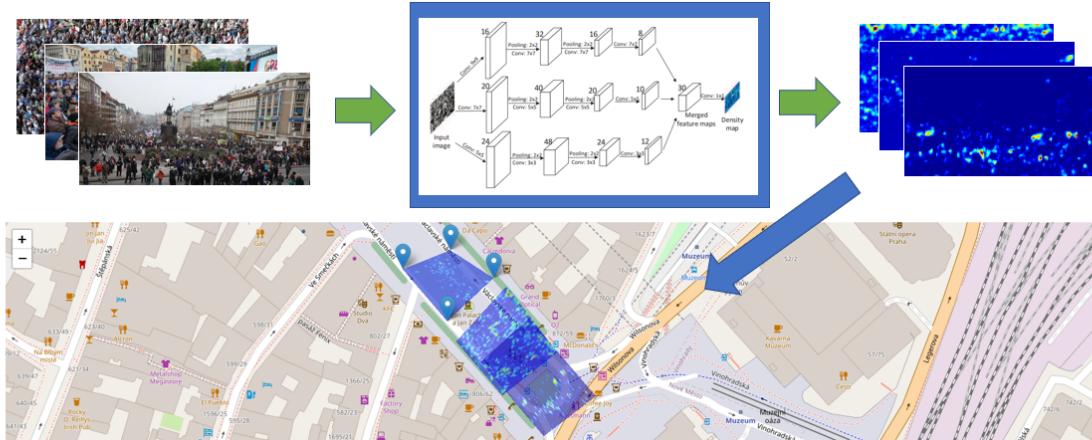


Počítání lidí v davu pomocí neuronových sítí s integrací do mapy

Adam Ferencz



Abstrakt

Cílem této práce je umožnit získat věrohodný odhad počtu lidí v davu na demonstraci či jiné hromadné akci z několika fotografií pořízených dronem, či jiných fotografií. Výsledkem jsou obarvené části mapy podle hustoty lidí v daném místě. Jednotlivé fotografie jsou dány do souvislosti s jejich umístěním do topologické mapy. Pro počítání lidí z fotky je použita metoda konvoluční neuronové sítě, která dokáže k fotografií vytvořit příslušnou mapu hustoty lidí. Integrace více fotek dohromady se pak provádí transformací do mapového podkladu, čímž se vezme v potaz i rozmištění davu v prostoru.

Aplikace je rozdělena na server a webový klient. Serverová část se stará o analýzu davu a vytvoření map hustoty k jednotlivým obrázkům. Toho dosáhne pomocí natrénovaného modelu neuronové sítě Multi-Column Convolutional Neural Network. Klient se pak kromě vstupních obrázků stará o jejich zobrazení do mapy, případně modifikaci dat. Výsledkem je interaktivní mapa, ve které jsou umístěny mapy hustoty davu.

Klíčová slova: Počítání lidí — Počítačové vidění — Konvoluční neuronové sítě — Keras — Leaflet — OpenStreetMap

Přiložené materiály: [Demonstrační video](#)

*xferen05@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

O co nejpřesnější určení celkového počtu lidí na demonstraci či na veřejné události mají velký zájem jak novináři, tak i policie. Aktuální postupy balancují mezi rychlosťí a přesností. Například pomocí "ručního" počítání lze dosáhnout poměrně přesného odhadu, ale za cenu dlouhého úsilí.

Problém je v tom, že při demonstraci nelze celý dav zaznamenat na jednu fotografiu. Často jsou lidé také zakryti stánky a budovami. Při získání více fotografií pak může být celý dav pokryt, ale je zde problém se souvislostí jednotlivých fotek mezi sebou. Proto je vhodné do řešení zahrnout mapu a ujasnit si tak rozložení davu v ní. Právě integrace mapy a přesné

analýzy fotek pomocí neuronové sítě může přinést věrohodný obrázek o proběhlé demonstraci.

