



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**VERIFIKACE OTISKŮ PRSTŮ S VYUŽITÍM GRAFOVÝCH
NEURONOVÝCH SÍTÍ**

FINGERPRINT RECOGNITION WITH GRAPH NEURAL NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ POSPÍŠIL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL HRADIŠ, Ph.D.

BRNO 2022

Zadání diplomové práce



Student: **Pospíšil Ondřej, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Strojové učení
Název: **Verifikace otisků prstů s využitím grafových neuronových sítí**
Fingerprint Recognition with Graph Neural Networks
Kategorie: Zpracování obrazu
Zadání:

1. Prostudujete základy grafových neuronových sítí.
2. Prostudujete základy verifikace otisků prstů.
3. Vyberte nebo navrhnete vhodnou metodu pro verifikaci otisků prstů využívající grafových neuronových sítí.
4. Obstarejte si datovou sadu vhodnou pro natrénování a vyhodnocení zvolené metody.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- R. Wang, J. Yan and X. Yang: Neural Graph Matching Network: Learning Lawler's Quadratic Assignment Problem with Extension to Hypergraph and Multiple-graph Matching, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- Michal Rolínek et al.: Deep Graph Matching via Blackbox Differentiation of Combinatorial Solvers, 2020.

Při obhajobě semestrální části projektu je požadováno:

- Body zadání 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**
Konzultant: Sochor Jakub, Ing., UPGM FIT VUT
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 18. května 2022
Datum schválení: 1. listopadu 2021

Abstrakt

Tato diplomová práce řeší verifikaci otisků prstů na základě jejich grafové reprezentace. Navržená metoda využívá grafovou neuronovou síť a kombinatorický solver pro získání přiřazení mezi markanty páru otisků. Přiřazené markanty jsou použity pro zarovnání otisků pomocí odhadnuté transformace algoritmem RANSAC. Zarovnané otisky jsou zpracovány modelem SimGNN. Výsledné skóre podobnosti je pak zkombinováno s metrikami získanými ze zarovnaných otisků prstů. Přínosem této práce je nový stabilní způsob zarovnání otisků pomocí vyřešení problému grafového přiřazení. Navrhovaný způsob verifikace nedosahuje vysoké přesnosti z důvodu příliš malého počtu atributů markantů a slabé rozlišovací schopnosti použitých metrik.

Abstract

This thesis deals with the verification of fingerprints based on their graph representation. The proposed method uses a graph neural network and a combinatorial solver to obtain the matching between the minutae points of a pair of fingerprints. The matched minutae points are used to align the fingerprints using an estimated transformation by the RANSAC algorithm. The aligned fingerprints are processed by the SimGNN model. The resulting similarity score is then combined with the metrics obtained from the aligned fingerprints. The experiments summarize the selection of method parameters and the evaluation of fingerprint matching and verification accuracy. The contribution of this work is a new stable method of fingerprint alignment by solving the graph matching problem. The proposed verification method does not achieve high accuracy due to too few minutae attributes and poor discriminating power of the metrics used.

Klíčová slova

verifikace otisků prstů, grafové neuronové sítě, párování grafů, podobnost grafů

Keywords

fingerprint verification, graph neural networks, graph matching, graph similarity

Citace

POSPÍŠIL, Ondřej. *Verifikace otisků prstů s využitím grafových neuronových sítí*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, Ph.D.

Verifikace otisků prstů s využitím grafových neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše, Ph.D. Další informace mi poskytl pan Ing. Jakub Sochor Ph.D. a firma Innovatrics. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Ondřej Pospíšil
14. května 2022

Poděkování

Rád bych poděkoval Ing. Jakubovi Sochorovi Ph.D. a firmě Innovatrics za odbornou pomoc a poskytnuté prostředky. Také bych rád poděkoval Ing. Michalovi Hradišovi za vedení práce. Dále bych chtěl poděkovat své přítelkyni a rodičům za morální podporu a poskytnuté zázemí.

Obsah

1	Úvod	2
2	Otisky prstů	3
2.1	Předzpracování otisku prstu	4
2.2	Markanty	5
2.3	Rozpoznání otisků prstů	8
2.4	Grafová reprezentace	11
3	Grafové neuronové sítě	14
3.1	Vstupní graf	15
3.2	Architektura	16
3.3	Konvoluční GNN	18
3.4	Problém kvadratického přiřazování	22
4	Navrhované řešení	25
4.1	Data	26
4.2	Přiřazení markantů dvojice otisku	31
4.3	Verifikace otisku	32
5	Experimenty	35
5.1	Přiřazování markantů	35
5.2	Transformace	38
5.3	Verifikace	38
5.4	Shrnutí experimentů	43
6	Závěr	46
	Literatura	47
A	Obsah příloženého paměťového média	51

Kapitola 1

Úvod

Otisky prstů patří mezi nejpoužívanější biometrické vlastnosti. Otisk prstů je pro každého člověka unikátní a nemění se s časem. Otisky prstů se dnes používají nejen v kriminalistice, ale můžeme se s nimi setkat i v každodenním životě například při odemknutí mobilního zařízení nebo dveří domu. Uživatel se zaregistruje do systému pomocí svého otisku prstu a při dalším přihlášení se jeho otisk porovnává s již uloženým vzorem v databázi. Tento proces se nazývá verifikace.

Současné metody nejčastěji využívají pro práci s otisky prstu markanty. Markanty jsou klíčové body, které jsou vyjádřeny pomocí jejich pozice, úhlu a typu. Tradiční metody používají pro porovnání markantů globální a lokální překryv. Otisk prstu lze reprezentovat i pomocí grafu, který je tvořen markanty. Problém verifikace otisků prstů lze poté převést na problém párování nebo problém podobnosti dvou grafů. S růstem popularity umělé inteligence začíná přibývat metod využívajících neuronové sítě, trendem posledních let se staly grafové neuronové sítě.

Tato práce shrnuje současné metody pro verifikaci otisků prstů a uvádí do problematiky grafových neuronových sítí. Poté je předloženo vlastní navržené řešení verifikace pomocí grafových neuronových sítí. Řešení se skládá ze zarovnání otisků a z jejich porovnání. Otisky prstů jsou vyjádřeny ve formě grafu. Vrcholy grafu jsou reprezentovány markanty otisků. Závislost mezi vrcholy je určena Delaunayho triangulací. Grafy otisků jsou zpracovány grafovou neuronovou sítí a předloženy na vstup kombinatorickému solveru, řešící problém grafového přiřazení. Výstupem solveru je optimální přiřazení markantů. Z přiřazení markantů jsou pomocí RANSAC odhadnuty parametry transformace zarovnání. Ze zarovnaných otisků je pomocí rozdílů úhlů markantů určen poměr správných přiřazení. Poměr správných přiřazení je zkombinován se skóre podobnosti získaným zpracováním zarovnaného páru otisků grafovou neuronovou sítí a výsledné skóre je použito pro verifikaci.

Struktura práce je rozložena do čtyř hlavních kapitol. V první kapitole je uvedena reprezentace otisků prstů pomocí markantů, existující metody verifikace otisků a různé přístupy grafové reprezentace otisku prstů. Druhá kapitola obsahuje úvod do grafových neuronových sítí a přehled existujících metod, které je možné využít k verifikaci otisků prstů. Třetí kapitola obsahuje popis datové sady otisků prstů a popis navrženého řešení. Poslední kapitola obsahuje experimenty, které zdůvodňují rozhodnutí volby jednotlivých částí architektury modelu, vyhodnocují použité metody a verifikaci na datové sadě otisků prstů.

Kapitola 2

Otisky prstů

Dlaně rukou obsahují zvláštní zvrásnění kůže, kterým říkáme papilární linie. Papilární linie jsou definovány mezi pokožkou a škárkou a lze je vidět na povrchu prstu. Výška papilárních linií leží v rozmezí 0.1–0.4 mm a jejich šířka v rozmezí 0.2–0.6 mm [13]. Otisk prstu je vzor tvořený obrazem papilárních linií. Otisky prstů můžeme rozdělit na válené, píchané a latentní otisky [12]. Liší se ve způsobu snímání. Válený otisk je snímán rolováním prstu na ploše senzoru, píchaný otisk – přiložením a latentní (skrytý) otisk je otisk zanechaný na různé ploše. Senzory využívají válené a píchané typy otisků, latentní otisk se využívá například v kriminalistice. Příklady těchto otisků lze vidět na obrázku 2.1.

Otisky prstů můžeme rozdělit i podle jejich třídy. Třída otisku je určena na základě polohy a počtu speciálních prvků v otisku – jádra, delty a typových linií. Tyto vzory lze v otisku prstu identifikovat a podle nich určit jejich třídu [22]. Třídy otisků:

- Oblouk (*Arch*)
- Klenutý oblouk (*Tended Arch*)
- Spirála (*Whorl*)
- Levá/pravá smyčka (*Left/Right Loop*)
- Dvojitá smyčka (*Twin Loop*)



Obrázek 2.1: Příklady váleného, píchaného a latentního otisku [12].

2.1 Předzpracování otisku prstu

Otisk prstu je nasnímán senzorem nebo načten z fotografie. Snímek otisku prstu obsahuje velké množství šumu. Povrch prstu může obsahovat znečištění, poranění nebo je jeho vzhled ovlivněn kožní nemocí. Z tohoto důvodu je nutné snímek otisku prstu dále zpracovat a extrahovat pouze papilární linie. Standardně se celý proces se skládá ze tří kroků. Určení pole orientací, extrahování linií a jejich ztenčení [12]. Proces je znázorněn na obrázku 2.2.

Pole orientací

V každém pixelu obrazu otisku je na základě okolí pixelu spočtena orientace papilární linie. Tyto hodnoty orientace jsou transformovány do blokového pole orientací. Pomocí pole je možné vyfiltrovat vstupní obrázek na základě orientace papilární linie v daném bloku pole.

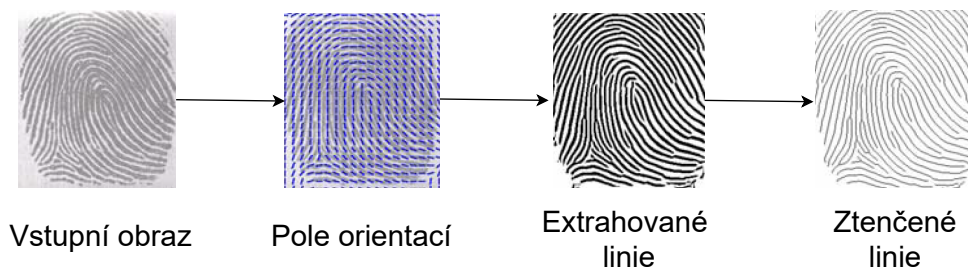
Extrahované linie

Extrakce linií probíhá úpravami histogramu. Obraz je filtrován 2D Gaborovou funkcí a filtrován ve frekvenční doméně pomocí následujících filtrů: dolní propust, *Butterworth*, *Ikonomopoulos* nebo *Sherlock* filtr. Detekce papilárních linií se provádí prahováním podle schéma *RAT* (*Regional Average Thresholding*).

Algoritmus *RAT* rozdělí obrázek na 8×8 bloky, pro každý blok je spočtena průměrná úroveň barevné intenzity stupni šedi a je uložena v levé části bloku (8×4). Operační okénko je posouváno po celém obraze o 4 body doprava a při dosažení pravého okraje o 8 bodů dolů.

Ztenčené linie

Extrahované papilární linie z předchozího kroku mají různou šířku, pro účel dalšího zpracování je nutné tyto linie ztenčit na hodnotu 1 pixelu. Pro ztenčení papilárních linií se nejčastěji používá metoda *Emyroglu* [11], která používá dva typy bodů, *Ridge Meeting Point* a *Ridge Continuity Point*.



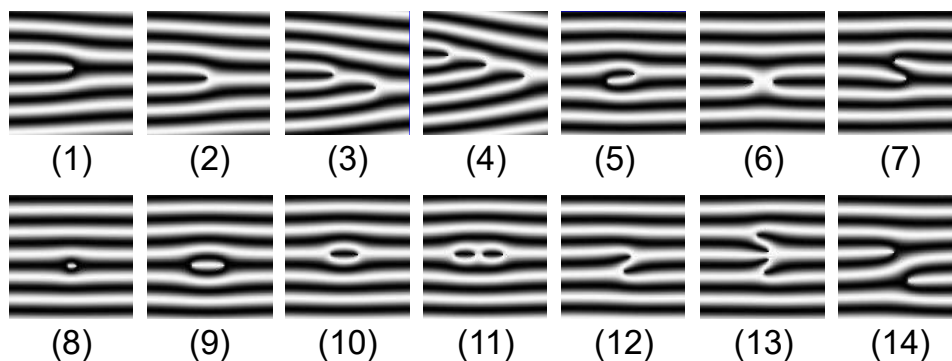
Obrázek 2.2: Znázornění postupu předzpracování otisku [12].

2.2 Markanty

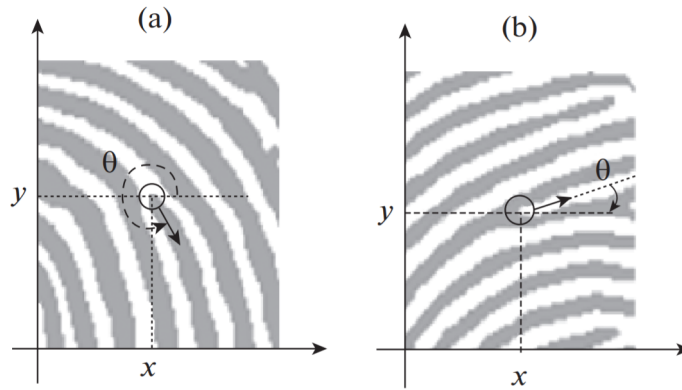
Nejčastěji se pro práci s otisky prstů používají markanty. Markanty jsou klíčové body získané ze snímku otisku prstu. Z jednoho snímku je obvykle možné extrahovat přibližně 25–80 markantů. Počet markantů závisí na poloze prstu a rozlišení použitého senzoru [33]. Markanty identifikují místa, kde dochází k větvení nebo ukončení papilárních linií [22]. Můžeme je rozdělit na čtrnáct základních typů:

1. Ukončení (*ridge ending*)
2. Jednoduchá vidlička (*bifurcation*)
3. Dvojitá vidlička (*double bifurcation*)
4. Trojitá vidlička (*triple bifurcation*)
5. Hák (*spur*)
6. Křížení (*ridge crossing*)
7. Boční kontakt (*opposed bifurcation*)
8. Bod (*dot*)
9. Interval (*island*)
10. Jednoduchá smyčka (*enclosure*)
11. Dvojitá smyčka (*double whorl*)
12. Jednoduchý most (*bridge*)
13. Dvojitý most (*twin bridge*)
14. Průsečná linie (*through line*)

Příklady těchto typů lze vidět na obrázku 2.3. V daktyloskopii se ovšem používají i další typy markantů. Naopak přístupové systémy využívají pouze první dva typy – ukončení a vidličku. Všechny ostatní typy je možné vyjádřit jako kombinaci těchto dvou typů.



Obrázek 2.3: Příklady základních typů markantů [12].



Obrázek 2.4: (a) typ ukončení, (b) typ vidlička : x, y je pozice a θ je orientace markantu [1].

Ke každému markantu uloží algoritmus extrakce jeho polohu (souřadnice X, Y), orientaci pokračování papilární linie (úhel), typ a kvalitu. Určení orientace u dvou základních typů markantů lze vidět na obrázku 2.4.

Extrakce markantů

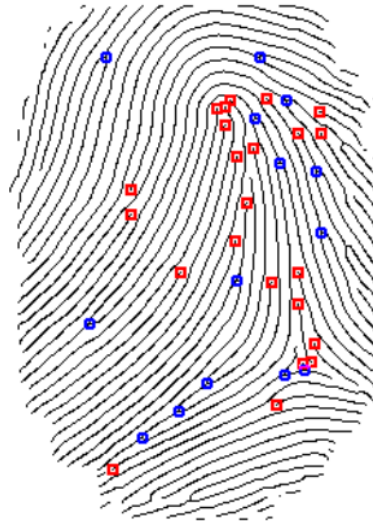
Nejpoužívanější metodou pro extrakci markantů je metoda využívající *Crossing Number (CN)* [29]. Metoda vyžaduje vstup v podobě ztenčených papilárních linií. Proces získání těchto linií byl uveden v kapitole 2.1. Pro každý pixel papilární linie je provedena analýza jeho lokálního okolí pomocí 3×3 okénka. Hodnota CN je spočtena jako polovina součtu rozdílu mezi páry pixelů v okolí [33]. Výpočet CN je vyjádřen rovnicí

$$\begin{array}{|c|c|c|} \hline P_4 & P_3 & P_2 \\ \hline P_5 & P & P_1 \\ \hline P_6 & P_7 & P_8 \\ \hline \end{array} \quad CN_P = \frac{1}{2} \sum_{i=1}^8 |P_i - P_{i+1}|, \quad P_9 = P_1. \quad (2.1)$$

Na základě hodnot CN jsou poté extrahovány markanty. Význam jednotlivých hodnot CN lze vidět v tabulce 2.1. Příklad extrahovaných markantů je pak znázorněn na obrázku 2.5.

CN	Vlastnost
0	izolovaný bod
1	ukončení
2	pokračování
3	vidlička
4	křížení

Tabulka 2.1: Detekce markantů na základě hodnoty *Crossing Number*.



Obrázek 2.5: Příklad detekovaných markantů na ztenčených papírných liniích. Typ ukončení je zobrazen červeně a typ vidlička modře [39].

Šablona pro uložení markantů

Z důvodu zajištění kompatibility je potřeba pevně definovat formát uložení otisků prstů. Pro uložení markantů se používá standardizovaná šablona. Šablonu popisuje například standard *ISO/IEC 19794-2:2005* [18]. Šablona je binární soubor skládající se z jednoduchých polí uspořádaných do skupin. Jednotlivá pole nebo skupiny se mohou opakovat. Pole jsou identifikovány pouze podle jejich pozice v šabloně. Hodnoty jsou uloženy v pořadí *big-endian*.

Šablonu můžeme rozdělit podle hierarchie na hlavičku šablony, otisk prstu, markant a rozšíření [34]. Hlavička šablony obsahuje informace o senzoru, rozlišení senzoru, velikost výstupu senzoru a počet uložených otisků prstů. Pro každý otisk je uložena informace ze kterého prstu pochází, kvalita a počet uložených markantů. Otisk dále obsahuje jednotlivé markanty a informaci o rozšíření. Pro každý markant je uvedena jeho pozice, úhel, typ a kvalita. Rozšíření šablony je volitelné a může obsahovat například informaci o hranách mezi markanty nebo detekované delty a jádra.

Markant je uložen ve struktuře o velikosti 6 bytů [34]. Rozložení vlastností ve struktuře znázorňuje tabulka 2.2. Typ markantu může nabývat tří hodnot – ukončení, vidlička a ostatní. Každá složka souřadnic markantu je uložena na 14 bitech. Pozice musí být uvedena u každého typu markantu, u složitějších typů záleží na implementaci extrakce. Úhel markantu je normalizován do rozsahu 0–255 (8 bit) a jeho určení bylo znázorněno na obrázku 2.4. Kvalita markantu je volitelná a je uvedena v rozsahu 1–100, kde 100 vyjadřuje nejlepší kvalitu. Pokud implementace extrakce markantů neuvádí kvalitu, je toto pole prázdné.

Standard *ISO/IEC 19794-2:2005* vychází ze standardu *ANSI 378-2004*. Uložení markantů v těchto šablonách se mírně liší. Standard *ISO* umožňuje různé vyjádření pozice bodu, v *ANSI* je pozice bodů pevně daná. Naopak pozice ostatních markantů v *ISO* je blíže specifikována aby se shodovala s výpočet u typu ukončení a vidličky. Úhel markantu je v *ANSI* šabloně vyjádřen v rozsahu 0–179. Šablona *ISO* využívá pro uložení úhlu celý rozsah 8 bitů.

