek 6.6: Variační koeficient označeného počtu osob na fotografii

Z dat můžeme vyčíst, že u většiny fotografií byl variační koeficient kolem 0,14 s nejnižší variací 0,092 u foto3 a nejvyšší variací 0,185 u foto4. Rozdíly počtů označených osob na fotografiích různých testerů mohou být z několika důvodů.

Asi nejzřejmějším důvodem je počet osob na fotografii. Vyšší počet osob si vyžaduje větší trpělivost a důkladnost uživatele. Přestože je na foto7 nejvíce osob, je variace 0,145, což odpovídá variaci ostatních fotografií s podstatně méně vyfočenými osobami. Uvedený případ se dá vysvětlit tím, že osoby na foto7 jsou od sebe vzdáleny, a tudíž lehce od sebe rozpoznány a označeny uživatelem. To nás vede k dalšímu důvodu vyšší variace: hustota davu. Čím je vyšší hustota davu, tím je těžší rozpoznat od sebe osoby na fotografii. Tento problém lze vidět na foto4, fotografii s největší variací. Posledním důvodem je kvalita a ostrost fotografie. Foto3 je fotografie s nejnižší variací, což si můžeme odvodit dobrou kvalitou této fotografie.

Výsledkem této práce je počítačová aplikace pro systém Windows, sloužící k podpoře uživatele při počítání osob na fotografii. V aplikaci lze načíst fotografii, ve které může uživatel kliknutím myši přidávat a odebírat body pro označení osob. Fotografie lze manipulovat táhnutím myši a přiblížením kolečkem myši. Styl označovacích bodů (barva, tvar, velikost) lze v nastavení upravit. Po dokončení práce může uživatel exportovat položené body a nastavení stylu bodů. Naopak pro obnovení práce a nastavení jdou tato data importovat.

V rámci této bakalářské práce jsem provedl průzkum trhu aplikací souvisejících s počítáním osob na fotografii. Dále jsem prošel relevantní techniky a metody zpracování obrazu. Úspěšně jsem implementoval aplikaci, která s pomocí uživatele spočítá osoby na fotografii. Následně jsem navrhl a implementoval automatickou metodu pro počítání osob na fotografii použitím knihovny *OpenCV*. V závěrečné fázi jsem aplikaci otestoval ve spolupráci deseti testerů a zhodnotil naměřené výsledky.

Výsledky testování indikují, že pokud jsou fotografie dostatečně kvalitní a uživatelé dostatečně pečliví a trpěliví, jsou tedy počty osob konzistentní, a tudíž je aplikace spolehlivá. Podle kvality fotografie může automatická metoda urychlit a usnadnit označení osob před manuálním položením bodů. Takto lze pohodlně spočítat počet osob na fotografii s desítkami až stovkami osob. Tisíce osob je pro tuto metodou už nepraktické a zpracování jen jedné fotografie by bylo velmi časově náročné. Pokud je kvalita fotografie velmi nízká nebo je velmi vysoká hustota davu, jsou výsledky zkreslené. Aplikace je použitelná pro zjištění přibližného počtu osob např. při koncertech, sportovních utkáních či policejních zásazích na demonstracích nebo větších davových setkáních. Aplikace nemusí být používána pouze institucemi, ale také pro soukromé potřeby, např. při kontrole počtu pozvaných účastníků svatby nebo autorského čtení.