Existující neuronové sítě jsou schopné v praxi poskytnout dobrý přehled o hustotě v dané oblasti. [1] Používají se ale především na jeden statický pohled, například pro zjištění, zda se konkrétní parkoviště nepřeplní. Většinou jejich využití končí u čísla - počtu lidí a jedné mapy hustoty. V případě využití topologických map se odhad počtu lidí v dosavadních řešeních musí odhadnout z počtu lidí na metr čtverečný a plochy, jako například u aplikace Mapchecking¹.

Vém řešení se kombinuje možnost analýzy obrázku neuronovou sítí s topografickou mapou. Neuronová síť podá údaj o počtu lidí na obrázku. Když se tento údaj vloží do mapy, získáváme z toho druhou zajíma-vou informaci, a to je, na jaké ploše se vyskytuje daný dav. Z toho lze pak odvodit hustotu davu na metr čtverečný a ne na jeden pixel, jak to je u samotného modelu neuronové sítě.

Vzniklá aplikace má především za cíl efektivně využít neuronových sítí pro počítání davu. Uživateli je umožněno pro konkrétní hromadnou akci analyzovat velikost a hustotu davu v topografické mapě.

2. Existující řešení v oblasti analýzy davu

V této kapitole popíší konkrétní existující aplikace pro analýzu a počítání davu. Metodám, které tyto aplikace či systémy využívají se pak bud věnovat v kapitole 3.

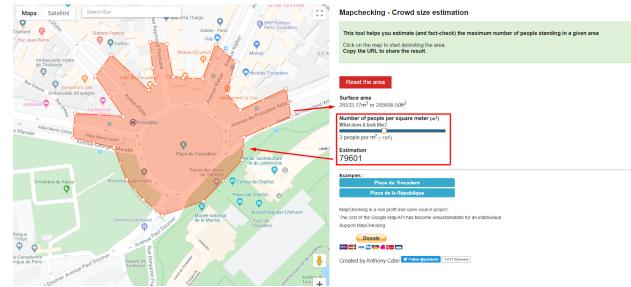
2.1 Mapchecking

Webová aplikace mapchecking.com (viz Obr. 1) pro odhad celkového počtu davu používá pouze mapu bez dalších složitých metod. Uživatel zadá polygonem do mapy plochu, kde se dav vyskytoval. Poté musí správně odhadnout, jaká byla hustota daného davu. K tomu by podle autora měly pomoci modelové ukázky nasimulovaných hustot davu dostupné na webu Crowd Safety and Risk Analysis², který se zabývá porozuměním bezpečnosti davu. Výhodou aplikace je její jednoduchost a rychlosť. Problém je ale v samotné nepřesnosti a nevěrohodnosti výsledků. I když má člověk dostupné fotografie daného davu, musí se na ně pouze podívat a odhadovat, jaká je hustota.

Právě na základě tohohle problému vznikala má myšlenka zapojit do řešení neuronové sítě, které počet a hustotu lidí dokáží odhadnou s dostačeně nízkou chybou.

¹<https://www.mapchecking.com/>

²<http://www.gkstill.com/Support/crowd-density/625sm/Density6.html>



Obrázek 1. Webová aplikace Mapchecking pro pomoc odhadu počtu lidí na demonstraci. Na obrázku vidíte označené území, kde se dav pravděpodobně nacházel. Na slideru vpravo je nastavena odhadnutá hustota davu na metr čtverečný. Z těchto dvou údajů je odvozen celkový počet lidí na demonstraci.

2.2 Systémy na analýzu davu

Například velký zájem o analýzu návštěvníků je v obchodních centrech, kde účelem je jak prevence bezpečnostního rizika a předpověď potřeby evakuace, tak i touha získávat informace o chování nakupujících, jejich pohlaví, atd.