Název	Kód	Délka (bit)
Typ	MINTYPE	2
Pozice X	MINX	14
Pozice Y	MINY	14
Úhel	MINANGLE	8
Kvalita	MINQUALITY	8

Tabulka 2.2: Uložení markantu v šabloně *ISO/IEC 19794-2:2005* [34].

2.3 Rozpoznání otisků prstů

Proces verifikace otisků prstů v biometrickém systému začíná registrací uživatele do systému. Uživatel se zaregistruje svým otiskem prstu, ten je poté uložen do databáze šablon. Formát uložené šablony záleží na použité metodě pro porovnávání otisků. Metody porovnání můžeme rozdělit na metody založené na korelaci [5], vlastnostech papilárních linií [12], rozpoznávání vzorů [2, 6] a markantech [28, 19].

Metody založené na korelaci

Metody založené na korelaci provádějí 2D korelaci mezi vstupním otiskem a uloženou šablonou. Pro porovnání jsou použity snímky otisku v odstínech šedi. Tento proces je výpočetně velmi náročný. Metody využívající korelaci lépe zvládají horší kvalitu otisků, které jsou špatně nasnímané nebo poškozené. Z těchto otisků často není možné extrahovat všechny markanty. *Robust Correlation* [5] používá pro korelaci referenční bod, nalezený v obou porovnávaných otiscích (například jádro). Na základě vzdálenosti od tohoto bodu provede určitý počet korelací.

Metody založené na vlastnosti papilárních linií

Tyto metody používají k porovnání otisků orientaci a frekvenci papilárních linií, tvar linie, nebo informace o textuře. Obecně mají nízkou rozlišovací schopnost. Příkladem může být metoda *Ridge Count*, která využívá počet papilárních linií mezi dvěma definovanými body v otisku prstu [12]. Nejčastěji mezi jádrem a deltou, nebo na základě několika bodů v pevně definovaných intervalech, jak lze vidět na obrázku 2.6.

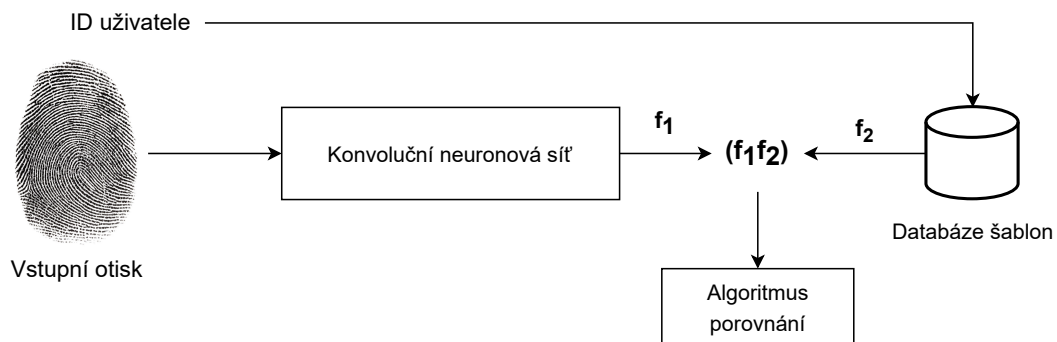
Metody založené na rozpoznání vzorů

Rozpoznávání vzorů je provedeno ze snímku otisku prstu. Z nasnímaných otisků jsou extrahovány příznaky, které jsou následně porovnány. Pro extrakci a porovnání příznaků je možné použít neuronové sítě.

Často bývá pro extrakci příznaků použita konvoluční neuronová síť [2, 6]. Obrázky otisků jsou zpracovány a přiloženy na vstup konvoluční neuronové sítě. Z každého snímku je extrahován vektor příznaků. Tyto vektory jsou poté použity pro porovnání obou otisků. Do databáze šablon se potom ukládá pouze vektor příznaků. Velikost vektoru závisí na architektuře neuronové sítě. Architekturu takového systému lze vidět na obrázku 2.7.



Obrázek 2.6: Zpracovaný otisk algoritmem *Ridge Count*, který používá pevně definované intervaly [12].

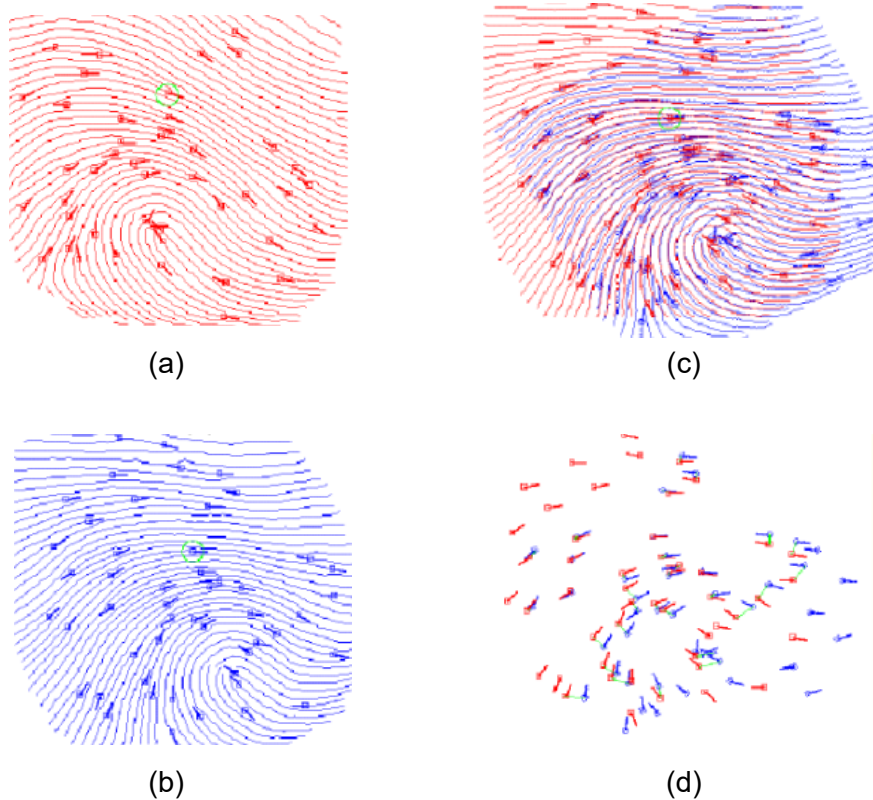


Obrázek 2.7: Systému využívající konvoluční neuronovou síť pro extrakci příznaků. Zde je uveden proces přihlášení uživatele do systému.

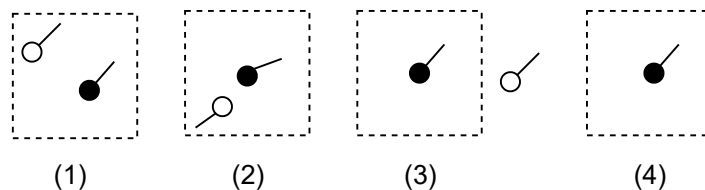
Metody založené na markantech

Nejčastěji používané metody pro porovnání otisků prstů pracují s markanty. Rozpoznání otisků probíhá porovnáním dvou množin markantů, je tedy nutné otisk zarovnat na šablonu. Pro nalezení vhodného přiřazení nemůžeme vyzkoušet všechny možnosti (*brute-force*), jelikož počet možných přiřazení roste exponenciálně s počtem markantů [23]. Cílem těchto metod je nalezení transformace, která markanty vstupního otisku zarovná na otisk šablony. Tento proces je znázorněn na obrázku 2.8.

Pro přiřazení markantů lze použít metodu Ratha [28] nebo metodu Hong [19]. Metoda Ratha je založena na dvou krocích: generování globálního překryvu (zarovnání) a hledání lokálního překryvu (porovnání). Zarovnání probíhá nalezením parametrů geometrické transformace ve čtyřrozměrném Houghově prostoru. Zadáním parametrů transformace (změna měřítka, translace, rotace) lze provést Houghovu transformaci na množině markantů vstupního otisku prstu a tím dostat zarovnání markantů na šablonu.



Obrázek 2.8: Proces zarovnání dvou otisků prstů. Obrázky (a,b) znázorňují detekované markanty v otiscích a obrázky (c,d) již zarovnané otisky podle jejich markantů [12].



Obrázek 2.9: Postup přiřazení markantů Rathovy metody [28].

Druhým krokem Rathovy metody je hledání lokálního překryvu. Ze všech zarovnaných párů markantů vybere pouze ty správné. Správnost přiřazení markantů je rozhodnuta na základě oblasti tolerance (*bounding-box*) bodu ze vstupního vzorku a na základě orientace obou bodů. Rozlišujeme čtyři situace, které jsou zobrazeny na obrázku 2.9. Body jsou přiřazeny pouze v prvním případě:

1. Vzorový bod leží v oblasti tolerance a oba body mají stejnou orientaci.
2. Vzorový bod leží v oblasti tolerance a oba body mají odlišnou orientaci.
3. Vzorový bod neleží v oblasti tolerance.
4. Vstupnímu bodu nebyl přiřazen vzorový bod.

Hongova metoda [19] také pracuje na principu zarovnání a porovnání. Pro použití této metody je nutné při extrakci markantů uložit i korespondující papilární linii. Linie je repre-

zentována jako křivka, která má počátek v souřadnicích markantu. V prvním kroku jsou parametry transformace spočteny pomocí korespondujících linií a je určen referenční markant, vůči kterému se provádí transformace. Každý bod je převeden do souřadnic polárního systému. Výpočet souřadnic vyjadřuje rovnice

$$\begin{pmatrix} r_i \\ e_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i^* - x^r)^2 + (y_i^* - y^r)^2} \\ \tan^{-1} \left(\frac{y_i^* - y^r}{x_i^* - x^r} \right) \\ \theta_i^* - \theta^r \end{pmatrix}, \quad (2.2)$$

kde $(x_i^*, y_i^*, \theta_i^*)$ jsou souřadnice markantu k převodu, (x^r, y^r, θ^r) jsou souřadnice referenčního markantu a (r_i, e_i, θ_i) jsou souřadnice markantu v polárním systému (r_i je vzdálenost mezi markanty, e_i je úhel mezi markanty a θ_i reprezentuje rozdíl orientace).

Souřadnice vstupních i vzorových bodů jsou seřazeny podle úhlu a konkatenovány do řetězců. Mezi vstupním a vzorovým řetězcem je nalezena editační vzdálenost pomocí algoritmu dynamického programování [8]. Obdobně jako u Rathovy metody je ještě určen počet validních bodů, které se nachází v toleranční oblasti vzorového bodu. Výsledné přiřazení je spočteno pomocí editační vzdálenosti a počtu validních markantů.

2.4 Grafová reprezentace

Při práci s otisky prstů se nejčastěji používají markanty. Markanty definují pouze lokální parametry každého bodu. Množina markantů neobsahuje závislosti mezi jednotlivými body. Otisk prstu můžeme reprezentovat grafem. Graf je uspořádaná dvojice $G = (V, E)$, kde V je množina vrcholů (uzlů) a $E = \{(n_1, n_2) | n_1, n_2 \in N\}$ je množina hran.

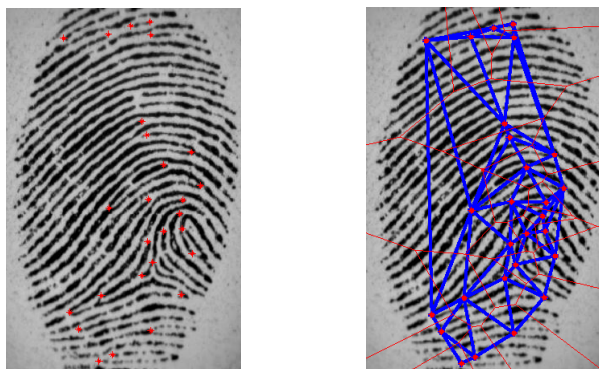
V případě vytváření grafové reprezentace otisku lze definovat V jako množinu markantů extrahovaných z kostry otisku [38]. Větší volnost poskytuje definice hran, kde existuje více způsobů definování závislosti mezi vrcholy. Existují metody, které jako vrcholy grafu používají úseky papilárních linií [16], nebo metody které markanty k vytvoření grafů vůbec nepoužívají [25, 17].

Triangulace

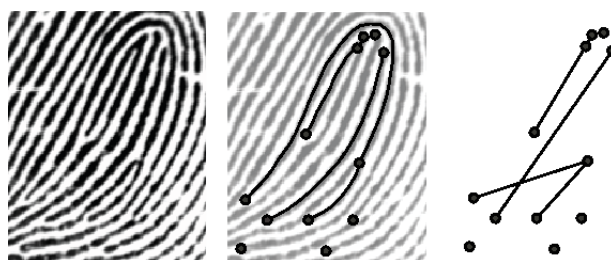
Triangulace je rozdělení konvexní obálky množiny bodů V na 2-simplexy (trojúhelníky) s vrcholy v bodech V . Při triangulaci je žádoucí, aby trojúhelníky byly co nejvíce rovnostranné. Jednou z metod triangulace je Delaunayho triangulace, která vytváří trojúhelníky pomocí Voroného diagramu. Voroného diagram rozdělí 2D prostor na regiony takové, že každý region okolo bodu p obsahuje pouze body, které jsou blíže p než k jakémukoli jinému bodu z V [4]. Triangulaci lze použít i pro verifikaci dvou otisků [26, 24]. Při zarovnání otisků se používá porovnání vlastností jednotlivých trojúhelníků (úhly, velikost a typ markantu vrcholů). Příklad aplikace triangulace na otisk prstů lze vidět na obrázku 2.10.

Okolí markantu

S myšlenkou použít pro definici hran vzdálenost jednotlivých markantů přišel již Ratha [27]. Definuje okolí bodu jako množinu bodů, které se nacházejí do určité vzdálenosti od daného bodu. Tyto body určují tzv. *star* okolí, které je spočteno pro každý markant otisku. Při porovnání dvou otisků jsou porovnávány *star* okolí jednotlivých markantů. Na tuto práci



Obrázek 2.10: Příklad triangulace markantů otisků prstů, na pravém obrázku je znázorněn Voroného diagram (červeně) a triangulace (modře) [38].



Obrázek 2.11: Horadam [16] grafová reprezentace otisku prstu.

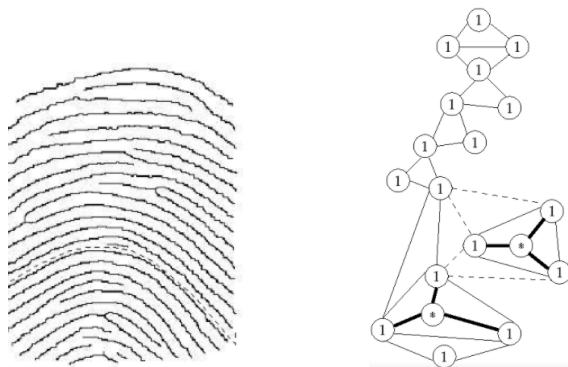
navazuje Chikkerur [7], který definuje tzv. K -plet reprezentaci. Na rozdíl od předchozí metody vytváří okolí bodů z pevně daného množství sousedů pomocí algoritmu K -nejbližších sousedů. Tyto body mají relativní souřadnice vůči danému markantu. Použita je upravená verze algoritmu, která vyžaduje vybrání bodu z každého kvadrantu v relativním souřadném systému bodu. Tato úprava zaručuje lepší konektivitu jednotlivých markantů. Práci ještě vylepšuje Chen [37], ke každému markantu používá *Supplementary associative point (SAP)*. *SAP* je bod vytvořený ve směru orientace markantu. Výsledná K -plet (*ImKplet*) reprezentace obsahuje markanty i *SAP* body.

Papilární linie

Hrany grafu otisku prstu jde určit přímo z papilárních linií. Takovou grafovou reprezentaci použil například Horadam [16]. Hrana mezi dvěma vrcholy existuje, pokud je mezi markanty papilární linie. Cesta v grafu mezi dvěma vrcholy existuje, pokud markanty leží na společné papilární linii. Mohou zde vznikat izolované body v grafu, jelikož se na snímku otisku často nenachází celá papilární linie. Otisk obsahující izolované body lze vidět na obrázku 2.11. Pro porovnání dvou otisků (grafů) je použita vzdálenost mezi grafy definována rovnicí

$$d(g, g') = 1 - \frac{s(g, g')}{\sqrt{|g||g'|}}, \quad (2.3)$$

kde g a g' jsou porovnávané grafy, $|g|$ je počet hran a vrcholů grafu g a $s(g, g')$ je počet přiřazených hran a vrcholů. Přiřazení je určeno na základě euklidovské vzdálenosti.



Obrázek 2.12: Příklad reprezentace otisku prstů topologií papilárních linií. [25]

Topologie papilárních linií

Při porovnávání otisků pomocí markantů je rozhodnuto, že jsou dva otisky stejné, pokud se shodují v určitém počtu markantů. Porovnání však nebere v potaz pozici markantů na papilárních liniích. Dva otisky mohou mít podobné markanty, ale topologie papilárních linií bude odlišná [25].

Reprezentaci v podobě topologie používá Ölz [25]. Prvním krokem zpracování je vytvoření mřížkového grafu (*grid graph*) z pixelů zpracovaného otisku prstu. Každý vrchol grafu obsahuje informaci, jestli pixel obsahuje papilární linii, větvení nebo jestli je pixel prázdný. Každý uzel je propojen se svými sousedními pixely, graf tvoří mřížku. Hrany obsahují informaci, jestli se vrcholy nacházejí na stejné linii, na nejbližší cestě k linii, nebo nejsou propojeny vůbec. Tento graf je poté zjednodušován pomocí kontrakcí. Operace kontrakce sloučí vrcholy podle daných pravidel. Výsledný graf již neobsahuje vrcholy, které nereprezentují papilární linii, ale pouze topologii papilárních linií.

Graf papilárních linií

Markanty otisků prstů jsou silně spjaty s jejich polohou v souřadném systému. Otisknutý prst však může být posunutý, orotovaný a může mít rozdílnou velikost. Všechny tyto změny způsobí změnu souřadnic markantů. Reprezentace grafu otisku prstů pomocí grafu papilárních [17] je invariantní vůči transformacím.

Z předzpracovaného snímku kostry otisku jsou extrahovány jednotlivé papilární linie. Každá papilární linie je určena svojí délkou, dvěma koncovými body a dvěma stranami. Sousedství dvou linií může být koncové (linie sdílejí bod větvení), nebo boční (kolmice z bodu jedné linie do bodu druhé neprotíná žádnou jinou linii). Ke každé linii je ještě navíc určena její úroveň. Vrcholy grafu reprezentují zpracované papilární linie. Hrany grafu reprezentuje sousedství linií. Porovnání grafů je provedeno pomocí rozdílu délky reprezentovaných linií a počtu sousedních linií na korespondujících stranách.

Kapitola 3

Grafové neuronové sítě

Hluboké učení je disciplína strojového učení, která posledních pár let nabývá na popularitě. To je způsobeno zejména zvýšením výpočetního výkonu a větším množstvím dat. Neuronové sítě obvykle pracují s daty euklidovského prostoru, který lze obecně definovat jako \mathcal{R}^n . Například obrázky je možné vyjádřit v lineárním n -dimenzionálním prostoru, kde souřadnice x, y určují polohu pixelu a z barvu. Data ale mohou pocházet z neeuklidovského prostoru, například grafy. Struktura grafu se dá zobrazit do euklidovského prostoru, nicméně toto zobrazení nelze použít k vyjádření atributů jednotlivých vrcholů a hran nebo k vyjádření směru hran.