# Přílohy

## Uživatelská příručka

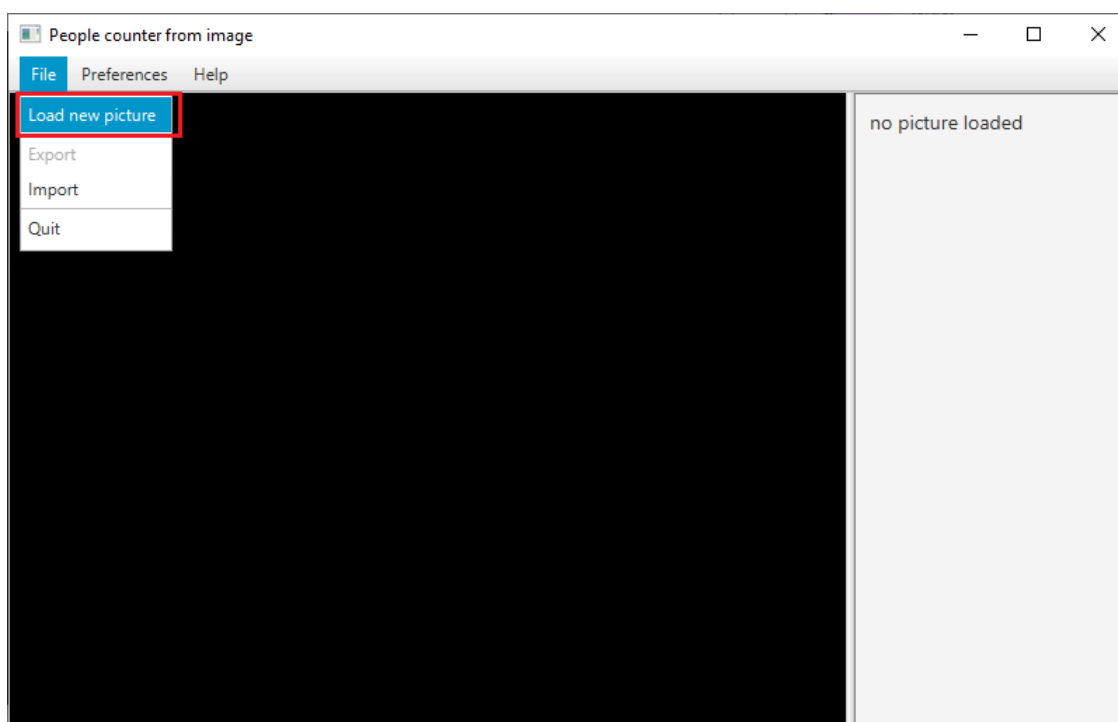
Aplikace je určena pro systém Windows. Před spuštěním aplikace není potřeba instalace žádných knihoven. Spuštěním souboru *PeopleCounter.exe* se otevře aplikace.

Uživatelské rozhraní se dělí na tři hlavní části: lišta nástrojů (horní část), načtená fotografie (levá část) a panel s informacemi (pravá část). V liště nástrojů jsou tlačítka pro načtení nové fotografie, import a export, vypnutí aplikace, nastavení stylů bodů, informace o aplikaci a ovládání aplikace. V okně s načtenou fotografií můžeme vidět položené body s posledním položeným bodem označeným rozdílnou barvou. V panelu s informacemi můžeme vidět cestu k načtené fotografii, přiblížení v procentech a počet položených bodů. Popsané části uživatelského rozhraní můžeme vidět na obrázku 7.1.



Obrázek 7.1: Části uživatelského rozhraní

Pro načtení nové fotografie klikněte na **File > Load New Picture** a vyberte si obrázek k načtení. Načtení obrázku můžeme vidět na obrázku 7.2.



Obrázek 7.2: Načtení nového obrázku

Když máte kurzor na načtené fotografii:

**levé kliknutí** - položte bod na souřadnice myši

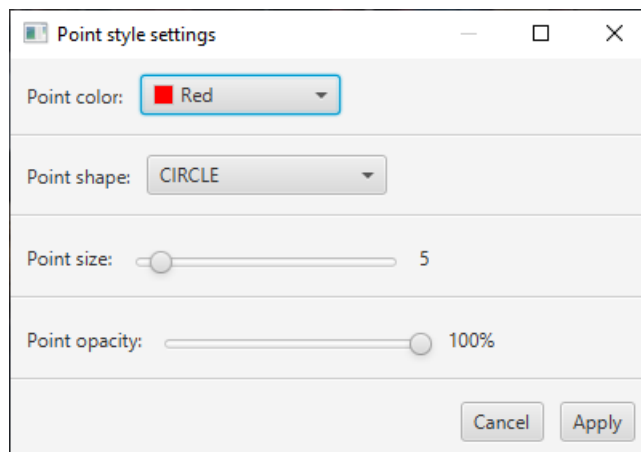
**levé kliknutí na položený bod** - odstraňte bod, na který kliknete

**držení pravého tlačítka** - posuňte fotografii

**posunutí kolečka** - přiblížte a oddalte fotografii

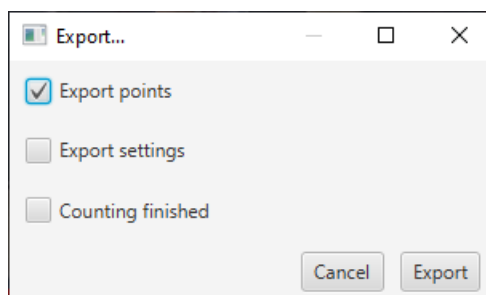


Kliknutím na **Preferences > Point Style** můžete změnit barvu, tvar, velikost a neprůhlednost. Na obrázku 7.3 můžeme vidět červenou barvu bodu, tvar bodu kruh, velikost 5 a 100% neprůhlednost.