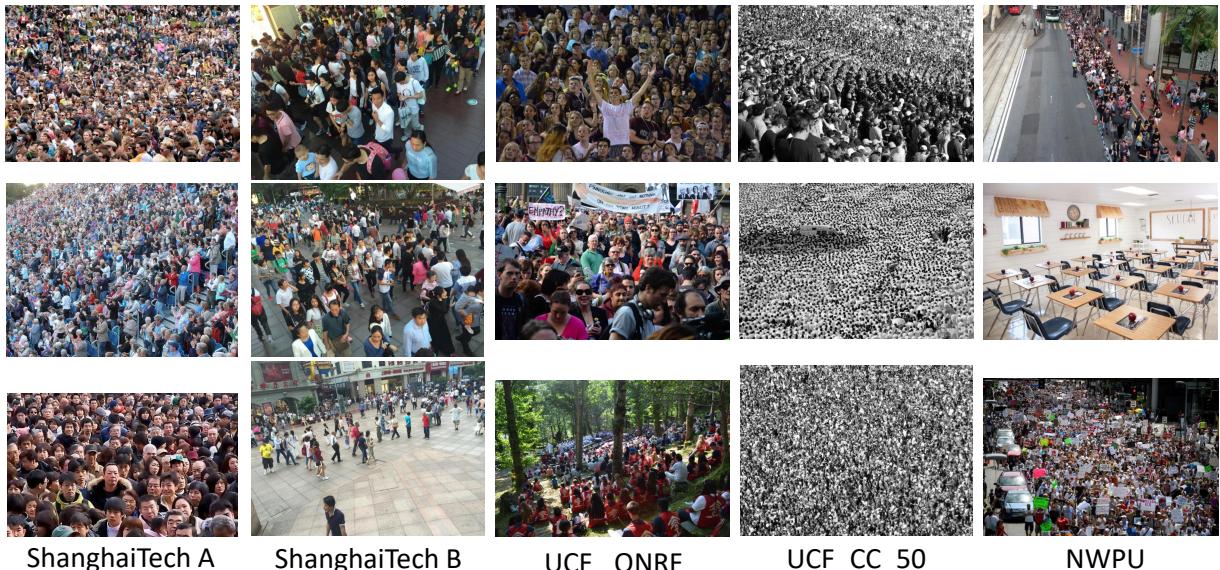
Společnost iOmniscient nabízí řešení pro komplexní analýzu chování davu v různých scénářích. Především se soustředí na CCTV bezpečnostní kamery, prevenci přelidnění a realtime monitoring. Kromě početních výstupů nabízí možnost mapy četnosti výskytu lidí a další analytická data.

V rámci projektu LETSCROWD³ vznikl komplexní nástroj pro dohled nad hromadnými akcemi. Ten sbírá data z bezpečnostních kamer pro předpovídání rizikových situací. Vše je konvertováno na serveru společně s dalšími informacemi jako je analýza příspěvků na sociálních sítích, hlášení policie a organizátorů. Mapa je zde použita především pro zobrazení výskytu kamer, vozidel či strážníků. Vše je zobrazováno v klientské aplikaci tak, aby vyšší autority měly dostatečný přehled o probíhající akci. Toto řešení je nejaktuálnější a stále se vyvíjející.

3. Metody počítání a analýzy davu z fotografie a videa

Metody řešící analýzu davu lze rozdělit na klasické a metody využívající neuronových sítí. **Klasické metody** se snaží analyzovat fotku či video pomocí detekce hran, segmentace popředí a pozadí, sledování toku pixelů, shlukováním, či hledají ručně stanovené příznamky. [5] Častým problémem těchto metod je, že jejich přesnost selhává při velmi hustých davech, nebo že potřebují video - sekvenci více po sobě jdoucích snímků ke svému fungování. Tyto problémy se snaží

³<https://letscrowd.eu/>



Obrázek 2. Ukázka nejvýznamnějších datasetů pro počítání davu. ShanghaiTech [2], UCF-QNRF [3], UCF_CC_50 [4] a NWPU-Crowd [1]

překonat přístupy založené na konvolučních neuronových sítích. [6]

Přístup pomocí konvolučních neuronových sítí je úspěšný, protože s dostatečnými datasety se model dokáže naučit rozpoznávat příznaky v různých situacích, a tak lépe generalizovat. Snižování nákladů a dostupnost zařízení pro pořizování dat ze vzduchu přinesl větší počet dostupných kvalitních obrazových datových sad. Datasety jsou anotované pomocí souřadnic bodů hlav lidí (head dot annotation), výjimečně pomocí boundingboxu. Mapa hustoty se pak generuje pro trénink pomocí konvoluce s gausovou maticí či komplexnějším algoritmem. Na obrázku 2 jsou ukázky nejvýznamnějších z nich. Aktuálně nejnovější dataset NWPU-Crowd [7] byl vydaný na začátku roku 2020, obsahuje 5109 anotovaných obrázků a slouží jako benchmark⁴ pro nynější architektury. Je možné říct, že situace s datasety pro počítání davu je velmi přívětivá.