Grafy vyjadřují množinu objektů a jejich vzájemné propojení. Grafem lze vyjádřit složení molekul v chemii nebo třeba propojení uživatelů na sociálních sítích [36]. Tento typ dat je velmi obtížné zpracovat. Grafy mohou obsahovat různou strukturu, různý počet vrcholů a hran.

Ke zpracování grafů nelze použít běžné metody hlubokého učení, neplatí totiž předpoklad, že jednotlivé instance dat jsou vzájemně nezávislé. Každá instance (vrchol) grafu je závislá na ostatních instancích, které jsou s ní spojeny hranou [36]. Síť zpracovávající grafy se nazývají grafové neuronové sítě (GNN). Grafové neuronové sítě můžeme rozdělit například na čtyři kategorie:

- Rekurentní grafové neuronové sítě (*RecGNN*)
- Konvoluční grafové neuronové sítě (*ConvGNN*)
- Grafové autoenkodéry (*GAE*)
- Časoprostorové grafové neuronové sítě (*STGNN*)

Prvními vzniklými modely grafových neuronových sítí jsou rekurentní grafové neuronové sítě [36]. Principem učení je neustálé předávání informací (zpráv) mezi vrcholem a jeho sousedy, dokud se celý systém neustálí.

Konvoluční grafové neuronové sítě vychází z rekurentních. Generalizují konvoluci z klasických konvolučních neuronových sítí (CNN) na grafová data. Grafovou konvoluci lze chápat jako předávání zpráv mezi vrcholy. Na rozdíl od rekurentních GNN uvažují více vrstev grafových konvolucí [10]. *ConvGNN* hrají zásadní roli při vytváření složitějších GNN modelů.

Grafové autoenkodéry jsou modely využívající učení bez učitele. Autoenkodér převede vstupní graf do vektorového prostoru a z něho následně rekonstruuje výsledný graf. *GAE* se používá například pro generování grafových dat.

Časoprostorové grafové neuronové sítě zpracovávají časoprostorové grafy, tedy grafy znázorňující pohyb tělesa v čase. *STGNN* uvažuje závislost prostoru současně se závislostí času. Časoprostorové modely mohou být využity v dopravě například k předpovědi rychlosti dopravy nebo předpovědi chování řidičů.

Tato kapitola se zaměří na základy grafových neuronových sítí – reprezentaci dat, použité algoritmy a různé typy úloh, které se dají řešit pomocí GNN. Dále se zaměří na konvoluční grafové sítě a existující metody, které mohou být využity právě k verifikaci otisků prstů.

3.1 Vstupní graf

Modely strojového učení typicky požadují vstup ve formě vektoru nebo matice, obecně tensoru. Graf můžeme definovat vrcholy, hranami, globálním kontextem a konektivitou [31]. První tři typy informací lze vyjádřit jednoduše jako tensor. Reprezentace konektivity je komplikovanější.

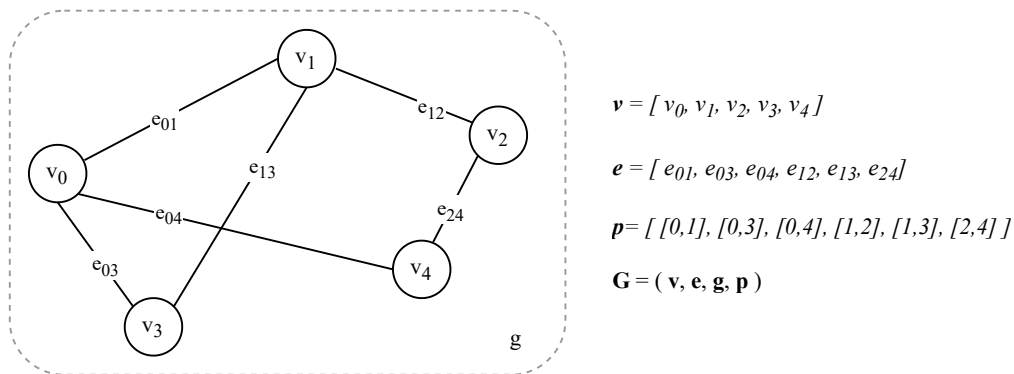
Reprezentace vrcholů i hran je definována tensorem. Vstupní graf je definován tensorem vrcholů o velikosti $[n, d]$, kde n je počet vrcholů a d jsou dimenze reprezentace vrcholu. Vrchol může být reprezentován i skalárem, v tom případě by velikost tensoru vrcholů byla pouze $[n]$. Podobně jsou reprezentovány i hrany. Je třeba zmínit, že reprezentace hran obsahuje pouze atributy jednotlivých hran, a ne vrcholy, které hrana propojuje. Tuto informaci nese konektivita.

Globální kontext grafu obsahuje informaci o celém grafu. Stejně jak u vrcholů a hran je definována tensorem. Kontext může být reprezentován pouze skalární hodnotou, například třída grafu.

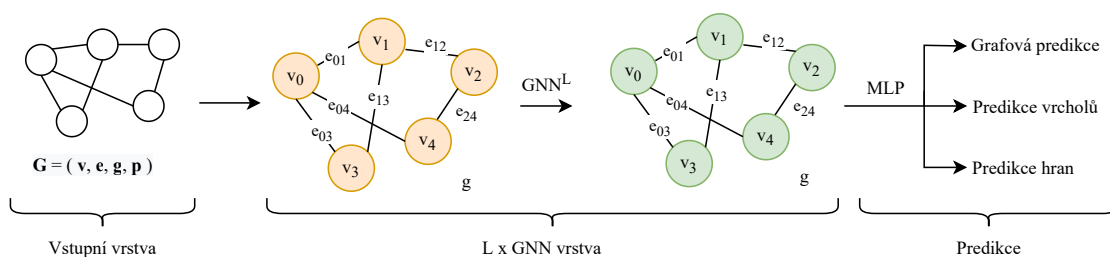
Reprezentace konektivity

Konektivitu lze reprezentovat maticí sousednosti. Matice sousednosti je čtvercová matice o velikosti $[n, n]$, kde n je počet vrcholů. Hodnota matice a_{ij} odpovídá informaci, jestli existuje hrana z i -tého vrcholu do j -tého vrcholu. Nevýhoda matice sousednosti spočívá v efektivitě uložení informace. Velké množství vrcholů zvyšuje paměťovou náročnost a také obvykle vede k velmi řídkým maticím, které nejsou paměťově efektivní. Dalším problémem je existence mnoha maticí sousednosti, které reprezentují stejnou konektivitu. Matice sousednosti nejsou invariantní vůči permutaci, tudíž různé reprezentace stejné konektivity nemusí produkovat stejný výstup neuronové sítě.

Vhodnější způsob reprezentace konektivity je pole sousednosti. Konektivita hrany e_k mezi vrcholy n_i a n_j je popsána jako dvojice (i, j) v k -tém prvku pole sousednosti. Velikost pole je $[e, 2]$, kde e je počet hran.



Obrázek 3.1: Příklad reprezentace vstupního grafu grafové neuronové sítě, v je tensor vrcholů, e tensor hran, p je konektivita reprezentovaná polem sousednosti a g je globální kontext grafu.



Obrázek 3.2: Architektura obecné grafové neuronové sítě.

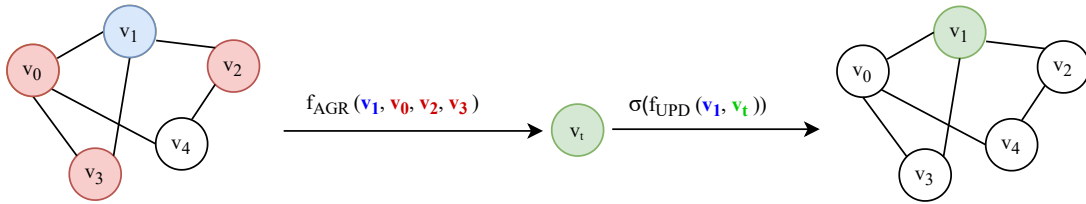
3.2 Architektura

Většinou lze architekturu grafové neuronové sítě rozdělit na vstupní vrstvu, GNN vrstvu a vrstvu predikce. Vstupní vrstva načte graf, konkrétně atributy vrcholů, hran, konektivitu a globální kontext. Takto reprezentovaný graf vstupuje do GNN vrstvy. GNN vrstva upraví atributy grafu, ale struktura grafu zůstane stejná [21]. Informace se šíří pomocí předávání zpráv mezi vrcholy a hranami. Více GNN vrstev za sebou zvyšuje stupeň agregace v grafu, hodnoty jednotlivých atributů jsou pak závislé na větší oblasti v grafu. Poslední vrstvou je vrstva predikce. Vrstva predikce závisí na problému, který neuronová síť řeší, ten určuje kde a jak získat informace ze zpracovaného grafu. Architekturu popisuje obrázek 3.2.

GNN vrstva

Každý vrchol a každá hrana obsahuje ve vstupní vrstvě svoje vlastní atributy. Tyto atributy nejsou ovlivněny strukturou grafu. Pokud bychom provedli predikci bez zpracování vstupního grafu, tak by výstup sítě nebyl ovlivněn konektivitou grafu. Můžeme využít GNN vrstvu, která provede agregaci a zajistí tak povědomí vrcholů nebo hran o struktuře grafu. Proces agregace může být definován pomocí algoritmu předávání zpráv [21, 15]. Algoritmus předávání zpráv mezi vrcholy pracuje ve třech krocích:

1. Pro každý vrchol v grafu, shromáždí všechny atributy sousedících vrcholů (zprávy).
2. Zpracuje všechny shromážděné zprávy pomocí agregační funkce.
3. Všechny zpracované zprávy předloží na vstup aktualizací funkce.



Obrázek 3.3: Příklad algoritmu předávání zpráv pro jeden vrchol, v_0 až v_4 jsou vrcholy grafu, v_t je agregovaná zpráva a f_{AGR} , σ , f_{UPD} jsou po řadě agregační, aktivační a aktualizací funkce.

Agregační funkce může být definována například jako maximum, minimum nebo průměr zpráv. Jedinou podmínkou je aby funkce byla invariantní vůči pořadí zpráv. Výstupem agregace každého vrcholu je jedna zpráva. Aktualizační (kombinační) funkce zkombinuje vygenerovanou zprávu s aktuálními parametry vrcholu. Kombinaci lze implementovat jako konkatenaci, součet, nebo průměr jednotlivých atributů. Zkombinovaná zpráva je ještě předložena na vstup aktivační funkce, kterou může být funkce *ReLU*. Podobně se tento postup uplatní i na hrany. Předávání zpráv může docházet mezi vrcholem a sousedními hranami nebo mezi hranou a sousedními vrcholy. Záleží na podobě řešeného problému a návrhu grafové neuronové sítě.

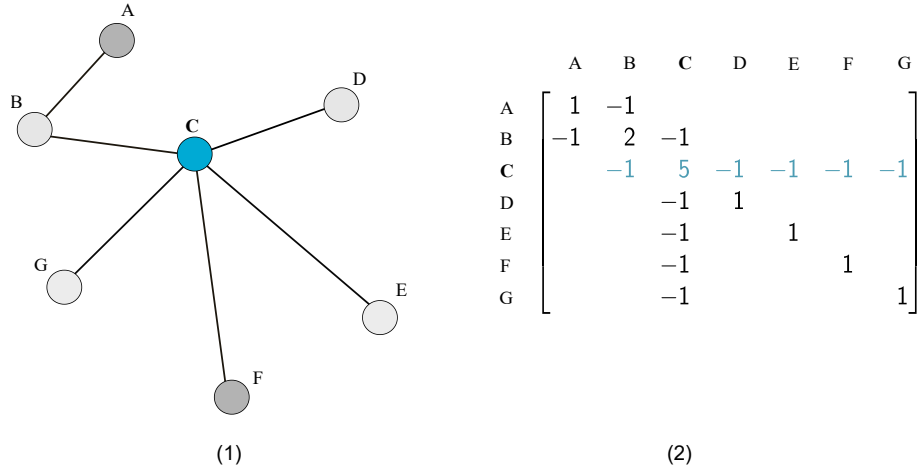
Predikce

Graf může obsahovat atributy vrcholů, atributy hran i atributy celého grafu. Každá kombinace těchto atributů může být použita pro predikci. Obecně lze typ predikce nad grafy rozdělit na tři typy – predikce na grafové úrovni, na úrovni vrcholů a na úrovni hran [31].

Predikce na grafové úrovni spočívá v predikci jedné vlastnosti celého grafu. Prvním krokem je agregace atributů ze všech vrcholů nebo hran pomocí *Global pooling* vrstvy. Tato vrstva je podobná *Global pooling* vrstvám v konvolučních neuronových sítích. Pro agregaci se používá součet, průměr nebo maximum ze všech atributů grafu. Výstup agregace je poté zpracován neuronovou sítí, která poskytne odpovídající predikci. Alternativním způsobem predikce vlastnosti celého grafu může být definování tzv. *Master* vrcholu, tento vrchol je přidán do grafu uměle a sousedí se všemi vrcholy, zajistí tak lepší konektivitu celého grafu. Následně pak můžeme předložit na vstup neuronové sítě přímo atributy *Master* vrcholu.

Predikce na úrovni vrcholu znamená predikovat vlastnost pro každý vrchol v grafu. V případě, že máme k dispozici atributy pro každý vrchol, je postup získání predikce přímočarý. Atributy vrcholu položíme na vstup neuronové sítě, která predikuje vlastnosti daného vrcholu. Neuronová síť takto zpracuje všechny vrcholy. V případě že vrcholy neobsahují žádné atributy a informace je uložena pouze v attributech hran, je nutné provést agregaci. Pro každý vrchol jsou agregovány atributy ze sousedních hran. Atributy jsou poté zpracovány neuronovou sítí.

Analogicky probíhá i predikce na úrovni hran. Predikci atributů hran lze použít například v případě predikce konektivity, kdy neznáme vztahy mezi jednotlivými vrcholy. V tomto případě je vstupní graf plně propojený a na základě predikovaných atributů hran je vyvozena závislost mezi vrcholy.



Obrázek 3.4: Tvorba Laplaceovy matice, na levém obrázku (1) je příklad vstupního grafu a na pravém jeho Laplaceova matice (2) [9].

3.3 Konvoluční GNN

Konvoluční grafové neuronové sítě (*ConvGNN*) vycházejí z rekurentních grafových sítí. Rekurentní grafové neuronové sítě používají stále jednu vrstvu pro aktualizaci parametrů grafu. *ConvGNN* definují fixní počet vrstev s rozdílnými vahami [36]. Tento typ grafových sítí lze rozdělit na spektrálně založené a prostorově založené. Dále se zaměříme na spektrálně založené metody.

Spektrálně založené metody definují grafovou konvoluci jako filtry z pohledu zpracování grafového signálu. Z tohoto pohledu slouží operace konvoluce k odstranění šumu ze signálu [36]. Metody počítají s neorientovaným grafem, který je reprezentovaný jako normalizovaná Laplaceova matice [9]. Laplaceova matice lze spočítat jako $L = D - A$, kde $D_v = \sum_u A_{vu}$ pro vrchol v , A je matice sousednosti grafu. Příklad tvorby Laplaceovy matice lze vidět na obrázku 3.4. Tato matice je symetrická a semidefinitní, lze ji vyjádřit jako $L = U\Lambda U^T$, kde $U = [u_0, u_1, \dots, u_{n-1}] \in \mathbb{R}^{n \times n}$ je matice vlastních vektorů a Λ diagonální matice vlastních čísel. Grafový signál je vektor příznaků všech vrcholů grafu a je vyjádřen jako $x \in \mathbb{R}^n$. Pomocí grafové Fourierovy transformace lze promítnout vstupní grafový signál do ortonormálního prostoru s bází tvořenou normalizovanou Laplaceovou maticí. Grafovou konvoluci grafového signálu x filtrem $g \in \mathbb{R}^n$ lze definovat rovnicí

$$x *_G g = \mathcal{F}^{-1}(\mathcal{F}(x) \odot \mathcal{F}(g)) = U(U^T x \odot U^T g), \quad (3.1)$$

kde $\mathcal{F}, \mathcal{F}^{-1}$ značí grafovou Fourierovu transformaci a \odot operaci *element-wise* násobení. Spektrálně založené metody čelí třem problémům. Jakékoliv narušení grafové struktury způsobí změnu ve vlastní bázi Laplaceovy matice grafu. Druhý problém je ten, že naučené filtry jsou závislé na struktuře daného grafu. Třetím problémem je složitost rozkladu matice na vlastní vektory $O(n^3)$. Tyto problémy částečně řeší metody ChebNet [10] a GCN [20].

ChebNet

ChebNet [10] reprezentuje filtry grafové konvoluce jako polynomy, které lze definovat rovnicí

$$p_w(L) = \sum_{i=1}^d w_i T_i(\tilde{L}), \quad (3.2)$$

kde d značí stupeň lokalizace, w_i váhu filtru, L je Laplaceova matice, \tilde{L} je normalizovaná matice pomocí největšího vlastního čísla L , $\tilde{L} = \frac{2L}{\lambda_{max}(L)} - I_n$ a T_i je Chebyshevův polynom i -tého stupně prvního druhu. Takto definované polynomiální filtry jsou invariantní vůči seřazení vrcholů. Tuto vlastnost lze vidět při prvním stupni polynomu p_w , kde jsou příznaky vrcholu agregovány sčítáním příznaků jeho sousedů. Podobný důkaz následuje i pro polynomy vyššího stupně.

Grafovou neuronovou síť můžeme složit z několika vrstev grafové konvoluce (polynomiální filtr) a aktivační funkce, podobně jako v standardní konvoluční neuronové síti (CNN). Při použití K různých vrstev filtrů, kde k -tá vrstva obsahuje váhy $w^{(k)}$ k učení, můžeme průchod vzorku x sítí definovat následujícími rovnicemi

$$h^{(0)} = x \quad (3.3)$$

$$p^{(k)} = p_{w^{(k)}}(L) \quad (3.4)$$

$$g^{(k)} = p^{(k)} \times h^{(k-1)} \quad (3.5)$$

$$h^{(k)} = \sigma(g^{(k)}). \quad (3.6)$$

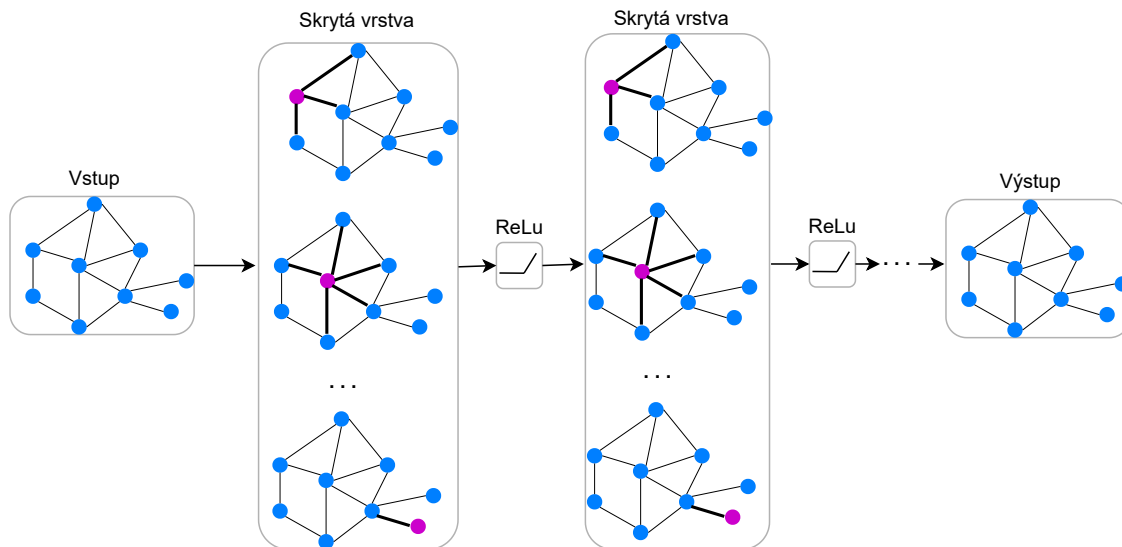
Vstupní příznak vrcholu $h^{(0)}$ je přiložen na vstup první vrstvy a postupně projde všemi K vrstvami, jednotlivé vrstvy jsou značeny $k = 1, 2, \dots, K$. V každé vrstvě je spočtena matice $p^{(k)}$ jako Laplaceova matice L vyhodnocená polynomem, který je definovaný vahami filtru $w^{(k)}$ (3.4). Matice $p^{(k)}$ je násobena výstupním vektorem příznaků z předchozí vrstvy $k - 1$ označen $h^{(k-1)}$ (3.5), poté je aplikována nelineární aktivační funkce σ (3.6). Grafová neuronová síť používá stejné váhy filtru, zajišťuje tak sdílení vah mezi různými vrcholy.