Obrázek 7.3: Nastavení stylu bodů

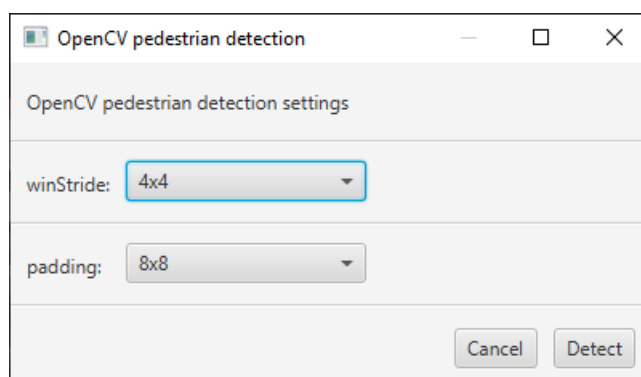
Kliknutím na **File > Export** se otevře menu pro exportování. V tomto menu si můžete vybrat, jestli chcete exportovat položené body, nastavení stylů bodů a jestli bylo počítání dokončeno. Po kliknutí na tlačítko *Export* se otevře okno, ve kterém si vyberete složku a exportovaný soubor uložíte.



Obrázek 7.4: Export bodů a nastavení

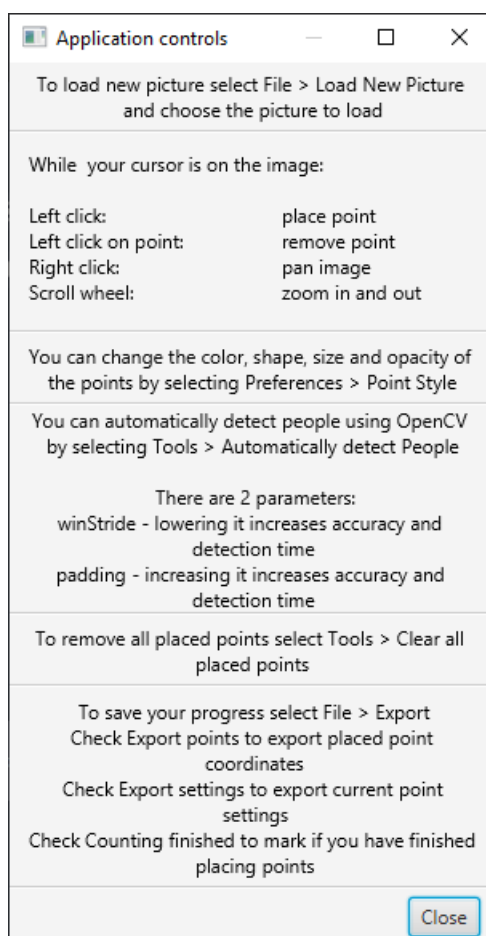
Kliknutím na **File > Import** se otevře okno ve kterém vyberte soubor, který byl exportovaný z této aplikace.

Kliknutím na **Tools > Detect People** se otevře okno s nastavením pro automatickou metodu detekce osob na fotografii. V okně na obrázku 7.5 můžeme vidět dva parametry. Snížení parametru *winStride* dojde ke zvýšení přesnosti detekce, ale také zvýšení času detekce, zatímco zvýšení parametru *padding* způsobí zvýšení přesnosti a zvýšení času detekce. Po kliknutí na tlačítko *Detect* se spustí metoda pro detekci osob na fotografii. Po úspěšné detekci se zobrazí okno s výsledky a souřadnice detekovaných osob se položí na fotografii.