Pro účely počítání davu se ukázalo vhodné upustit od detekce konkrétních jedinců a jejich počítání. [6] Místo toho se ukázalo vhodnější mapovat vstupní data na mapy hustoty. V mapě hustoty k danému obrázku pak hodnota určuje hustotu lidí na pixel. To zlepšilo výsledky především v analýze velmi hustých davů. Pokud se ve fotce davu vyskytuje více jedinců velmi blízko u sebe či za sebou, v mapě se to projeví vyššími hodnotami v daných pixelech a není potřeba se starat o to, že se jedinci vzájemně překrývají a nelze je jednotlivě přesně detektovat.

Modely byly nejdříve jednoduché s jedno sloupcové architekturou [6], kdy jedním sloupcem je myš-

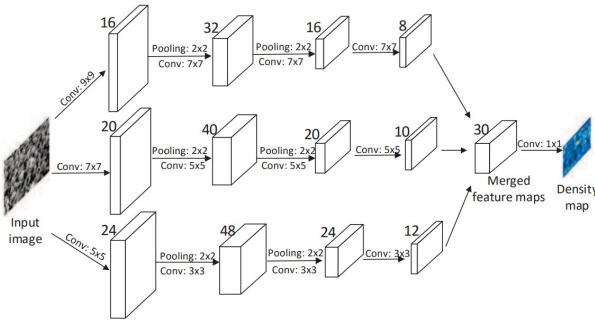
lena sekvence střídajících se konvolučních filtrů a max poolingu o různých parametrech. Dokázaly klasifikovat dav v daném místě obrazu na stupně hustoty od nízké po vysokou. [8] Problém, který bránil lepším výsledkům, byla perspektiva fotografie. Ta se pak projevovala rozdílnou velikostí lidských hlav, či těl. Řešením bylo přidání více různých sloupců do architektury neuronové sítě. Jejich kombinaci pro lepší výsledek lze pak provést více způsoby. Buď spojením jejich výstupu, nebo jejich přepínáním pro části obrazu.[6]

Multi-Column Convolutional Neural Network [2], zkráceně MCNN, ve své architektuře obsahuje tři sloupce neuronových vrstev (viz Obr. 3), které se liší jen svými parametry konvolučních filtrů. Výstupy sloupců jsou pak spojené a vytváří tak výstup celé neuronové sítě. Vstupní obrázek prochází současně přes všechny tři sloupce. Každý sloupec je citlivý na různou úroveň zvětšení hlav. Tři výstupy sloupců se pak dají dohromady a výsledkem je tedy mapa hustoty, která zohledňuje perspektivu fotografie.

Další možností je Switching Convolutional Neural Network [9], kde autoři natrénovali samostatně jednotlivé sloupce na datech, která jsou přesně v jejich optimálním přiblížení. Poté natrénovali přepínač, klasifikátor, který je schopný rozhodovat, jaký ze samostatně natrénovaných modelů je vhodné pro danou část obrázku použít.

Poslední z populárních architektur je využití multitask learningu, kdy je neuronová síť trénována na více různých pod-úloh. V případě počítání davu pak úlohami může být klasifikace úrovně hustoty a generování mapy hustoty, což použil Sindagi et al. [10].

⁴<https://www.crowdbenchmark.com/nwpucrowd.html>



Obrázek 3. Architektura Multi-Column Convolutional Neural Network představená Yingying Zhangem v roce 2016[2]

Ve své práci jsem použil již natrénovaný model s architekturou MCNN [2] od Priyanka Kasture⁵, která model natrénovala v rámci MindSpark Hackathon 2018 a zpřístupnila⁶.