GCN

Grafová konvoluční neuronová síť GCN [20] je jeden z nejpoužívanějších modelů GNN. Narozdíl od modelu ChebNet přidává vazbu vrcholu sám na sebe explicitním přidáním zpětné hrany. Využívá Kipf normalizaci agregace na základě stupně použitých vrcholů. Model by měl snížit váhu zpráv sousedícím vrcholům s vyšším stupněm, tedy vrcholům obsahující velký počet sousedů. Příklad vícevrstvé GCN lze vidět na obrázku 3.5. Kipf normalizace je popsána rovnicí

$$h_v^{(k)} = \sigma \left(W^{(k)} \cdot \sum_{u \in \mathcal{N}(v)} \frac{h_u^{(k-1)}}{\sqrt{|\mathcal{N}(u)| |\mathcal{N}(v)|}} + B^{(k)} \cdot h_v^{(k-1)} \right) \quad \text{pro } \forall v \in V, \quad (3.7)$$

kde $h_v^{(k)}$ je vektor příznaků vrcholu v ve vrstvě k , σ je aktivační funkce (ReLU), $\mathcal{N}(v)$ množina sousedních vrcholů vrcholu v a $W^{(k)}, B^{(k)}$ jsou matice bias a matice vah vrstvy k .



Obrázek 3.5: Vícevrstvá grafová konvoluční síť GCN s filtry prvního řádu [20].

SimGNN

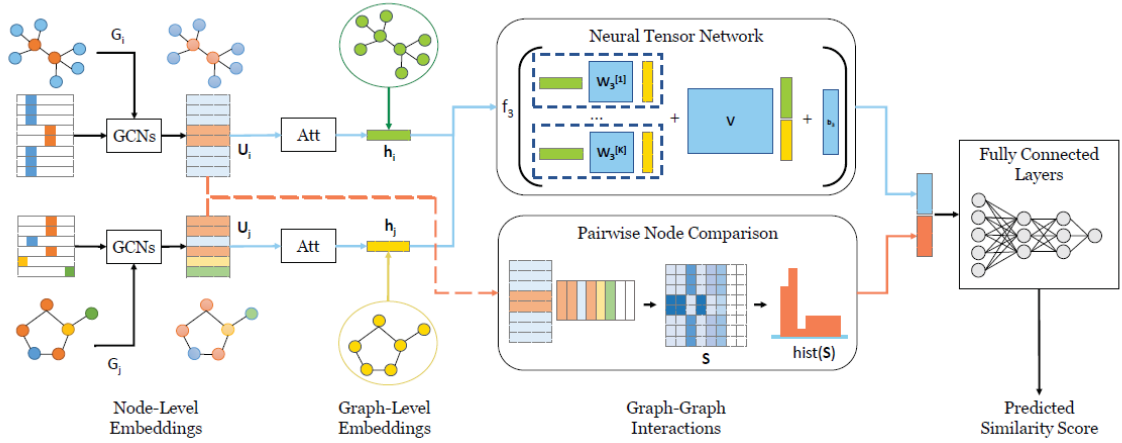
Na velký úspěch grafových neuronových sítí v oblasti klasifikace vrcholů i celých grafů navázala metoda SimGNN [3], která se zaměřuje na problém podobnosti grafů. Grafová podobnost je jedním z nejdůležitějších problémů grafových aplikací. Problém spočívá v nalezení grafové editační vzdálenosti (*GED*) nebo maximálního společného podgrafu (*MCS*). Získání podobnosti grafů je velmi výpočetně náročné.

SimGNN mapuje pár grafů na skóre podobnosti. Nejdříve model transformuje vrcholy každého grafu na vektor, který popisuje strukturální vlastnosti každého vrcholu. Podobnost grafu je získána dvěma strategiemi. Na základě interakce na grafové úrovni nebo porovnání vrcholů. Výstup z obou strategií je zkombinován do výsledného skóre. Architektura modelu SimGNN je znázorněna na obrázku 3.6.

První strategie interakce na grafové úrovni využívá grafovou konvoluční neuronovou síť GCN. Vstupní graf je zpracován několika vrstvami GCN. Vektory vrcholů jsou předány modulu pozornosti (*Attention module*). Modul pozornosti spočte globální kontext grafu a z něho váhu pro každý vrchol. Vrcholům podobným globálnímu kontextu je přidána větší váha. Z vah vrcholů je provedena agregace váženým součtem do jednoho vektoru popisujícího graf. Tento postup je aplikován pro oba vstupní grafy. Vektory obou grafů jsou zpracovány pomocí modelu neuronových tenzorových sítí (NTN) [32], jehož výstupem je určený počet hodnot skóre. Vektor skóre je zpracován vícevrstvou neuronovou sítí, která zredukuje vektor pouze na jedinou hodnotu skóre podobnosti.

Problémem první strategie je ztráta informace o jednotlivých vrcholech, rozložení jednotlivých příznaků vrcholů a informace o jejich počtu. Problém řeší strategie porovnání vrcholů. Využívá výstup z grafových konvolučních sítí GCN z první strategie. Vektory grafu s menším počtem vrcholů jsou doplněny nulovými vektory, aby počet vektorů obou grafů byl stejný. Poté je proveden výpočet podobnosti pro každý pár vrcholů. Vznikne tak matice $S^{n \times n}$, kde n je počet vrcholů většího grafu. Z matice je extrahován histogram: $hist(S) \in R^b$, kde b je parametr určující počet binů v histogramu.

Samotné porovnání vrcholů nestačí pro natrénování modelu, proto se používá pouze v kombinaci s neuronovými tenzorovými sítěmi (NTN). Výstupy z obou strategií jsou zkom-



Obrázek 3.6: Přehled architektury modelu SimGNN [3]. První strategie (na grafové úrovni) je znázorněna modrou šipkou. Druhá strategie (porovnání vrcholů) je znázorněna červenou šipkou.

binovány a zpracovány plně propojenou neuronovou sítí, která predikuje finální skóre podobnosti.

SplineCNN

S cílem generalizace tradiční konvoluce konvolučních neuronových sítí vznikla metoda SplineCNN [14]. Práce zavádí nový konvoluční operátor založený na B–splinech. Na rozdíl od příbuzných přístupů, které filtrují ve spektrální doméně, agreguje metoda příznaky pouze v prostorové doméně.

Funkce konvolučního jádra B–spline báze je parametrizována konstantním počtem trénovatelných parametrů. Parametry lze chápat jako řídicí hodnoty určující výšku B–spline povrchu. Konvoluční operátor agreguje příznaky vrcholů v lokálním sousedství vrcholu vážených trénovatelnou spojitou funkcí jádra. Operátor lze definovat rovnicí

$$\mathbf{x}'_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{x}_j \cdot h_{\Theta}(\mathbf{e}_{i,j}), \quad (3.8)$$

kde $\mathcal{N}(i)$ určuje sousední vrcholy vrcholu i , x_j atributy vrcholu j a h_{Θ} funkci jádra definovanou nad bází B–spline.

3.4 Problém kvadratického přiřazování

V podkapitole 2.4 byly definovány různé grafové reprezentace otisku prstu. Pro účel verifikace dvou otisků je nutné tyto grafy zarovnat. Zarovnání dvou grafů spočívá v přiřazení vrcholů jednoho grafu ke druhému, tento problém lze definovat jako *Quadratic Assignment Problem* (QAP). QAP je problém kombinační optimalizace. Cílem je přiřadit n různých zařízení n lokacím, tak aby se minimalizovala hodnota všech přiřazení a ke každému zařízení byla přidělena právě jedna lokace a každé lokaci právě jedno zařízení. Jedná se o jeden z nejnáročnějších NP – úplných problémů, neexistuje známý algoritmus pro vyřešení tohoto problému v polynomiálním čase. Problém lze řešit pomocí programu pro řešení kombinatorických problémů (*Combinatorial solver*), například pomocí Lagrangeovy dekompozice. Metody používající grafové neuronové sítě pro řešení problému přiřazení grafů lze rozdělit podle způsobu zpracování vstupních grafů. První způsob spočívá ve zpracování vstupních grafů zvlášť, přičemž z každého grafu extrahuje příznaky a ty poté zkombinuje a dál zpracuje. Druhý způsob vstupní grafy zkombinuje do jednoho grafu, ve kterém každý vrchol reprezentuje přiřazení dvou vrcholů ze vstupních grafů. Řešení problému je poté určeno zpracováním tohoto grafu.

Kombinace vstupních grafů

Kombinaci vstupních grafů využívá například práce Wang [35]. Problém přiřazení je transformován na predikci vrcholů grafu přiřazení (assignment graph). Architektura se skládá z vytvoření grafu přiřazení, enkodéru, dekodéru a konvolučních bloků.

Graf přiřazení $\mathcal{G}^A = (v^A, e^A)$ vznikne kombinací grafů $\mathcal{G}^{(1)} = (v^{(1)}, e^{(1)})$ a $\mathcal{G}^{(2)} = (v^{(2)}, e^{(2)})$, vrcholem grafu je korespondence $(v_i^{(1)}, v_a^{(2)}) \in v^{(1)} \times v^{(2)}$ vyjádřená jako $v_{ia}^A \in v^A$. Hrana se nachází mezi dvojicí vrcholů $v_{ia}^A, v_{jb}^A \in v^A$ právě tehdy, když existují dvě hrany $(v_i^{(1)}, v_j^{(1)}) \in e^{(1)}$ a $(v_a^{(2)}, v_b^{(2)}) \in e^{(2)}$. Atributy vrcholů i hran jsou vytvořeny kombinací atributů ze vstupních grafů. Příklad tvorby grafu je zobrazen na obrázku 3.7.

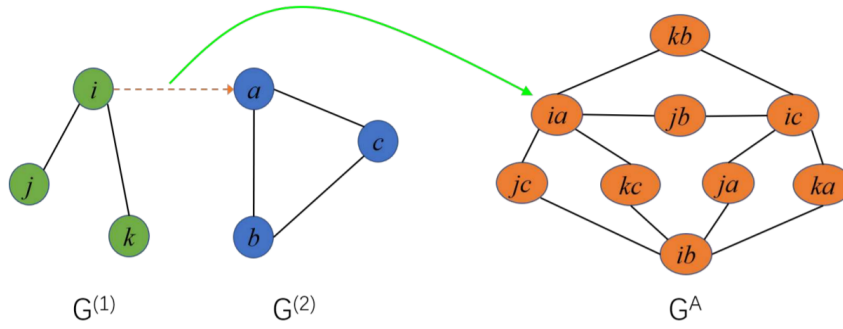
Vytvořený graf přiřazení je přiložen na vstup enkodéru ϕ_{enc}^v a ϕ_{enc}^e . Enkodér je definovaný pomocí dvou MLP (*Multi-layer perceptron*), zvlášť pro vrcholy a hrany. Zpracovaný graf je předán na vstup prvního konvolučního bloku.

Konvoluční blok je tvořen vrstvou konvoluce hran a vrstvou konvoluce vrcholů. Hranová konvoluce agreguje informaci z obou asociovaných vrcholů do korespondující hrany a aktualizuje její atributy. Následuje konvoluce vrcholů, která pro každý vrchol agreguje informaci ze všech sousedních hran. V modelu se vyskytuje několik na sebe navazujících konvolučních bloků.

Na závěr je výstup konvolučních bloků zpracován dekodérem vrcholů ϕ_{dec}^v , ten je definován stejně jak enkodér, pomocí MLP. Každý vrchol obsahuje predikované příznaky. Predikce vrcholu v_{ia}^A reprezentuje skóre korespondujícího přiřazení vrcholu $v_i^{(1)}$ a $v_a^{(2)}$. Pro trénování modelu je použita *cross entropy loss*, definována rovnicí

$$L = - \sum_{i=1}^{n^2} \left(Y_{i,1}^{gt} \log Y_{i,1} + Y_{i,2}^{gt} \log Y_{i,2} \right), \quad (3.9)$$

kde $Y^{gt} \in \{0, 1\}^{n^2 \times 2}$ je vzorové přiřazení z datové sady a Y jsou predikované skóre vrcholů.



Obrázek 3.7: Tvorba grafu přiřazení [35].

Zpracování grafů samostatně

Zpracování grafu přiřazení může být pro větší grafy pomalé, proto může být výhodnější zpracovat grafy samostatně. Tento způsob je velmi podobný architektuře siamským neuronových sítí. Z každého grafu jsou pomocí grafové neuronové sítě extrahovány příznaky a ty následně porovnány. Tento přístup využívá *Deep Graph Matching* [30]. Tato metoda kombinuje použití grafové neuronové sítě a existujícího *state-of-the-art* matematického solveru, který řeší problém grafového přiřazení. Tato architektura přistupuje k solveru jako k černé skřínce a je plně diferenciovatelná. Cílem této práce bylo nalézt korespondenci mezi dvěma obrazy objektu, tedy nalézt v každém snímku klíčové body a jejich korespondence napříč snímky.

Vstupní obrázky jsou zpracovány pomocí předtrénované konvoluční neuronové sítě *VGG-16*. Výstupem sítě jsou klíčové body s příznakovými vektory. Navíc je pomocí *max-pooling* vrstvy extrahován globální vektor příznaků. Klíčové body jsou poté použity při tvorbě grafů.

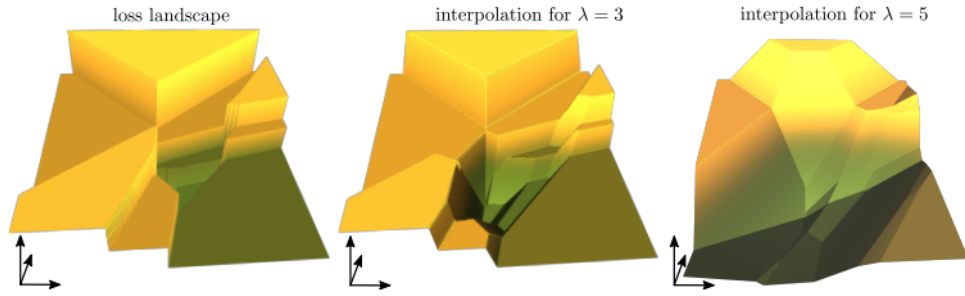
Z klíčových bodů extrahovaných z *VGG-16* je vytvořena grafová reprezentace. Nad klíčovými body je provedena Delaunayho triangulace, která spojí klíčové body hranami. Vrcholy obsahují příznaky extrahované z neuronové sítě. Atributy hrany jsou složeny z atributů vrcholů mezi kterými se hrana nachází jako normalizované relativní souřadnice obou vrcholů.

Graf je zpracován dvouvrstvou grafovou konvoluční neuronovou sítí *SplineCNN* [14] s agregační funkcí maximum. Atributy vrcholů zpracovaného grafu jsou poté sečteny s originálními atributy původního grafu, tedy s vektory příznaků z prvního kroku. Atributy hran jsou nastaveny na rozdíl atributů vrcholů zpracovaného grafu.

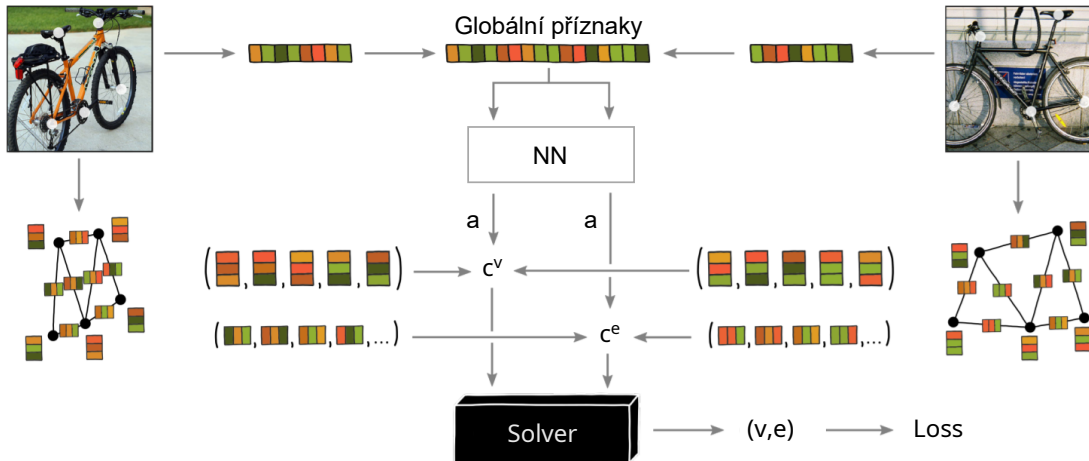
Následně je spočtena kombinatorická instance v podobě (c^v, c^e) , kde $c^v \in \mathcal{R}^{|V_1||V_2|}$ a $c^e \in \mathcal{R}^{|E_1||E_2|}$ jsou cenové vektory vrcholů a hran. Proces tvorby instance je znázorněn na obrázku 3.9. Výpočet cenového vektory definuje rovnice

$$c_{i,j}^v = \sum_k f_s^v(i)_k a_k f_t^v(j)_k, \quad (3.10)$$

kde $f_s^v(i)$ je vektor příznaků i -tého vrcholu vstupního grafu s , hodnota a vznikla zpracováním sjednoceného globálního vektoru příznaků obou snímků pomocí jednovrstvé neuronové sítě. Analogicky je vypočítán cenový vektor hran c^e .



Obrázek 3.8: Znázornění diferenciací účelové funkce s využitím interpolace [30].



Obrázek 3.9: Konstrukce kombinatorické instance pro přiřazení bodů pomocí solveru [30].

Posledním krokem metody je přiložení instance (c^v, c^e) na vstup kombinatorického solveru, který řeší problém grafového přiřazení. Solver je založený na duálním blokovém souřadnicovém vzestupu pro Lagrangeovy rozklady. Autoři k němu přistupují jako k černé skřínce, která mapuje spojitý vstup ve formě instance na diskretní výstup indikující optimální přiřazení. Mapování je po částech konstantní. Pro diferenciaci tak není možné použít optimalizaci založenou na gradientu, jako se například používá v neuronových sítích, ale je zde použita implicitní interpolace. Vliv interpolace znázorňuje obrázek 3.8.