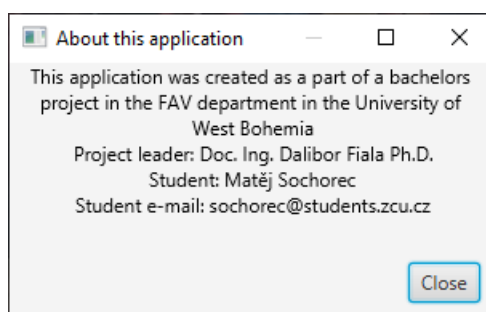
Obrázek 7.5: Nastavení metody pro automatickou detekci osob na fotografii

Kliknutím na **Help > Controls** se otevře okno s instrukcemi k ovládání v angličtině, které můžeme vidět na obrázku 7.6.



Obrázek 7.6: Okno s instrukcemi k ovládání

Kliknutím na **Help > About** se otevře okno s informacemi o aplikaci, které můžeme vidět na obrázku 7.7.



Obrázek 7.7: Okno s informacemi o aplikaci

# Bibliografie

- [AMA22] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. *International Conference on Engineering and Technology (ICET)*. 2022. Dostupné z DOI: 10.1109/icengtechnol.2017.8308186.
- [Can86] CANNY, John. A Computational Approach to Edge Detection. *Transactions on Pattern Analysis and Machine Intelligence*. 1986, roč. 8, s. 679–698. Dostupné z DOI: 10.1109/tpami.1986.476785.
- [Dat17] DATACLUSTERLABS. *Crowd in a rally | Crowd Counting | Crowd Human*. 2017. Dostupné také z: <https://www.kaggle.com/datasets/dataclusterlabs/religious-gathering-crowd-rally-crowd>.
- [Fan+22] FAN, Zizhu et al. A survey of crowd counting and density estimation based on convolutional neural network. *Neurocomputing*. 2022, roč. 472, s. 224–251. ISSN 0925-2312. Dostupné z DOI: <https://doi.org/10.1016/j.neucom.2021.02.103>.
- [Her23] HERBERG, Evelyn. Lecture Notes: Neural Network Architectures. 2023. Dostupné z DOI: <https://doi.org/10.48550/arXiv.2304.05133>.
- [HL10] HUANG, Wenhua Ma Lei; LIU, Changping. Crowd density analysis using co-occurrence texture features. *5th International Conference on Computer Sciences and Convergence Information Technology*. 2010. Dostupné z DOI: doi:10.1109/iccit.2010.5711051.
- [JMM96] JAIN, Anil K.; MAO, Jianchang; MOHIUDDIN, K. M. Artificial neural networks: a tutorial. *Computer*. Jain1996, roč. 29, s. 31–44. Dostupné z DOI: doi:10.1109/2.485891.
- [KMH22] KHAN, Muhammad Asif; MENOVAR, Hamid; HAMILA, Ridha. Crowd Density Estimation using Imperfect Labels. 2022. Dostupné z DOI: <https://doi.org/10.48550/arXiv.2212.01450>.

- [Kon+23] KONG, Weihang; SHEN, Jienan; LI, He; LIU, Jiayu; ZHANG, Junge. Direction-aware attention aggregation for single-stage hazy-weather crowd counting. *Expert Systems with Applications*. 2023, roč. 225, s. 120088. ISSN 0957-4174. Dostupné z DOI: <https://doi.org/10.1016/j.eswa.2023.120088>.
- [LSF19] LIU, Weizhe; SALZMANN, Mathieu; FUA, Pascal. Context-Aware Crowd Counting. 2019. Dostupné z DOI: <https://doi.org/10.48550/arXiv.1811.10452>.
- [Loy+13] LOY, Chen Change; CHEN, Ke; GONG, Shaogang; XIANG, Tao. Crowd Counting and Profiling: Methodology and Evaluation. In: *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective*. Ed. ALI, Saad; NISHINO, Ko; MANOCHA, Dinesh; SHAH, Mubarak. New York, NY: Springer New York, 2013, s. 347–382. ISBN 978-1-4614-8483-7. Dostupné z DOI: [10.1007/978-1-4614-8483-7\\_14](https://doi.org/10.1007/978-1-4614-8483-7_14).
- [Pri+16] PRIETO, Alberto et al. Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing*. 2016, roč. 214, s. 242–268. ISSN 0925-2312. Dostupné z DOI: <https://doi.org/10.1016/j.neucom.2016.06.014>.
- [Yua+22] YUAN, Lixian; CHEN, Yandong; WU, Hefeng; WAN, Wentao; CHEN, Pei. Crowd counting via Localization Guided Transformer. *Computers and Electrical Engineering*. 2022, roč. 104, s. 108430. ISSN 0045-7906. Dostupné z DOI: <https://doi.org/10.1016/j.compeleceng.2022.108430>.
- [ZNW08] ZHAO, Tao; NEVATIA, Ram; WU, Bo. Segmentation and Tracking of Multiple Humans in Crowded Environments. *Transactions on Pattern Analysis and Machine Intelligence*. 2008, roč. 30, s. 1192–1211. Dostupné z DOI: [10.1109/tpami.2007.70770](https://doi.org/10.1109/tpami.2007.70770).