Vyhodnocení přesnosti MCNN

Úspěch neuronových sítí pro počítání davu se hodnotí pomocí dvou metrik: MEA (Mean Average Error) a MSE (Mean Absolut Error),

$$MAE = \frac{1}{N} \sum_1^N |z_i - \hat{z}_i| \quad (1)$$

$$MSE = \sqrt{\frac{1}{N} \sum_N^1 (z_i - \hat{z}_i)^2} \quad (2)$$

kde N je počet testovacích obrázků, z_i je opravdový počet lidí v itém obrázku, a \hat{z}_i je stanovený počet lidí v itém obrázku. Obecně řečeno MAE vypovídá o přesnosti a MSE indikuje robustnost odhadů. K výpovědní síle této metriky je vždy potřeba dodat, na jakém datasetu podávala jaké výsledky. Je velmi pravděpodobné, že MCNN bude podávat nejlepší výsledky na Shanghaitech datasetu, na který byl designován. Generalizovat pak bude nejlépe na datech podobných jeho tréninku. Podle článku [2] jsem výsledky shrnul do tabulky 1. V tabulce můžete vidět výsledky MCNN v souvislosti s průměrným počtem lidí na jednom obrázku daného datasetu. Z toho lze odvodit, jaká chyba je pro jak hustý dav očekávatelná.

V závislosti na datasetu NWPU [1] pak MCNN podává následující výsledky v porovnání s předními ostatními architekturami. (viz Tab. 2) Je podstatné říct, že NWPU dataset je aktuálně nejkomplexnější dostupný dataset.

⁵<https://github.com/priyanka-kasture/Crowd-Counting-with-MCNNs>

⁶<https://github.com/priyanka-kasture/Crowd-Counting-with-MCNNs/blob/master/LICENSE>

Dataset	Prům. poč.	MAE	MSE
ShanghaiTech P. A	501.4	110.2	173.2
ShanghaiTech P. B	123.6	26.4	41.3
WorldExpo	50.2	11.6	-
UCF CC 50	1279.5	377.6	509.1
UCSD	24.9	1.07	1.35

Tabulka 1. Tabulka shrnující výsledky MCNN [2] na různých datasetech.

Metoda	MAE	MSE
MCNN [2]	232.5	714.6
C3F-VGG [11]	127.0	439.6
CSRNet [12]	121.3	387.8
CANNet [13]	106.3	386.5
SCAR [14]	110.0	495.3
SFCN+ [15]	105.7	424.1

Tabulka 2. Tabulka výsledků nejlepších metod na datasetu NWPU [7].

4. Architektura aplikace

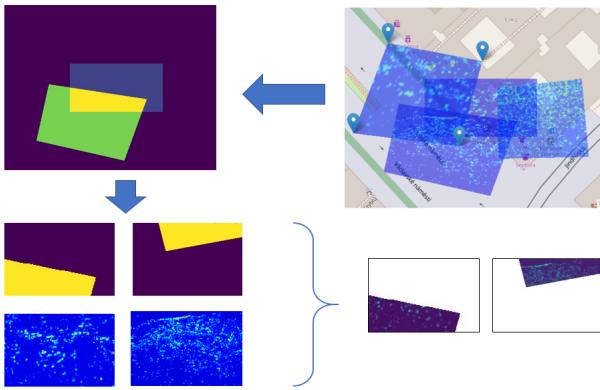
Uživatel chce spočítat celkový počet lidí na demonstraci z fotografií, které pořídil. Aplikace mu umožní nahrát a rozmištít fotografie v mapě a vypočítat hustotu davu v jednotlivých fotografiích. Pro rozmištěné fotografie je provedena korekce počtu u překryvajících se. Výsledkem je vizualizace rozmištění a počet lidí na demonstraci.

Architekturu aplikace tedy navrhoji jako klient-server s využitím webových technologií. Interakce s uživatelem bude realizována na klientovi a vlastní výpočet pomocí CNN na serveru.

Klientská část umožní nahrání fotografií a jejich umístění do mapy. Tato data je třeba synchronizovat se serverem. Ten bude počítat hustotu davu, převádět na počet osob, transformovat fotografie a provádět detekci překryvů fotografií, díky které poté vypočte redukovaný výsledný počet osob. Výsledky pak budou opět synchronizovány s klientskou částí, kde budou uživateli vizualizovány.