Výstupem solveru jsou vektory v, e . Vektor $v \in \{0, 1\}^{\{|V_1|+|V_2|\}}$, kde $v_{i,j} = 1$ pokud vrchol $i \in V_1$ je přiřazen k vrcholu $j \in V_2$, jinak $v_{i,j} = 0$. Analogicky je definován vektor $e \in \{0, 1\}^{\{|E_1|+|E_2|\}}$. Účelová funkce L je pak definována jako Hammingova vzdálenost mezi navrženým přiřazením v a správným přiřazením v^* , které je dostupné z datové sady

$$L(v) = v \cdot (1 - v^*) + v^* \cdot (1 - v). \quad (3.11)$$

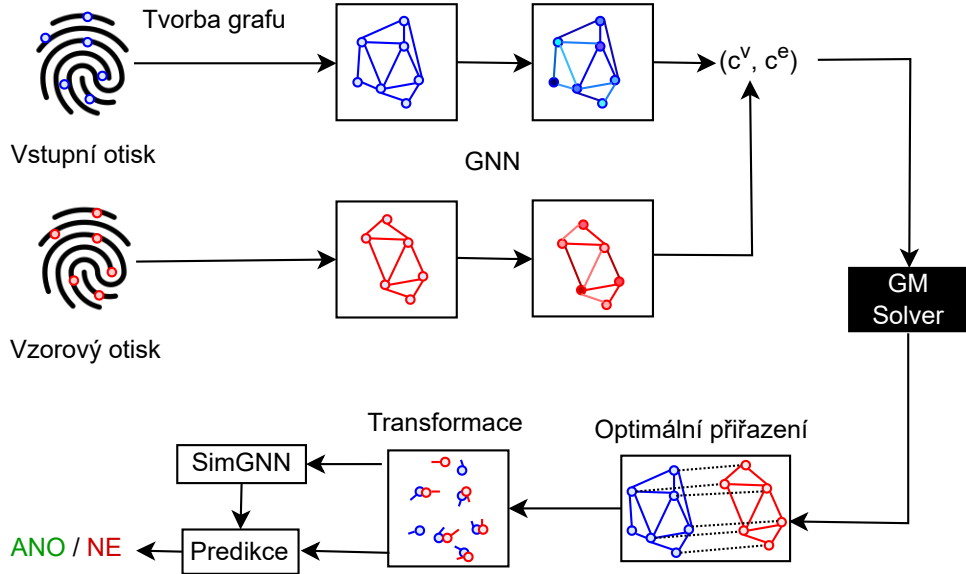
Kapitola 4

Navrhované řešení

Řešení počítá se zadanými vstupními otisky ve formě markantů, jelikož markanty jsou nej-používanějším způsobem reprezentace otisků a v tomto formátu je zadána i poskytnutá datová sada. Každý markant obsahuje souřadnice, typ a úhel. Prvním problémem, který je potřeba vyřešit je tvorba grafové reprezentace. Jednotlivé markanty otisku reprezentují vrcholy grafu, vrchol obsahuje normalizované atributy jako jednotlivé vlastnosti markantu. Získání jednotlivých hran grafu probíhá pomocí Delaunayho triangulace nebo pomocí metody K – nejbližších sousedů.

Existující metody verifikace otisků prstů s markanty jsou Ratha [28] a Hong [19]. Obě tyto metody pracují ve dvou krocích – zarovnání otisků a porovnání. Navržené řešení zarovnání otisků spočívá v grafové reprezentaci otisku prstů a vyřešení problému grafového přiřazení. Problém grafového přiřazení řeší existující metody Wang [35] a *Deep Graph Matching* [30]. Metoda Wang není pro zarovnání otisků vhodná, jelikož vyžaduje tvorbu kombinovaného grafu a vzhledem k počtu markantů otisků by tento graf byl příliš složitý. V navrhovaném řešení je tedy využita metoda *Deep Graph Matching*. Původní metoda předpokládá na vstupu obrázek a z něho získává klíčové body. V případě otisků prstů lze tento krok přeskočit, klíčové body jsou již k dispozici. Výstupem této metody je matice reprezentující optimální přiřazení vrcholů v rámci páru grafů. Z optimálního přiřazení jsou vypočteny parametry afinních transformací – změna měřítka, translace a rotace. Výpočet lze realizovat například pomocí metody nejmenších čtverců, nebo pomocí algoritmu RANSAC. Výsledná transformace je použita na zarovnání jednoho otisku na druhý.

Posledním krokem je fáze porovnání otisků (verifikace). Pro verifikaci lze použít již získaná transformace, která byla odhadnuta z optimálního přiřazení. Ze zarovnaných otisků prstů je k dispozici informace o vzdálenosti jednotlivých zobrazených markantů, rozdílu jejich úhlů nebo shoda jejich typu. Tyto metriky používá pro porovnání markantů i metoda Ratha [28]. Alternativně lze použít na zarovnané otisky model SimGNN [3]. Model byl zvolen z důvodu získání dodatečné informace o zarovnaných otiscích. Výstupem modelu je skóre podobnosti. Ve výsledné predikci lze využít pouze skóre podobnosti nebo kombinaci skóre s metrikami získanými ze zarovnání.



Obrázek 4.1: Architektura navrženého modelu. Z otisků jsou vytvořeny grafy a získáno optimální přiřazení markantů. Je zjištěna transformace a otisky jsou zarovnané. Zarovnané otisky jsou zpracovány SimGNN a je provedena predikce.

4.1 Data

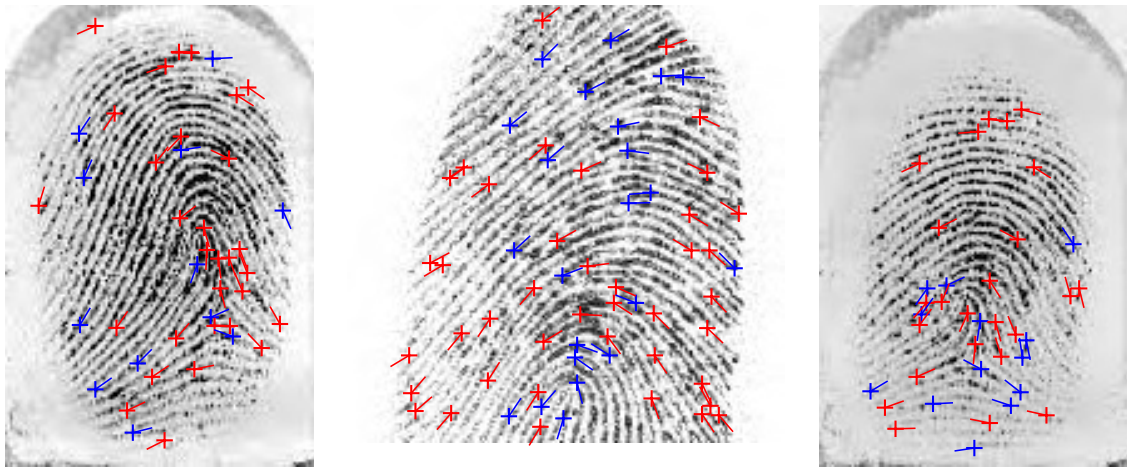
Pro natrénování a vyhodnocení metody jsou potřeba data. Datovou sadou otisků prstů poskytla společnost Innovatrics s.r.o., se kterou jsem při tvorbě této diplomové práce spolupracoval. V této podkapitole jsou shrnuty základní statistické vlastnosti datové sady, zpracování jednotlivých vzorků dat a možné přístupy pro vytvoření grafové reprezentace. Celkově jsou potřeba dvě datové sady, *FingerGenerate* pro natrénování a vyhodnocení zarovnání otisků a *FingerPairs* pro verifikaci.

Datová sada *FingerGenerate* je použita pro natrénování upravené metody *Deep Graph Matching* [30], která využívá kombinatorický solver pro určení optimálního přiřazení markantů. Tato datová sada obsahuje páry grafů a jejich vzorové přiřazení klíčových bodů. Originální datová sada neobsahuje informace o přiřazení, vzorky se musí uměle vygenerovat.

Datová sada *FingerPairs* je potřeba pro vyhodnocení verifikace otisků prstů a natrénování metody SimGNN [3]. Originální datová sada obsahuje seznam pozitivních párů otisků. Z poskytnutého seznamu pozitivních párů lze poté generovat libovolně velký seznam negativních párů. Data jsou v datové sadě uložena ve formě grafové reprezentace otisků a seznamu pozitivních a negativních párů.

Analýza dat

Originální poskytnutá datová sada je rozdělená na trénovací a evaluační část. Každý otisk prstu obsahuje snímek ve formátu *WSQ* (*Wavelet Scalar Quantization*), jedná se o komprimovaný formát snímku otisku ve stupních šedi. Kromě snímku jsou zde uvedeny detekované markanty. Markanty výrazně ulehčují předzpracování dat, není potřeba se zabývat detekcí a klasifikací klíčových bodů a lze tak zcela přeskočit obrazovou reprezentaci otisků a pracovat pouze s markanty otisku. Příklady otisků prstů z datové sady jsou vidět na obrázku 4.2.



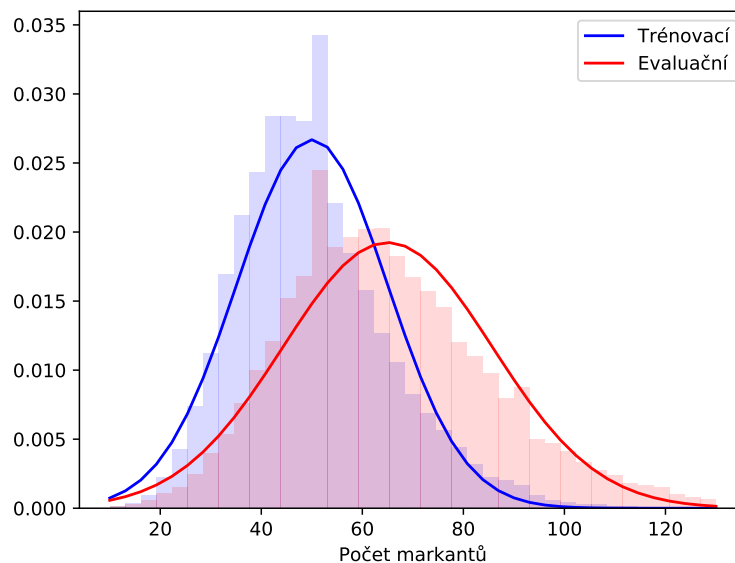
Obrázek 4.2: Příklady otisků z datové sady, která byla poskytnuta společností Innovatrics s.r.o., Modré markanty znázorňují typ vidlička a červené typ ukončení

Markanty jsou zde zadány ve standardním formátu – souřadnice, úhel, typ. Souřadnice markantu jsou definovány dvojicí (x, y) . Souřadnice nejsou normalizovány. Úhel je udán ve stupních a normalizovaný na rozsah $(0, 255)$. Jak již bylo zmíněno v kapitole 2, pro účel verifikace se obvykle používají dva typy markantů, stejně je tomu tak i v této datové sadě. Typ markantu nabývá dvou hodnot – hodnotu typu vidličky a typu ukončení.

Datová sada obsahuje velké množství vzorků. Evaluační sada je tvořena 400 tisíci otisky a trénovací až dvěma miliony. Jednotlivé sady obsahují i seznamy pozitivních párů otisků, získaných ze stejného prstu stejné osoby. Důležitým parametrem otisku prstu je počet markantů. Počet markantů může indikovat kvalitu nasnímaného otisku. Rozložení počtu markantů v jednotlivých sadách lze vidět na histogramech na obrázku 4.3. Z obrázku vyplývá že rozsah počtu markantů je poměrně velký. Datová sada obsahuje i částečné otisky, tvořené pouze malým počtem markantů, ale také otisky dobré kvality obsahující přes sto klíčových bodů. Jednotlivé statistiky jsou uvedeny v tabulce 4.1.

Datová sada	Počet otisků	Pozitivních párů	Markanty (max)	Markanty (průměr)
Trénovací	2 miliony	924 tisíc	196	54.556
Evaluační	400 tisíc	620 tisíc	186	57.084

Tabulka 4.1: Statistické vlastnosti jednotlivých datových sad.



Obrázek 4.3: Rozložení počtu markantů v jednotlivých datových sadách znázorněné pomocí histogramu.

Tvorba grafu z otisku prstů

Pro zpracování otisku grafovou neuronovou sítí je potřeba vytvořit z otisku graf, definovat jeho vrcholy a závislosti mezi nimi. Jako vrcholy grafu lze použít zadané markanty. K definici hran ovšem datová sada neposkytuje žádné další informace.

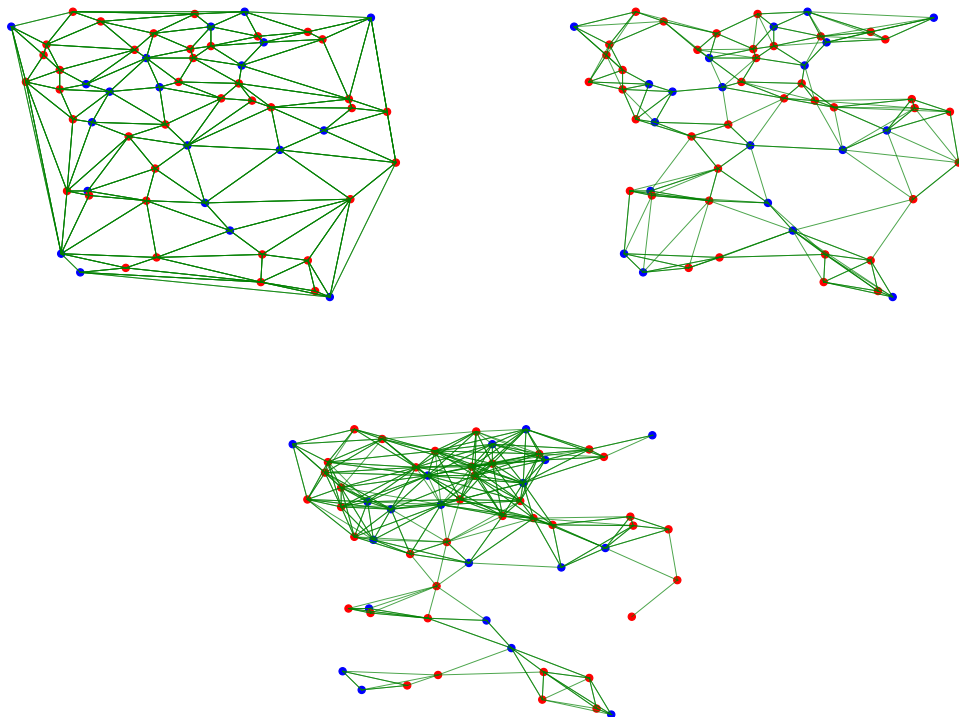
Nejpoužívanějším způsobem pro tvorbu grafu z klíčových bodů je Delaunayho triangulace. Povrch oblasti tvořený markanty se pokryje trojúhelníky tak, aby vytvořené trojúhelníky byly co nejvíce rovnostranné. Tento způsob je při práci s grafovou reprezentací otisků velmi populární [4, 26, 24]. Triangulaci využívá i metoda *Deep Graph Matching* [30].

Alternativní způsob vytvoření závislosti mezi markanty je využití vzdálenosti mezi jednotlivými body. Tento způsob využívá metoda *K – nejbližších sousedů* [37, 7]. Každý vrchol je tak spojený hranou s K vrcholy, které se nachází k vrcholu nejbližší. Tento způsob produkuje velké množství překrývajících se hran a oblasti s více vrcholy obsahují mnohem více hran než odlehlé vrcholy. Počet sousedů je předem pevně daný. Dalším způsobem může být volba hyperparametru jako vzdálenosti od vrcholu. Místo fixního počtu nejbližších vrcholů bude vrchol spojen se všemi vrcholy do určité vzdálenosti. Při fixní vzdálenosti se můžou tvořit samostatné podgrafy, nebo naopak husté shluky. Příklady všech tří přístupů lze vidět na obrázku 4.4.

Generování otisků prstů

Datová sada *FingerGenerate* je použita k natrénování a vyhodnocení predikce optimálního přiřazení markantů. Pozitivní páry datové sady neobsahují vzorové přiřazení markantů, proto je potřeba data uměle vygenerovat.

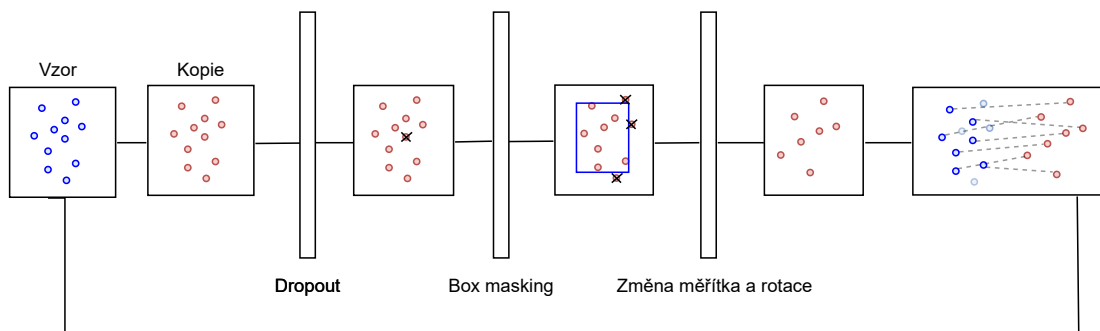
Generování páru otisku probíhá načtením vzorového otisku a vytvoření jeho kopie. Z kopie vzorového otisky jsou náhodně vypuštěny markanty a provedeno maskování podle vý-



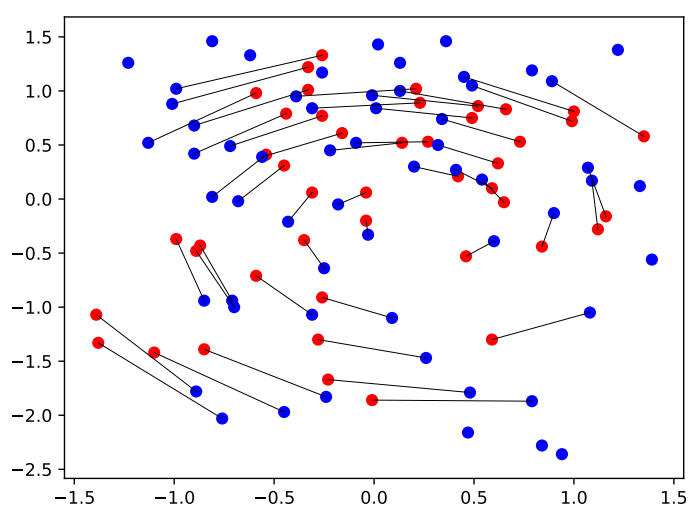
Obrázek 4.4: Příklady tří různých grafových reprezentací otisku prstu. Triangulace (vlevo), K –nejbližších sousedů pro $K=5$ (vpravo), sousedé do vzdálenosti R pro $R = 70$ pixelů (dole).

řezu. Výřez otisku je poté vycentrován a náhodně transformován. Celý proces generování otisku do páru je popsán obrázkem 4.5 a dá se rozdělit do následujících fází:

1. *Dropout* – každý markant je s určitou pravděpodobností smazán.
2. *Box masking* – vymaskování markantů v oblasti určené náhodným výřezem.
3. Vycentrování – posun markantů podle souřadnic středu.
4. Náhodná změna měřítka – vynásobení souřadnic markantů podle náhodně zvolené změny měřítka.
5. Náhodná rotace – rotace markantů podle náhodně zvoleného úhlu.
6. Prohození otisků v rámci páru – prohození otisků v uložené datové struktuře.

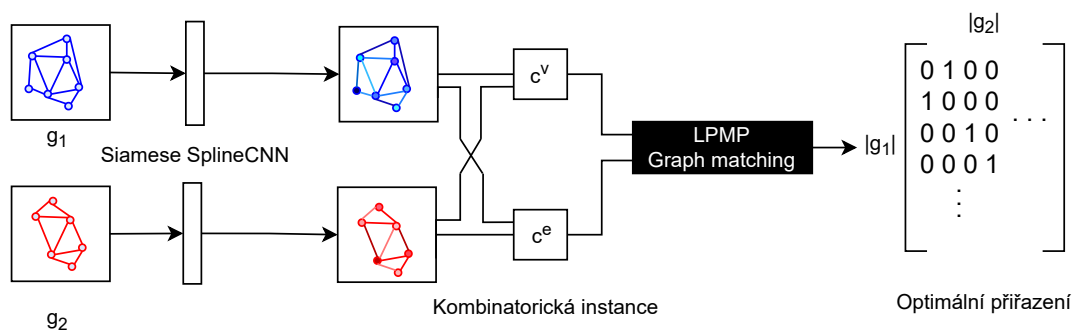


Obrázek 4.5: Proces generování párového otisku ze vzorového.