# Seznam obrázků

2.1	Obrazovky fotoaparátu a vybírání předmětu . . . . .	7
2.2	Obrazovky s úpravou bodů a hlavní menu . . . . .	8
2.3	Okno aplikace DotDotGoose . . . . .	10
2.4	Okno s třídami předmětů . . . . .	11
2.5	Označování předmětů pomocí bodů . . . . .	12
2.6	Souhrn předmětů a možnosti exportu . . . . .	13
3.1	Variace hustoty davů, vedoucí k různým mapám hustoty [Yua+22] . . .	18
3.2	Naučené vlastnosti konvoluční neuronové sítě [AMA22] . . . . .	25
3.3	Konvoluce jako alternativa pro úplně připojenou síť [AMA22] . . . . .	25
3.4	Filtr se hýbe jen o jeden pixel, pro krok 1. [AMA22] . . . . .	26
5.1	Návrh hlavního okna . . . . .	29
5.2	Adresářová struktura projektu . . . . .	30
5.3	Formát exportovaného souboru . . . . .	33
5.4	Rozložení panelů levé části hlavního okna . . . . .	35
5.5	Snímek hlavního okna . . . . .	36
5.6	Snímek okna s nastavením bodů . . . . .	38
5.7	Snímek export okna . . . . .	39
5.8	Snímek okna s nastavením detekce osob . . . . .	40
5.9	Snímek okna s vysvětlivkami ovládání . . . . .	41
5.10	Snímek okna s informacemi o aplikaci . . . . .	42
6.1	Adresářová struktura testové sady . . . . .	43
6.2	Obsah souboru postup_testovani.txt . . . . .	44
6.3	Vybrané fotografie pro testování . . . . .	47
6.4	Počet označených osob na jednotlivých fotografiích . . . . .	49
6.5	Průměrný počet označených osob na fotografii . . . . .	49
6.6	Variační koeficient označeného počtu osob na fotografii . . . . .	50
7.1	Části uživatelského rozhraní . . . . .	52
7.2	Načtení nového obrázku . . . . .	53

7.3	Nastavení stylu bodů . . . . .	54
7.4	Export bodů a nastavení . . . . .	54
7.5	Nastavení metody pro automatickou detekci osob na fotografii . . . . .	55
7.6	Okno s instrukcemi k ovládání . . . . .	56
7.7	Okno s informacemi o aplikaci . . . . .	56

# Seznam tabulek

2.1	výčet vlastností jednotlivých aplikací . . . . .	15
6.1	počet označených osob na jednotlivých fotografiích testery . . . . .	48



# Zkratky

**CNN** - Convolutional neural network - konvoluční neuronová síť.

**MVC** - Model-view-controller - designový vzor pro rozdělení logiky aplikace.

**CMD** - Command line - příkazová řádka.

**HOG** - Histogram of oriented gradients - histogram orientovaných gradientů.

**GLCM** - Grey-level co-occurrence matrix - matice společného výskytu stupňů šedi.

**LBP** - Local binary pattern - lokální binární vzor.

**GOCM** - Gradient orientation co-occurrence matrix - matice společného výskytu orientace gradientu.

**SVM** - Support vector machines - metoda podpůrných vektorů.

**JDK** - Java development kit

**UTF-8** - Unicode transformation format - 8-bit

**HD** - High definition - vysoké rozlišení

1101001 1100001  
1010110001110010 1100001  
1010110101 10



11010011101101001  
01100001 10101  
1110001011101