Model na serveru odhaduje počet lidí pro celou fotografií vytvořením mapy hustoty. V řadě případů se ale fotografie překrývají a výsledný počet lidí na akci by byl tedy zkreslený, proto je třeba provést korekci počtu osob v místě překryvů. Celkový počet by neměl být jen součet všech hodnot samostatných obrázků, ale musí brát z překryvů poměrnou část.

Je podstatné nalézt masky překryvů pro jednotlivé kombinace fotografií. Server si drží vypočtené mapy hustoty jednotlivých obrázků. Na ty se použije tato získaná maska a je možné tedy vypočítat samostatně počet lidí v místě překryvu. Všechny kombinace těchto vymaskovaných překryvů lze reprezentovat maticí, která je vizualizována na obrázku 5. V matici jsou pak



Obrázek 4. Schéma získání a použití masky pro korekci překryvu.

místo map hustoty již konkrétní počty lidí v místech překryvu. Výsledný počet se pak získá pomocí vzorce:

$$SUM = D - \frac{M - D}{2} = \frac{1}{2} \cdot (3 \cdot D - M) \quad (3)$$

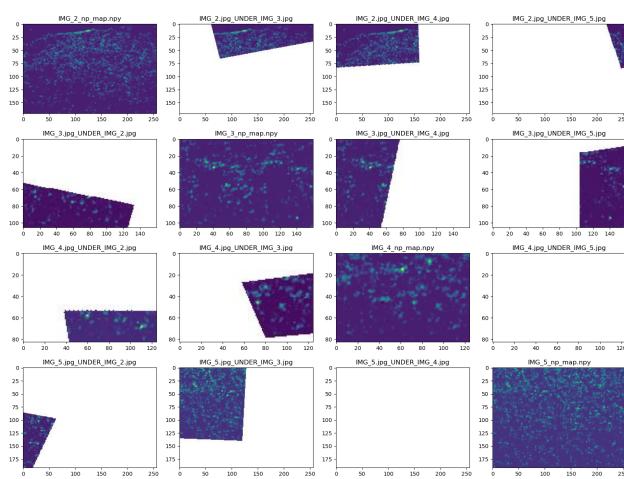
kde SUM je odhad s korekcí, D je součet prvků diagonály (zároveň také odhad před korekcí), M je součet celé matice.

Aby server věděl, jak jsou fotografie umístěné, je třeba zaslat polohu všech obrázků v mapě a s nimi i hranice mapového souřadného systému, na kterém se vyskytují. Nejdříve se vytváří masky pro překryvy transformovaných obrázků. Transformovaná mapa hustoty by ale dávala zkreslený počet. Je tedy třeba provést inverzní transformaci masky tak, aby byla použitelná na originální tvar mapy hustoty. Tento postup je znázorněn na schématu 4.

Server je implementován v jazyce Python. Komunikaci s klientem zajišťuje lehký WSGI⁷ webový aplikacní framework Flask⁸, který splňuje všechny

⁷Web Server Gateway Interface

⁸<https://pypi.org/project/Flask/>

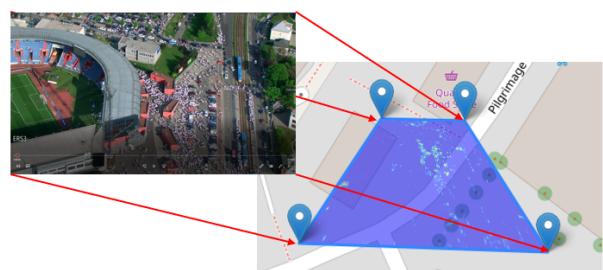


Obrázek 5. Vizualizace matice jednotlivých překryvů.

potřeby pro nasazení natrénovaného modelu. Na server přichází požadavky pro přepočítání obrazových vstupů na mapy hustoty. Ty se pak odesílájí k zobrazení zpět na klientskou stranu. Výhodou serveru je, že model se nedává na web veřejně přístupný v případě, že by ho bylo třeba udržet v soukromí. (Pokud by toto potřeba nebylo, nabízí se možnost model konvertovat do formátu, se kterým je schopná pracovat knihovna keras.js a aplikace by mohla být tak čistě klientská.) Natrénovaný model pak na serveru běží pomocí knihovny Keras. Další potřebné akce s obrazovými daty jsou pak prováděny pomocí knihovny pro počítačové vidění OpenCV a knihovny numpy.