Obrázek 4.6: Příklad vzorového otisku (modrý) a vygenerovaného otisku (červený) spolu s jejich vzorovým přiřazením.

Cílem generování je co nejvíce reálný otisk korespondující se vzorovým otiskem. Před samotným generováním je vytvořeno přiřazení markantů 1 : 1 mezi vzorem a kopií. Každý markant má určenou malou pravděpodobnost vyřazení, při snímání otisku totiž může dojít k sejmutí otisku ve špatné kvalitě a daný markant nemusí být možné detekovat. Maskování pomocí výřezu poté z otisku udělá částečný otisk. Náhodně je určen box a pomocí něho jsou vymaskovány markanty otisku v oblasti dané boxem. Následně jsou body vycentrovány. Poté je provedena náhodná změna měřítka. Změna měřítka je zde zavedena z důvodu elasticity kůže. Stejný otisk se tak může lišit v závislosti na tlaku, kterým daná osoba tlačila na senzor při snímání. Další transformací je rotace o náhodný úhel, zahrnuje tak při generování fakt, že otisk může být přiložen na senzor pod různým úhlem. Nakonec je provedeno náhodné prohození otisků v páru. Při generování totiž vzorový otisk bude mít vždy více markantů než otisk vygenerovaný. V reálných datech taková závislost není, otisky jsou proto náhodně prohozeny. Uloženy jsou vrcholy a hrany obou otisků, spolu s jejich vzorovým přiřazením. Příklad vygenerovaného páru lze vidět na obrázku 4.6.



Obrázek 4.7: Proces zpracování grafů upravené metody pro získání optimálního přiřazení otisků prstů.

Datová sada párů otisků

Datová sada *FingerPairs* je použita pro vyhodnocení verifikace a pro natrénování modelu SimGNN [3]. V originální datové sadě jsou uvedeny pozitivní páry reálných dat. Zbývá tedy vygenerovat negativní páry. Proces probíhá náhodným výběrem dvou různých otisků ze seznamu pozitivních párů a kontrolou, že právě tato dvojice není pozitivní. Proces generování negativních párů je velmi rychlý, jelikož není potřeba měnit grafy otisků, ale pouze seznam párů.

4.2 Přiřazení markantů dvojice otisku

První krok zarovnání otisků je nalezení optimálního přiřazení. K tomuto účelu je použita metoda *Deep Graph Matching* [30]. Autoři tuto metodu použili pro účel nalezení korespondence mezi objekty stejné třídy na snímcích. Celkově lze práci rozložit na tři fáze – extrakce obrazových příznaků, grafová úprava příznaků a konstrukce kombinatorické instance. První fáze z obrazu objektu detekuje klíčové body a jejich příznaky, vzhledem k tomu že datová sada otisku prstů již klíčové body obsahuje, proto je možné tuto fázi přeskočit. Jako příznaky klíčových bodů jsou použity atributy markantů.

Implementace metody *Deep Graph Matching*

Autoři metody poskytli implementaci v jazyce Python, která využívá knihovny PyTorch a PyTorch-geometric. Metoda také používá kombinatorický solver, který je implementován C++ knihovnou LPMP. Knihovna LPMP poskytuje také obal, který umožňuje diferenciaci solverů pomocí knihovny PyTorch.

Pro účel zpracování již zadaných markantů, musela být metoda upravena. Fáze zpracování obrazu byla přeskočena, byl upraven způsob načítání dat a způsob výpočtu kombinatorické instance. To vše z důvodu různého formátu vstupních dat.

Úprava metody pro účel přiřazení markantů

Data v dogenerované datové sadě otisku prstů jsou uloženy ve formě grafů. Část metody používající VGG pro extrakci klíčových bodů a příznaků lze přeskočit, klíčové body jsou již dostupné z datové sady. Nahrazena bude i způsob tvorby grafu za vlastní implementaci,

kteřá se provádí již při tvorbě datové sady, tak aby bylo možné vyzkoušet různé grafové reprezentace otisků. Načtené grafy z datové sady *FingerGenerate* jsou již připravené pro zpracování grafovou neuronovou sítí *SplineCNN* [14].

Další úprava spočívá ve změně výpočtu cenových vektorů vrcholů a hran (kombinatorické instance), která je použita pro vstup do kombinatorického solveru. Tato úprava je provedena z důvodu absence globálních příznaků, které v původní metodě poskytovala neuronová síť *VGG*. Původní výpočet kombinatorické instance pro vrcholy (analogicky i pro hrany) je definován rovnicí

$$c_{i,j}^v = \sum_k f_s^v(i)_k a_k f_t^v(j)_k, \quad (4.1)$$

kde $c_{i,j}^v$ je cenový vektor páru grafu i, j , $f_s^v(i)$ je vektor příznaků i -tého vrcholu vstupního grafu s a hodnota a reprezentující globální příznaky. Vzorec je definovaný jako součet součinů příznaků vrcholů obou grafů a hodnotou a , která vznikla právě sjednocením globálních příznaků. V novém vzorci bude výpočet instance definovaný bez hodnoty a rovnicí

$$c_{i,j}^v = \sum_k f_s^v(i)_k f_t^v(j)_k. \quad (4.2)$$

4.3 Verifikace otisku

Z metody *Deep Graph Matching* [30] je získáno optimální přiřazení markantů jednoho otisku na markanty otisku druhého. Přiřazení je zadáno ve formě matice o velikosti $m \times n$, kde m a n jsou počty vrcholů v grafech. Matice obsahuje vždy $k = \min(m, n)$ přiřazení. Získaná matice se použije k zarovnání otisků.

Prvním krokem zarovnání otisků je získání transformace. Z k přiřazených párů markantů je možné získat parametry afinních transformací – matici rotace, vektor translace a změnu měřítka. Pomocí matice transformace jsou promítnuty vrcholy jednoho otisku na druhý.

K verifikaci zarovnaných otisků lze přistoupit několika způsoby. Prvním použitým přístupem je k verifikaci použít samotné zarovnání, kde jsou k dispozici metriky jako například vzdálenost, rozdíl úhlů nebo správnost typů zarovnaných markantů. Dalším použitým způsobem je použití SimGNN [3] pro získání skóre podobnosti zarovnaných grafů. Oba tyto způsoby lze také zkombinovat a navrhnout funkci skóre. Funkce skóre využívá všechny tyto metriky. Na základě funkce skóre lze provést rozhodnutí o souhlasnosti otisků.

Zarovnání otisku

Zarovnání otisků začíná nalezením parametrů transformace. Prvním a nejjednodušším způsobem je metoda nejmenších čtverců. Metoda nejmenších čtverců je forma matematické regresní analýzy, používaná k nalezení nejlepší shody pro dané datové body, tak aby součet vzdáleností jednotlivých bodů byl co nejmenší. Cílem je tedy minimalizace chybové funkce

$$err = \sum_{i=1}^N \|RA^i + t - B^i\|^2, \quad (4.3)$$

kde A, B jsou matice datových bodů ve formátu $[x, y, 1]$, R je matice rotace a t je vektor translace. Tento způsobem je velmi náchylný na chybová data, jelikož používá pro získání transformace všechny datové body grafů.

Dalším možným způsobem je použití singulárního rozkladu *SVD* (Singular Value Decomposition) k nalezení matice rotace. *SVD* je rozklad matice M na součin $U\Sigma V^T$. Matice Σ o velikosti $m \times n$ obsahuje na hlavní diagonále singulární hodnoty matice M . Aplikováno na problém nalezení matice rotace lze matici M vyjádřit rovnicí

$$\begin{aligned} M &= (A - c_A)(B - c_B)^T \\ M &= U\Sigma V^T \\ R &= VU^T. \end{aligned} \tag{4.4}$$

Jakmile je nalezena matice rotace R lze vyjádřit vektor translace z rovnice

$$\begin{aligned} R \times A + t &= B \\ R \times c_A + t &= c_B \\ t &= c_B - R \times c_A, \end{aligned} \tag{4.5}$$

kde A, B jsou matice datových bodů, c_A, c_B jsou středy spočítané jako průměr souřadnic bodů a dvojice R, t pak matice rotace a vektor translace.

Posledním způsobem je použití algoritmu *RANSAC* (*Random sample consensus*). *RANSAC* je iterační metoda odhadu parametrů matematického modelu ze souboru pozorovaných dat, který obsahuje odlehlé hodnoty, přičemž odlehlým hodnotám nelze přiznat žádný vliv na hodnoty odhadů. To znamená, že dokáže dobře fungovat i na datech, které mohou obsahovat chybně přiřazené datové body.

Verifikace na základě transformovaných bodů

Otisky jsou zarovnány pomocí odhadnuté transformace. Nyní je možné přistoupit k verifikaci. Cílem je určit zda oba otisky na vstupu pochází ze stejného prstu a od stejné osoby. K tomuto účelu je potřeba navrhnout funkci skóre. Funkce pomocí získaných metrik vyhodnotí skóre pro daný pár otisků, ten je poté porovnán s určenou prahovou hodnotou a je provedeno rozhodnutí.

Prvním možným způsobem jak přistupovat k verifikaci na základě transformovaných bodů je rozhodnout o správnosti přiřazení každého páru markantů. U každého páru markantů jsou k dispozici následující informace: vzdálenost bodů, rozdíl úhlů, a shodnost jejich typů. Uvažme následující rozhodovací podmínku

$$\|a_i b_i\| < t_1 \wedge |\angle a_i - \angle b_i| < t_2 \wedge \text{type}(a_i) == \text{type}(b_i). \tag{4.6}$$

Podmínka se skládá ze tří predikátů. Pokud je vzdálenost bodů $\|a_i b_i\|$ menší než práh t_1 a rozdíl úhlů markantů $|\angle a_i - \angle b_i|$ menší než práh t_2 a jejich typ je shodný pak se jedná o správně přiřazený markant. Podmínka se vyhodnotí pro každou dvojici přiřazených markantů, výsledné skóre pak bude vypočteno jako podíl počtu správně přiřazených markantů a počtu všech dvojic. Skóre tak bude z rozsahu $(0, 1)$ a jeho hodnota bude reprezentovat míru jistoty, že daný pár otisků pochází od stejné osoby.

Alternativní způsob spočívá v agregaci jednotlivých metrik párů markantů a učinit rozhodnutí na základě agregovaných hodnotách. Lze například získat průměrnou vzdálenost mezi vrcholy otisků a průměrný rozdíl úhlů markantů. Na základě toho lze výpočet skóre \mathcal{S} definovat rovnicí

$$\mathcal{S} = \frac{1}{N} \left(\sum_{i=1}^N (d_i + r_i) - h \right), \quad (4.7)$$

kde d_i je vzdálenost a r_i rozdíl úhlu i -té dvojice, N počet dvojic a h pak poměr správných typů markantů. Ze vzorce také vyplývá že čím menší hodnota skóre, tím větší šance, že se jedná o stejné otisky. Výsledné skóre je před učiněním rozhodnutí nutno porovnat s hodnotou prahu.

Verifikace pomocí SimGNN

Dodatečně lze pro porovnání otisků použít model SimGNN [3]. Výstupem modelu SimGNN je skóre podobnosti. Již na základě tohoto skóre a prahu lze provést rozhodnutí o shodnosti otisků. Výsledek verifikace záleží hlavně na skóre podobnosti.

Dodatečně lze ještě skóre podobnosti zakomponovat do výpočtu skóre z předchozího způsobu verifikace na základě transformovaných bodů. Výsledný vzorec tak je definovaný podobně jako v rovnici 4.7, ale s tím rozdílem, že od něho je odečteno vážené skóre podobnosti $w * s$. Váha w určuje do jaké míry se má výsledek z modelu SimGNN projevit na výsledném rozhodnutí.

Kapitola 5

Experimenty

Při popisu návrhu byl v každém bodě řešení uveden výčet přístupů, které je možné na řešení dílčí problematiky aplikovat. Nad jednotlivými částmi řešení byly provedeny experimenty. Finální podoba architektury pak vychází z výsledků těchto experimentů. Postupně byla otestována metoda přiřazení markantů, odhad transformace a nakonec vyhodnocena verifikace otisků prstů.

5.1 Přiřazování markantů

Pro každou konfiguraci byl model natrénován na části vygenerované datové sady, která obsahovala 20 tisíc vzorků. Výsledky jsou poté uvedeny po vyhodnocení modulu validační částí datové sady (5 tisíc párů otisků). Model není možné kvantitativně vyhodnotit přímo nad páry otisků z originální datové sady, jelikož neobsahují vzorové přiřazení. Pro vyhodnocení jsou zde uvedeny dvě metriky – přesnost a F1 skóre. Přesnost je spočtena jako průměrný poměr správně přiřazených markantů. Poměr se získá vynásobením vzorové a predikované matice přiřazení po prvcích, následně je proveden součet všech prvků a vydělen počtem přiřazení (počtem vrcholů menšího grafu).

Vliv metody tvorby grafové reprezentace

Pro vyhodnocení tohoto experimentu byly zafixovány parametry generování otisků. Rotace v rozmezí $\pm 45^\circ$, změna měřítka $\pm 5\%$, vynechání bodů s pravděpodobností 0.05 a ořezání otisku v boxu, který je definován náhodnou vzdáleností do 50 pixelů z každé strany.

Výsledky experimentu je uveden v tabulce 5.1. Vyplývá z nich že nejlepší přesnosti dosáhla grafová reprezentace pomocí Delaunayho triangulace, která v této konfiguraci dosáhla průměrné přesnosti 69.1% správně přiřazených markantů. Metody využívající vzdálenosti bodů pro tvorbu grafu – K nejbližších sousedů (K-nearest) a sousedů do určité vzdálenosti (R-nearest) se ukázaly jako nepoužitelné. V dalších experimentech bude dále používána Delaunayho triangulace.

Vliv obtížnosti generovaných dat

Přesnost určení optimálního přiřazení je závislá na obtížnosti vygenerované datové sady. Cílem tohoto experimentu není dosáhnout nejlepší možné přesnosti, ale najít kompromis mezi obtížností a přesností. Obtížnost datové sady by měla reflektovat otisky v originální datové sadě. Obtížnost vygenerované datové sady je určena především dvěma parametry –

Grafová reprezentace	Přesnost	F1
Delaunayho triangulace	0.691	0.687
K-nearest (K=5)	0.127	0.117
K-nearest (K=3)	0.098	0.092
R-nearest (R=70 px)	0.084	0.081
R-nearest (R=70 px)	0.056	0.049

Tabulka 5.1: Vyhodnocení závislosti formy grafové reprezentace na přesnosti odhadu optimálního přiřazení.

úhlem rotace a poměrem bodů grafů. Rotace je generována náhodně v definovaném rozsahu. Poměr bodů je ovlivněn pravděpodobností výpadku bodů (v procentech) a velikostí oblasti ořezání. Oblast ořezu je definována náhodně vygenerovanou vzdáleností z každé strany, které pak definují výslednou oblast výřezu. V experimentu je zafixována metoda grafové reprezentace triangulace a parametr pravděpodobnosti výpadku bodu na hodnotu 5%.

Výsledky experimentu uvádí tabulka 5.2. Z výsledků vyplývá, že přesnost klesá rychleji při zvětšování rozsahu úhlu rotace než při zmenšování poměru bodů mezi otisky. Optimální konfigurace generátoru byla na základě tohoto experimentu zvolena jako rozmezí rotace $\pm 60^\circ$ a oblast ořezu v rozmezí 50 pixelů.

Oblast ořezu (px)	Úhel rotace ($^\circ$)	Poměr bodů	Přesnost	F1
50	30	0.76	0.838	0.826
50	45	0.77	0.691	0.687
50	60	0.78	0.572	0.554
25	45	0.84	0.787	0.751
25	60	0.85	0.615	0.607
100	45	0.59	0.594	0.588
100	60	0.59	0.512	0.503

Tabulka 5.2: Vyhodnocení závislosti obtížnosti generování dat na přesnosti odhadu optimálního přiřazení.

Vliv parametrů GM solveru

Metoda využívá kombinatorický solver řešící grafové přiřazení, konkrétně se jedná o solver z knihovny *LPMP*. Poskytnutá implementace umožňuje nastavit počet iterací solveru a parametr `primalInterval`. Cílem tohoto experimentu bude zjistit chování modelu při změně těchto parametrů. Se zvyšujícím se počtem iterací se počítá s faktem, že přesnost algoritmu by měla být vyšší, ale za cenu většího výpočetního času. V experimentu byla vyhodnocena přesnost na validační části datové sady a čas potřebný ke zpracování jednoho páru otisků prstu.

V první části experimentu byl zafixován parametr `primalInterval` a byla zkoumána závislost počtu iterací na přesnosti modelu. Výsledek vyjadřuje tabulka 5.3. Z tabulky plyne, že pro dosažení nejlepší přesnosti trvá zpracování jednoho páru otisků průměrně 28 milisekund. Kompromis mezi časem zpracování a přesností se nachází při volbě počtu iterací mezi hodnoty 50 a 75. Pár otisků se tak zpracuje průměrně do 10 milisekund a úbytek přesnosti není tak znatelný.

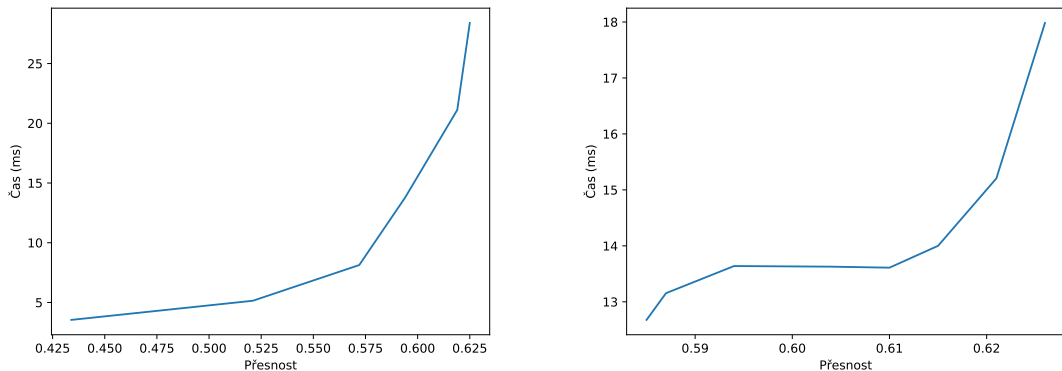
primalInterval	Počet iterací	Přesnost	Čas (ms)
10	300	0.625	28.393
10	200	0.619	21.119
10	100	0.594	13.782
10	50	0.572	8.132
10	25	0.521	5.144
10	10	0.434	3.543

Tabulka 5.3: Chování modelu při změně počtu iterací při zafixované hodnotě `primalInterval`. Přesnost je přímo úměrná počtu iterací, při malém počtu iterací začne přesnost metody rychle degradovat.

primalInterval	Počet iterací	Přesnost	Čas (ms)
30	100	0.585	12.677
20	100	0.587	13.154
10	100	0.594	13.639
5	100	0.604	13.628
1	100	0.626	17.982

Tabulka 5.4: Chování modelu při změně parametru `primalInterval` při fixním počtu iterací. Přesnost optimálního přiřazení je na hodnotě parametru závislá nepřímo úměrně.

V druhé části byla naopak zafixována hodnota počtu iterací a zkoumaným parametrem byl parametr `primalInterval`. Výsledek je vyjádřen v tabulce 5.4. S klesající hodnotou parametru roste přesnost výsledného modelu. Minimální hodnota, kterou umožňuje implementace nastavit je 1. Průměrný čas zpracování páru otisků při klesající hodnotě parametru mírně roste. Výsledky obou částí jsou také znázorněny grafy na obrázku 5.1, lze na nich pozorovat závislost času a přesnosti při různé volbě hodnot parametrů.



Obrázek 5.1: Grafy vyjadřující závislost výpočetního času na přesnosti odhadu optimálního přiřazení. Křivka na levém grafu vyjadřuje chování systému při zvyšování počtu iterací, na pravém grafu zmenšování parametru `primalInterval`. V prvním grafu jde vidět konvergence přesnosti při zvyšujícím se počtu iterací.

5.2 Transformace

Pomocí přiřazení lze odhadnout parametry transformace, které jsou použity k zarovnání otisků na sebe. V prvním experimentu jsou porovnány jednotlivé metody pro odhad transformace a v druhém je vyhodnocena konkrétní metoda *RANSAC*.

Metody odhadu parametrů transformace

V tomto experimentu byla porovnána metoda nejmenších čtverců (*LSM*), singulární rozklad (*SVM*) a metoda *RANSAC*. Cílem experimentu bylo sledovat a porovnávat chování jednotlivých metod při zvyšování počtu chyb F . Postupně je proveden odhad transformace ze vzorového přiřazení ($F = 0$) až po úplné náhodně přiřazené body. Pro účel experimentu byl vybrán pár otisků obsahující 33 a 49 markantů.

Výsledky experimentu znázorňují grafy na obrázku 5.2. Vyhodnocena je metrika změna průměrné vzdálenosti mezi přiřazenými body, změna parametrů rotace, translace a změny měřítka oproti vzorovým parametrům transformace. Změna měřítka pro metodu *SVD* není uvedena, jelikož použitá implementace neobsahovala odhad měřítka. Metoda nejmenších čtverců je použita jako *baseline* řešení a podle očekávání její odhad transformace není odolný vůči chybám. Z výsledků vyplývá, že nejstabilnější odhad transformace poskytuje metoda *RANSAC*. Odhad transformace je velmi stabilní a průměrná vzdálenost zarovnaných bodů je ze všech tří porovnaných metod nejnižší.

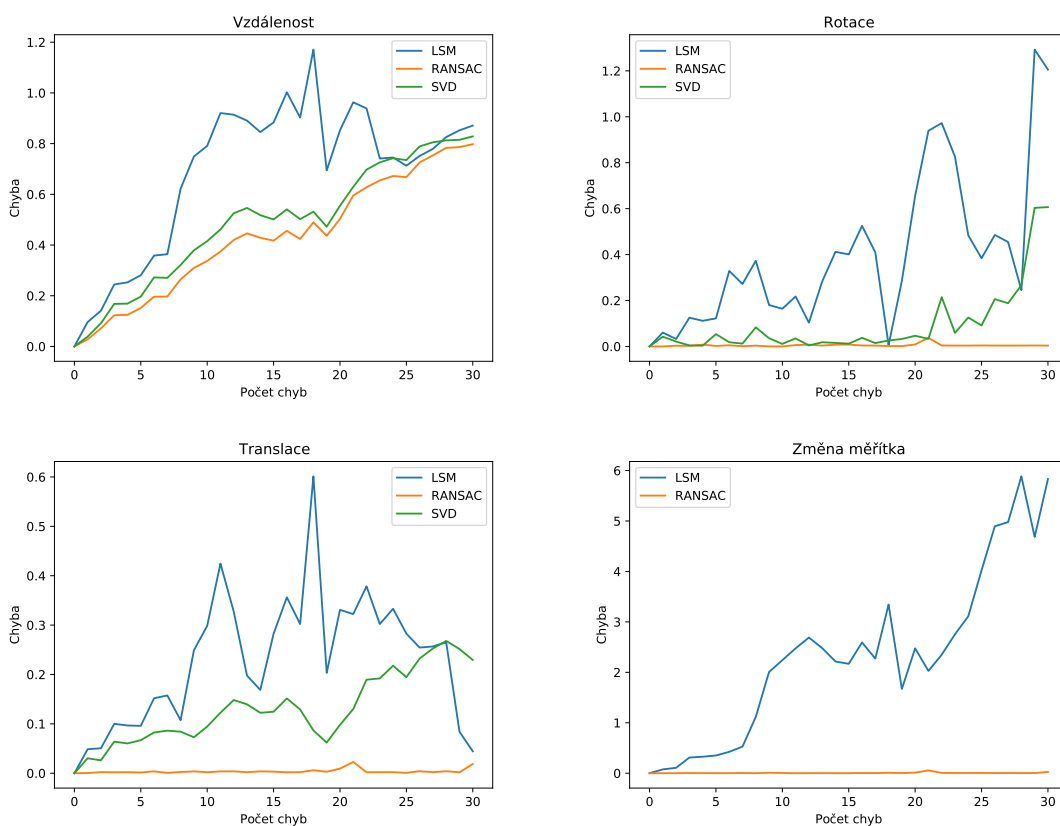
RANSAC

Předchozí experiment ukázal, že nejlepší z otestovaných metod je metoda *RANSAC*. *RANSAC* pracuje na principu vyhodnocení modelu za účelem hledání tzv. *inliers* bodů, které se nachází ve vzdálenosti od daného prahu, tyto body definují daný model. Cílem tohoto experimentu je nalezení optimálních parametrů pro nejlepší odhad transformace. Vyhodnocení probíhá na datech z originální evaluační sady.

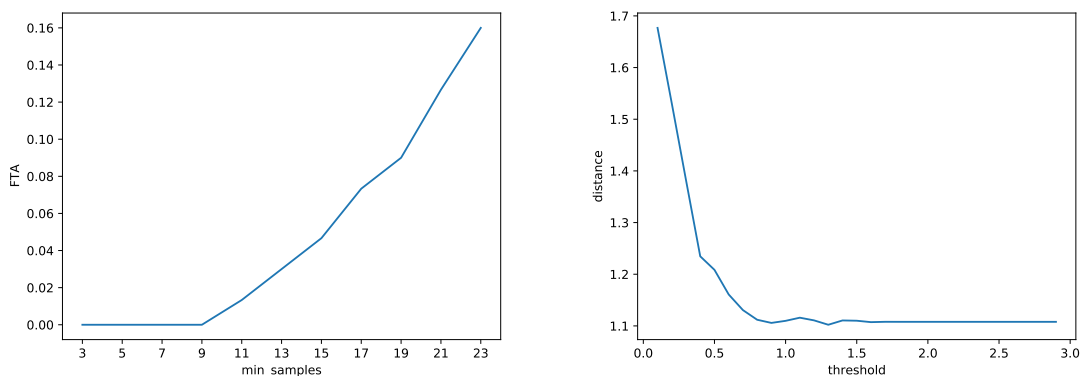
Výsledky experimentu jsou znázorněny na grafech na obrázku 5.3. Na prvním grafu je vyhodnocen vliv parametru minimálního počtu *inliers* na metriku FTA, práh byl nastaven na hodnotu 1.0. Metrika FTA (*Failure to Acquire Rate*) znázorňuje chybu systému, kdy se nepodaří systému zpracovat daný vzorek. Z grafu vychází, že bezpečnou volbou minimálního počtu *inliers* je hodnota 5. Druhý graf znázorňuje vliv volby prahu na průměrné vzdálenosti zarovnaných bodů. Z experimentu vyplývá že optimální hodnota prahu je 1.0.

5.3 Verifikace

Výsledky předchozího experimentu definují způsob zpracování optimálního přiřazení a zarovnání otisků. Tato část se zaměří na použití navržené a otestované metody na verifikaci otisků prstů. Verifikace spočívá v rozhodnutí zda dva otisky pochází ze stejného prstu stejné osoby. Pro vyhodnocení zde bude použita část originální evaluační sady s dogenerovanými negativními páry.



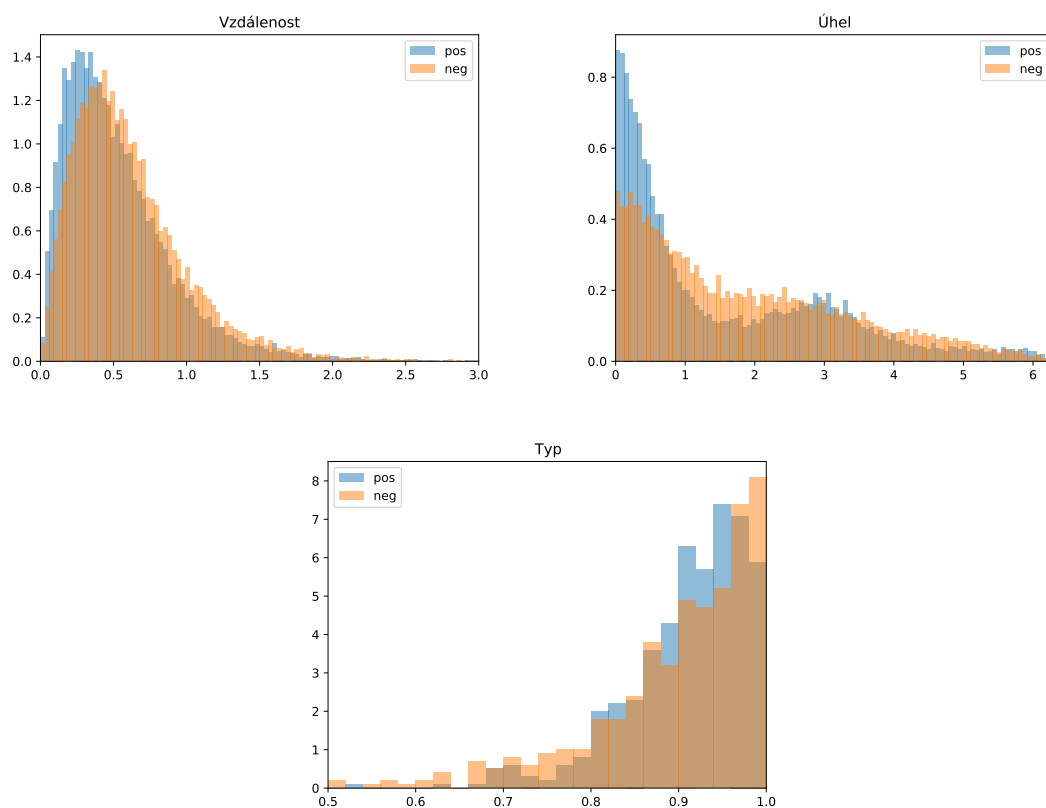
Obrázek 5.2: Porovnání chování jednotlivých metod pro odhad transformace při zvyšujícím se počtu chyb. Porovnání se provádí na základě rozdílu mezi sledovaným parametrem a vzorem. U vzdálenosti je sledován rozdíl průměrné vzdálenosti a vzdálenosti, která byla odhadnuta z bezchybného přiřazení.



Obrázek 5.3: Výsledky vyhodnocení metody *RANSAC*. Na levém grafu je sledována metrika FTA při zvyšujícím se počtu minimálního počtu *inliers*. Pravý graf znázorňuje vliv volby prahu na průměrnou vzdálenost mezi zarovnanými body otisků.

Analýza metrik zarovnaných otisků

Mezi zarovnaným párem otisků je možno sledovat tři metriky. První metrikou je vzdálenost mezi jednotlivými páry. Souhlasný pár otisků by měl mít vzdálenosti mezi zarovnanými



Obrázek 5.4: Rozložení vzdálenosti, rozdílů úhlů a souhlasnosti typu u pozitivních (modře) a negativních párů otisků (oranžově). Rozdíl úhlu je udáný v radiánech a typ v poměru souhlasných typů u otisku.

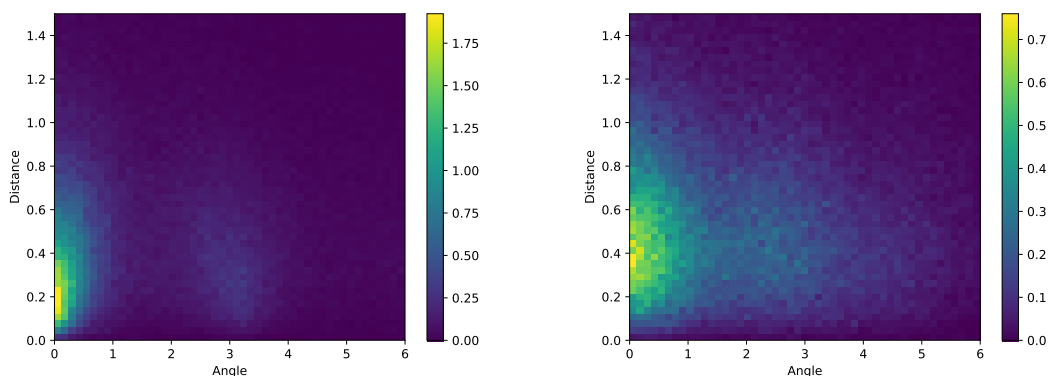
body nižší než nesouhlasný. Každý markant kromě souřadnic obsahuje i úhel a typ. Další metriky jsou rozdíl úhlů zarovnaných bodů a informace, jestli dané markanty mají shodný typ. Cílem tohoto experimentu je analyzovat tyto metriky a zjistit, jestli jsou jejich rozložení u souhlasných a nesouhlasných párů otisků natolik rozdílné, aby tyto metriky bylo možné použít pro verifikaci.

Odhad rozložení jednotlivých metrik lze vidět na obrázku 5.4. Z histogramů vyplývá, že nejvhodnější metrikou pro rozlišení otisků je rozdíl úhlů. Hranicí, která rozdělí souhlasné a nesouhlasné páry by potom mohla být hodnota rozdílů 0.7 radiánů. Vzdálenost také poskytuje možné rozdělení dat a to hranicí se vzdáleností 40 pixelů. Správnost typů, která je znázorněna třetím histogramem, není vhodnou vlastností pro rozhodování o souhlasnosti otisků. Kombinaci metrik vzdálenosti a rozdílů úhlu vyjadřují histogramy 5.5.

Verifikace na základě transformovaných markantů

Rozhodnutí o souhlasnosti otisků můžeme provést na základě dvou přístupů, oba přístupy využívají metriky vzdálenost, rozdíl úhlů a typ.

Prvním přístupem je rozhodnutí, jestli je pár markantů správně přiřazen. Toto rozhodnutí se provede pro všechny body přiřazení páru otisků a výsledné skóre je poměr správně přiřazených markantů. Hranice pro rozhodnutí byly v tomto experimentu zvoleny na zá-



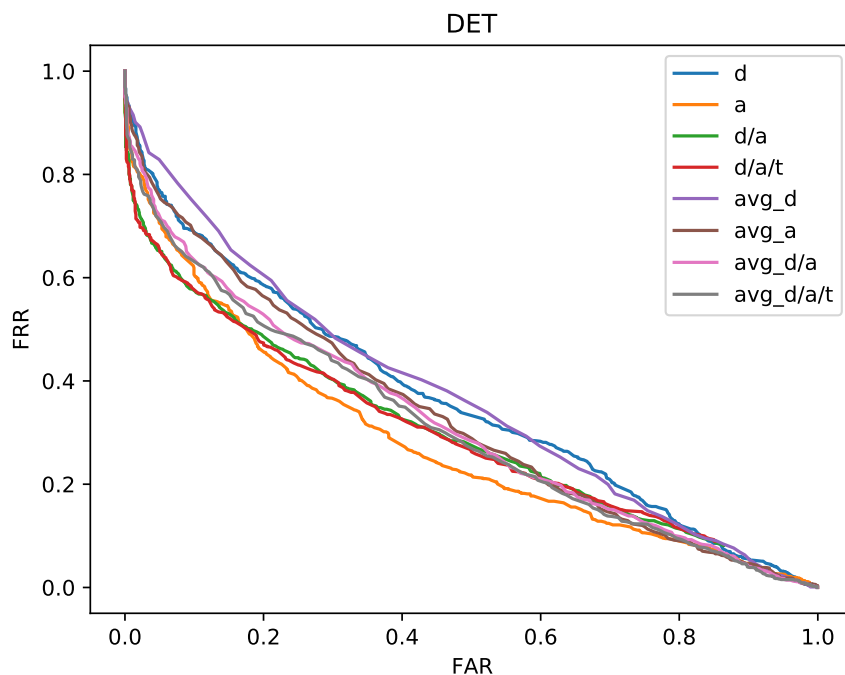
Obrázek 5.5: Rozložení kombinace vzdálenosti a rozdílu úhlů v pozitivních (levý histogram) a negativních párech otisků (pravý histogram).

kladě analýzy – vzdálenost 40 pixelů a rozdíl úhlů 0.7 radiánů. Druhý přístup průměruje vzdálenosti a rozdíly úhlů. Rozhodnutí je poté prováděno na součtu těchto průměrů. Typy lze do tohoto přístupu zahrnout odečtením poměru správně přiřazených typů od skóre.

Výsledky experimentu jsou vyjádřeny pomocí chybových metrik FAR (*False Acceptance Rate*), FRR (*False Rejection Rate*) a EER (*Equal Error Rate*). EER znázorňuje chybu při volbě prahu takového, že se rovná míra FAR a FRR. Vyhodnoceny byly oba přístupy pro různé kombinace použitých metrik. Výsledky EER pro jednotlivé konfigurace lze vidět v tabulce 5.5. Výsledky FAR/FRR jsou vyjádřeny ve formě DET (*Detection Error Tradeoff*) křivky na obrázku 5.6. Nejnižší chybu (EER = 0.338) vykazuje metoda pro verifikaci otisků je využití poměru správně přiřazených markantů, kde je rozhodnutí provedeno pouze na základě rozdílu úhlů.

Metoda	Vzdálenost	Úhel	Typ	EER
minutae_d	✓	-	-	0.397
minutae_a	-	✓	-	0.338
minutae_d/a	✓	✓	-	0.355
minutae_d/a/t	✓	✓	✓	0.359
avg_d	✓	-	-	0.428
avg_a	-	✓	-	0.380
avg_d/a	✓	✓	-	0.383
avg_d/a/t	✓	✓	✓	0.375

Tabulka 5.5: Výsledky verifikace otisků prstů při použití různých kombinací metrik. Metody *minutae* využívají poměr správně přiřazených markantů a metody *avg* využívají průměry metrik.



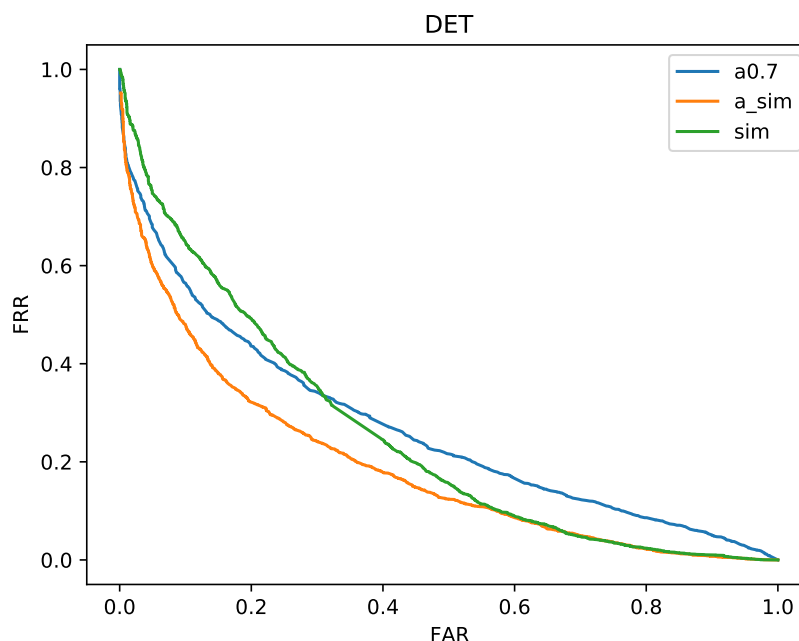
Obrázek 5.6: DET křivka znázorňující chybové míry FAR, FRR pro jednotlivé přístupy verifikace.