Logika webového klienta je implementována v JavaScriptu. Zobrazení mapy zajišťuje Leaflet.js, což je nejrozšířenější open-source JavaScriptová knihovna k tvorbě interaktivních map. Mapu lze posouvat, přibližovat, oddalovat. Je možné zvolit různé zdroje mapového podkladu (Mapbox, OpenStreetMap, Stamen, ...), já jsem zvolil čistý OpenStreetMap.

Přesouvání obrázku a jeho transformaci v mapě (viz. Obr. 6) jsem vyřešil použitím Leaflet pluginu Leaflet.Path.Drag a pluginu Leaflet.ImageTransform.

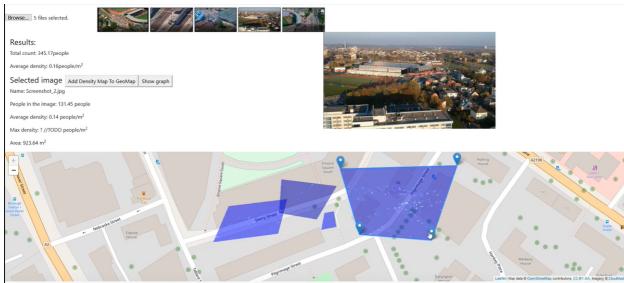


Obrázek 6. Ukázka perspektivní transformace mapy hustoty davu v obrázku do mapy podle čtyř určujících bodů.

5. Uživatelské prostředí

Pro uživatele je zásadní, aby získání celkového počtu lidí bylo co nejsnazší a intuitivní. V této kapitole popíši, jaké akce jsou od uživatele očekávané a kroky, které jsem provedl pro zjednodušení uživatelské činnosti.

Prvním krokem je výběr a nahrání fotografií davu, se kterými se dále pracuje. Poté má uživatel za úkol umístit do mapy jednotlivé obrázky podle 4 bodů, které korespondují se čtyřmi rohovými body obrázku. Obrázky posouvá jako celek, nebo je vhodně posunem krajních bodů perspektivně transformuje. Obrázkem je myšlena fotografie, nebo k ní korespondující mapa hustoty davu. Mezi těmito pohledy uživatel může přepínat.



Obrázek 7. Prototyp aplikace před pilotním testem.

Uživatel vidí, kolik je lidí na jednotlivých obrázcích a kolik je tedy celkově lidí na celé akci (suma jednotlivých obrázků s korekcí překryvů).

Tím, že se umístila mapa hustoty do topografické mapy, je možné vypočítat plochu, na které se dava vyskytuje. To může být zajímavý údaj, ze kterého lze vydovit hustotu vztaženou k metrům čtverečným.

První verze GUI, zobrazená na obrázku 7, reflektovala základní funkcionalitu aplikace. Vznikl tak MVP, se kterým byl proveden pilotní test (muž, 43 let, znalý práce s počítačem). Výstupem experimentu jsou následující požadavky na revizi GUI:

- Je potřeba, aby uživatel poznal, v jakém pořadí a proč má aplikaci ovládat. GUI musí reflektovat uživatelský proces.
- Mapa je klíčový interaktivní prvek (manipulace s fotkami), musí jí být dáně výrazně více prostoru.
- Vhodné zobrazovat vybraný obrázek a jeho mapu hustoty ve větším formátu, než jen náhledový.

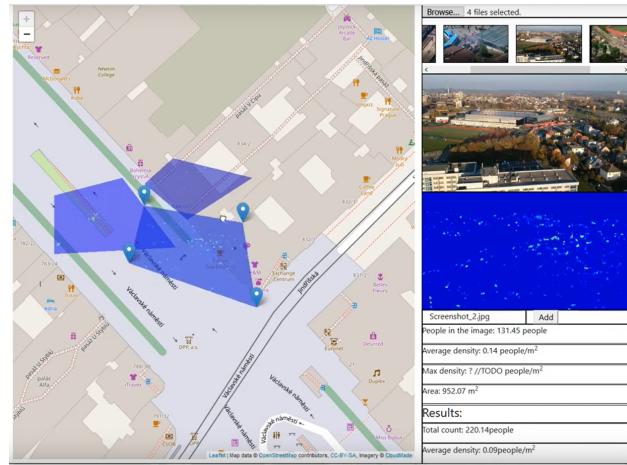
Po těchto požadavcích jsem navrhl 5 různých nových layoutů ve formě makety, ty jsem s účastníkem pilotního testu znova konzultoval a výsledné finální rozvržení implementoval (viz Obr. 8).

GUI je rozděleno na dva sloupce. V levém, který zabírá 60 procent celého prostředí, je zobrazena mapa. V pravém sloupci se nachází ovládací prvky. Od shora je zde umístěn výběr pro načtení fotografií, procházení náhledů načtených fotografií, velký náhled aktuální vybrané fotografie, velký náhled její mapy hustoty a na závěr informace o fotografii a celém projektu. Prvky v ovládání jsou seřazeny od shora dolů tak, jak se s nimi bude popořadě pracovat.

6. Závěr

Cílem práce bylo vytvořit aplikaci, která umožní spočítat lidi na hromadné akci s využitím topologické mapy, fotografií a neuronové sítě.

Své řešení jsem navrhl jako klient-server aplikaci. Server umožňuje nasazení natrénovaného modelu neuronové sítě s architekturou MCNN a komunikuje s



Obrázek 8. Výsledné GUI po pilotním testu.

klientem pomocí HTTP protokolu. Klient zajišťuje vstup dat, interakci s mapou a vizualizaci výsledku. Aplikace je schopná vzít v potaz překryvy umístěných fotografií.

Vzhledem k tomu, že téma mého projektu vzniklo jako jeden z pod-úkolů fakultního projektu VRASSEO, který se obecně zabývá analýzou davu, je v plánu rozšířit tuto demonstrační aplikaci o další moduly, například detekci anomálie v davu. V budoucnu by se kromě statických fotografií měla do mapy dát vizualizovat především videa z dronů. Takové video je schopné pokrýt větší území a poskytnout mnoho dalších informací, jako je nebezpečné chování, rychlosť davu, případně i detekce nebezpečných osob a jejich následná lokalizace v mapě. Bezpečnostní složky by pak měly možnost analyzovat situaci v reálném čase a lépe před-cházet nepříjemným situacím.

Poděkování

Velmi děkuji panu Ing. Vítězslavu Beranovi Ph.D. za jeho rady, které mi pomohly při tvorbě této práce.

Literatura

- [1] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting. *arXiv preprint arXiv:2001.03360*, 2020.
- [2] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [3] Haroon Idrees, Muhammad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss

- for counting, density map estimation and localization in dense crowds. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [4] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
 - [5] Sami Abdulla Mohsen Saleh, Shahrel Azmin Suandi, and Haidi Ibrahim. Recent survey on crowd density estimation and counting for visual surveillance. *Engineering Applications of Artificial Intelligence*, 41:103 – 114, 2015.
 - [6] Vishwanath A. Sindagi and Vishal M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3 – 16, 2018.
 - [7] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting, 2020.
 - [8] Min Fu, Pei Xu, Xudong Li, Qihe Liu, Mao Ye, and Ce Zhu. Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, 43:81 – 88, 2015.
 - [9] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
 - [10] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.
 - [11] Junyu Gao, Wei Lin, Bin Zhao, Dong Wang, Chenyu Gao, and Jun Wen. C³ framework: An open-source pytorch code for crowd counting. *arXiv preprint arXiv:1907.02724*, 2019.
 - [12] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018.
 - [13] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5099–5108, 2019.
 - [14] Junyu Gao, Qi Wang, and Yuan Yuan. Scar: Spatial-/channel-wise attention regression networks for crowd counting. *Neurocomputing*, 363:1–8, 2019.
 - [15] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8198–8207, 2019.