Verifikace na základě SimGNN

Alternativně lze pro verifikaci použít model SimGNN [3]. Tato architektura je určena pro natrénování odhadu podobnosti páru grafů. Autoři tuto metodu testují na odhadu GED (*Greatest Edit Distance*) nebo na nalezení největšího podgrafu. V tomto experimentu je model natrénován přímo na určení shodnosti grafů, výstup modelu je pak skóre podobnosti. Skóre podobnosti lze chápat jako pravděpodobnost, že dané dva otisky pochází od stejné osoby. V experimentu je využita architektura modelu, která využívá oba přístupy navržené autory SimGNN, tedy využití příznaků na grafové úrovni i porovnávání v rámci jednotlivých párů vrcholů.

Výsledky experimentu jsou uvedeny v tabulce 5.6, pro srovnání je zde uvedena nejlepší metoda pro verifikaci z předchozího experimentu, která využívá rozhodnutí o správnosti přiřazení markantů rozdílem úhlů. Samostatné použití skóre podobnosti dosahuje lepší EER než konfigurace získaná v předchozím experimentu. Experiment sleduje dva alternativní způsoby získání skóre – odhad skóre z nezarovnaných (`sim_pre`) a zarovnaných (`sim`) párů otisků. Zarovnání otisků má vliv na výslednou verifikaci. Skóre, které bylo získáno ze zarovnaných grafů, vykazuje lepší výsledky.

Skóre podobnosti lze zkombinovat s metrikou použitou v přechodném experimentu. Vzorec výsledného skóre je pak součet poměru správně přiřazených markantů (určených rozdílem úhlů) a skóre podobnosti. Váha jednotlivých složek je určena rovnoměrně, přiřazení větší váhy k poměru nebo ke skóre nevedlo ke zlepšení verifikace. Kombinace je ve výsledcích uvedena jako `a_sim`. Porovnání všech tří způsobů je shrnuto grafem DET křivky na obrázku 5.7. Kombinace obou přístupů dosahuje nejlepších výsledků verifikace s $EER = 0.267$. Problémové případy pozitivních párů jsou vyjádřeny na obrázku 5.8.



Obrázek 5.7: DET křivka znázorňující chybové metriky FAR, FRR pro tři způsoby verifikace otisků – metoda využívající pouze rozdíly úhlů (a0.7), skóre podobnosti(sim) a kombinaci obou přístupů (a_sim).

Metoda	Úhel	Podobnost	EER
a0.7	✓	-	0.338
sim_pre	-	✓	0.321
sim	-	✓	0.311
a_sim_pre	✓	✓	0.269
a_sim	✓	✓	0.267

Tabulka 5.6: Porovnání verifikace otisků prstů metod, které používají skóre podobnosti.

5.4 Shrnutí experimentů

První část experimentů se věnovala porovnání formátu grafových dat a nastavení metody pro určení optimálního přiřazení markantů pro zarovnání otisků. V experimentech byla sledována přesnost přiřazení markantů v generovaných párech otisků. Nejlepší výsledky vykazovala grafová reprezentace otisku pomocí Delaunayho triangulace. Na přesnost metody má vliv obtížnost použitých dat a parametry kombinatorického solveru. Obtížnost dat byla určena tak, aby generovaná data co nejlépe reflektovali reálnou sadu otisků prstů a zároveň byla zachována přesnost metody. Pro generování otisků byla zvolena rotace v rozsahu $\pm 60^\circ$ s poměrem bodů 0.78, tedy se ztrátou 22% bodů. Kombinatorický solver umožnil nastavit počet iterací algoritmu a proměnnou `primalInterval`. Volba počtu iterací byl kompromis mezi rychlostí a přesností, jeho hodnota byla nastavena na 50 iterací. Minimální hodnota `primalInterval`, nastavená na hodnotu 1, vykazovala nejlepší přesnost.

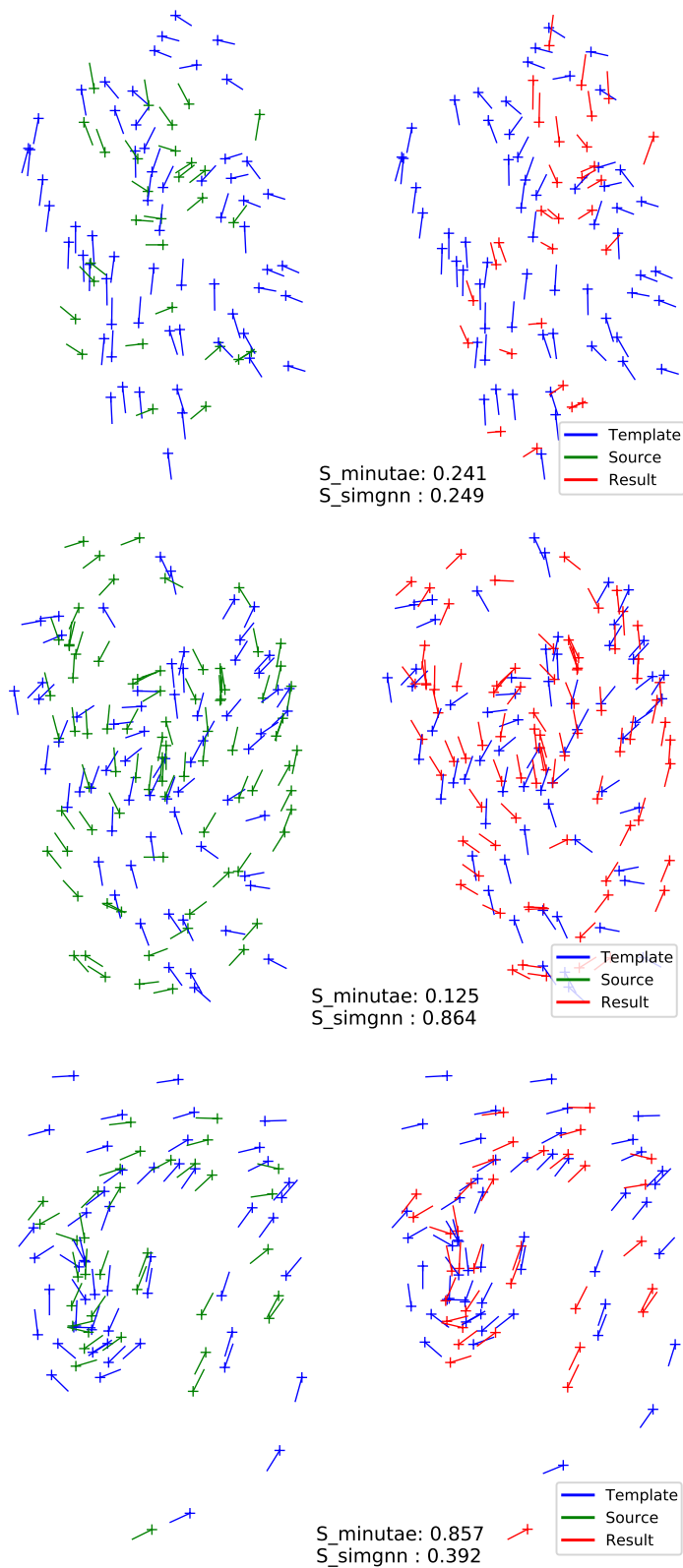
Druhá část experimentů se zaměřila na odhad transformace ze získaného přiřazení markantů. Porovnány byly tři metody. Nejpřesnější a nejodolnější vůči chybám se ukázala metoda RANSAC. Dále byl vyhodnocen vliv parametrů metody RANSAC na stabilitu a přesnost metody. Optimální konfigurace obsahuje minimální počet bodů *inliers* na hodnotu 5 a práh nastavený na hodnotu 1.0.

Třetí část experimentů vyhodnocuje verifikaci získaného modelu z předchozích částí experimentů. Nejdříve je otestována verifikace na základě odhadnuté transformace, pro rozhodnutí jsou použity metriky získané ze zarovnaných otisků podle přiřazení. Ze všech vyzkoušených kombinací se ukázal nejlepší přístup, který rozhodoval o správnosti každého přiřazení markantů na základě podmínky, jestli je rozdíl úhlu markantů menší než hodnota 0.7. Rozhodnutí o shodnosti otisků bylo provedeno na základě poměru správně přiřazených markantů. Následně byla vyhodnocena verifikace pomocí skóre podobnosti získané z modelu SimGNN [3]. Samotné použití skóre podobnosti dosahovalo lehce nižších hodnot chybové metriky EER. Nejlepších výsledků (EER= 0.267) dosahovala kombinace obou přístupů, tedy rozhodnutí na základě součtu poměru správně přiřazených bodů a skóre podobnosti.

Návrhy pro zlepšení

Dosahovaná přesnost verifikace je nepoužitelná pro reálný verifikační systém. Jedním z faktorů limitující výkonost metody je počet příznaků ke každému markantu. Každý markant obsahuje pouze čtveřici příznaků – souřadnice bodu, typ a úhel. Pro získání dalších příznaků by bylo možné použít například model založený na konvolučních neuronových sítích, který získá pro každý detekovaný markant určitý počet příznaků. Příznaky by tak byly získány z okolí bodu na snímku otisku prstů. Tímto způsobem je možné si pro každý markant definovat velké množství příznaků, které je možné použít pro atributy vrcholů ve vytvořeném grafu. Větší počet příznaků vrcholů by mohl do jisté míry zlepšit přesnost odhadu optimálního přiřazení, ale především pak výkon modelu SimGNN [3] pro výslednou verifikaci.

Dalším zlepšení by mohla poskytnout předem definovaná závislost markantů v rámci otisku prstu. V datové sadě by otisk nebyl definován pouze pomocí sady markantů, ale také pomocí hran mezi nimi. Tyto závislosti by poté bylo možné použít pro grafovou reprezentaci otisků, který lépe reprezentuje rozložení daných klíčových bodů.



Obrázek 5.8: Příklady špatně klasifikovaných pozitivních párů otisků prstů. Vstupní otisk (zeleně) je zarovnán na šablonu (modře), výsledek je zobrazen červeně. $S_{minutae}$ vyjadřuje poměr správných přiřazení a S_{simgmn} skóre podobnosti.

Kapitola 6

Závěr

Verifikace otisků prstů pomocí grafových neuronových sítí je nový a nevyzkoušený problém. V této práci byla navržena metoda pro zarovnání a porovnání grafů otisků prstů. Z otisků prstů jsou vytvořeny grafy pomocí zadaných vstupních markantů a aplikace Delaunayho triangulace. Triangulace byla zvolena na základě frekvence použití v již existujících řešeních a výsledků experimentu. Triangulace dosahovala nejlepších výsledků ze všech vyzkoušených možností grafové reprezentace.

Fáze zarovnání otisků byla implementována pomocí metody *Deep Graph Matching* [30], která řeší problém grafového přiřazení pomocí kombinatorického solveru. Problém grafového přiřazení je NP – úplný problém. Metoda používá grafovou neuronovou síť pouze pro předzpracování grafů. Výstupem metody je optimální přiřazení markantů. Natrénovaný model dosahoval vysoké přesnosti i na velmi obtížných vygenerovaných párech otisků prstů. Na datové sadě, kde se vygenerované otisky lišily až o 60° a průměrný poměr zachovalých markantů byl 59% dosahovala metoda 51.2% přesnosti. Výsledek je vzhledem k počtu možných přiřazení velmi dobrý. Z přiřazení markantů jsou odhadnuty parametry transformace pomocí algoritmu RANSAC. RANSAC byl použit, protože experimenty ukázaly jeho vysokou odolnost vůči chybným přiřazením. Výsledky zarovnání jsou vzhledem k náročnosti problému dobré. Možné zlepšení by mohlo přinést zvýšení počtu atributů každého markantu otisku použitím modelu konvoluční neuronové sítě pro extrakci příznaků.

První vyzkoušený způsob porovnání otisků (verifikace) byl naivní použití metrik získaných přímo z přiřazených markantů zarovnaných otisků. Každý přiřazený pár bodů obsahuje informaci o vzdálenosti, rozdílu úhlu a souhlasnosti jejich typů. Nejlepší výsledky vykazoval systém, který učinil rozhodnutí na základě poměru správně přiřazeným markantů. U každého páru markantů přiřazení bylo rozhodnuto o jeho správnosti na základě rozdílu úhlu. Systém dosahoval výsledků $EER = 0.338$, výsledek je lepší než náhodný systém ($EER = 0.5$), nicméně pro reálné použití je tato chyba příliš velká. Samotné metriky zarovnaných otisků nemají dobrou rozlišovací schopnost. Alternativním způsobem bylo použití modelu SimGNN [3], který využívá grafovou neuronovou síť pro získání skóre podobnosti dvou grafů. Rozhodování na základě skóre podobnosti mělo lehce nižší chybovost ($EER = 0.311$) než předchozí způsob, nejlepších výsledků však dosahovala kombinace obou přístupů, skóre podobnosti a poměru správně přiřazených markantů ($EER = 0.267$). Výsledek stále není dostačující. Relativně vysoká chybovost metody SimGNN může být způsobena malým počtem atributů markantů, tento problém by opět mohla vyřešit extrakce většího počtu příznaků každého markantu.

Literatura

- [1] AL REFOA, A., ALSHRAIDEH, M. a SHARIEH, A. A New Algorithm for Locating and Extracting Minutiae from Fingerprint Images. *Pattern Recognition and Image Analysis*. Duben 2019, sv. 29, s. 268–279. DOI: 10.1134/S1054661819010036.
- [2] ALRASHIDI, A., ALOTAIBI, A., HUSSAIN, M., ALSHEHRI, H., ABOALSAMH, H. et al. Cross-Sensor Fingerprint Matching Using Siamese Network and Adversarial Learning. *Sensors*. Květen 2021, sv. 21, s. 3657. DOI: 10.3390/s21113657.
- [3] BAI, Y., DING, H., BIAN, S., CHEN, T., SUN, Y. et al. SimGNN: A Neural Network Approach to Fast Graph Similarity Computation. In: Leden 2019, s. 384–392. DOI: 10.1145/3289600.3290967. ISBN 978-1-4503-5940-5.
- [4] BEBIS, G., DEACONU, T. a GEORGIPOULOS, M. Fingerprint identification using Delaunay triangulation. In: *Proceedings 1999 International Conference on Information Intelligence and Systems (Cat. No.PR00446)*. 1999, s. 452–459. DOI: 10.1109/ICIIS.1999.810315.
- [5] CAVUSOGLU, A. a GORGUNOGLU, S. A Robust Correlation Based Fingerprint Matching Algorithm for Verification. *Journal of Applied Sciences*. prosinec 2007, sv. 7, č. 21, s. 3286–3291. DOI: 10.3923/jas.2007.3286.3291.
- [6] CHERRAT, E. M., ALAOUI, R. a BOUZAHIR, H. Convolutional Neural Networks Approach for Multimodal Biometric Identification System Using the Fusion of Fingerprint, Finger-vein and Face images. *PeerJ Computer Science*. Leden 2020. DOI: 10.7717/peerj-cs.248.
- [7] CHIKKERUR, S., CARTWRIGHT, A. a GOVINDARAJU, V. K-plet and Coupled BFS: A Graph Based Fingerprint Representation and Matching Algorithm. In: Leden 2006, s. 309–315. DOI: 10.1007/1160828842. ISBN 978-3-540-31111-9.
- [8] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L. a STEIN, C. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN 0262033844.
- [9] DAIGAVANE, A., RAVINDRAN, B. a AGGARWAL, G. Understanding Convolutions on Graphs. *Distill*. 2021. DOI: 10.23915/distill.00032. <https://distill.pub/2021/understanding-gnns>.
- [10] DEFFERRARD, M., BRESSON, X. a VANDERGHEYNST, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *CoRR*. 2016, abs/1606.09375.

- [11] DRAHANSKÝ, M., ORSÁG, F., DOLEŽEL, M. a AL. et. *Biometrie*. 1. vydání. Computer Press, s.r.o, 2011. 294 s. ISBN 978-80-254-8979-6. Dostupné z: <https://www.fit.vut.cz/research/publication/9468>.
- [12] DRAHANSKÝ, M. *Biometrické systémy – Studijní opora* [online]. 2006 [cit. 2021-12-23]. Dostupné z: https://www.fit.vutbr.cz/study/courses/BIO/private/BIO_Studijni_opora.pdf.
- [13] DRAHANSKÝ, M., BŘEZINOVÁ, E., ORSÁG, F. a HEJTMÁNKOVÁ, D. *Dermatologické faktory ovlivňující snímání otisků prstů* [online]. 2010 [cit. 2021-12-23]. Dostupné z: <https://www.mvcr.cz/mvcren/article/dermatologicke-faktory-ovlivnujici-snimani-otisku-prstu.aspx>.
- [14] FEY, M., LENSSEN, J. E., WEICHERT, F. a MÜLLER, H. SplineCNN: Fast Geometric Deep Learning with Continuous B-Spline Kernels. *CoRR*. 2017, abs/1711.08920. Dostupné z: <http://dblp.uni-trier.de/db/journals/corr/corr1711.html#abs-1711-08920>.
- [15] HAMILTON, W. L. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan and Claypool. sv. 14, č. 3, s. 1–159.
- [16] HORADAM, K., DAVIS, S., ARAKALA, A. a JEFFERS, J. Fingerprints as spatial graphs: nodes and edges. In: United States: IEEE, 2011, s. 6. ISBN 9781457720062.
- [17] ISENER, D. K. a ZAKY, S. G. Fingerprint identification using graph matching. *Pattern Recognit.* 1986, sv. 19, s. 113–122.
- [18] *Information technology — Biometric data interchange formats — Part 2: Fingerprint minutiae data* [online]. ISO, 2005 [cit. 2021-12-23]. Dostupné z: <https://www.iso.org/standard/38746.html>.
- [19] JAIN, A., HONG, L. a BOLLE, R. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1997, sv. 19, č. 4, s. 302–314. DOI: 10.1109/34.587996.
- [20] KIPF, T. N. a WELLING, M. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR*. 2016, abs/1609.02907.
- [21] KOTADIA, Y. *Comparing Message Passing Based GNN Architectures* [online]. WeightsBiases, 2020 [cit. 2022-1-7]. Dostupné z: <https://wandb.ai/yashkotadia/benchmarking-gnns/reports/Part-2-Comparing-Message-Passing-Based-GNN-Architectures>.
- [22] MALTONI, D., MAIO, D., JAIN, A. K. a PRABHAKAR, S. Handbook of Fingerprint Recognition. *Springer Professional Computing*. 2003.
- [23] MEHTRE, B. M. Fingerprint image analysis for automatic identification. *Machine Vision and Applications*. Mar 1993, sv. 6, č. 2, s. 124–139. DOI: 10.1007/BF01211936. ISSN 1432-1769. Dostupné z: <https://doi.org/10.1007/BF01211936>.
- [24] MUÑOZ BRISEÑO, A., GAGO ALONSO, A. a HERNÁNDEZ PALANCAR, J. Fingerprint Matching Using a Geometric Subgraph Mining Approach. In: Listopad 2015. DOI: 10.1007/978-3-319-25751-8_20. ISBN 978-3-319-25750-1.

- [25] ÖLZ, W. a KROPATSCH, W. G. Graph Representation of Fingerprint Topology . In: Computer Vision, 2004, s. 51–58.
- [26] RAMÍREZ FLORES, M., AGUILAR TORRES, G. a GALLEGOS GARCÍA, G. Fingerprint verification using computational geometry. *Dyna*. 2016, sv. 83, č. 195, s. 128–137.
- [27] RATHA, N., BOLLE, R., PANDIT, V. a VAISH, V. Robust fingerprint authentication using local structural similarity. In: únor 2000, s. 29–34. DOI: 10.1109/WACV.2000.895399. ISBN 0-7695-0813-8.
- [28] RATHA, N., KARU, K., CHEN, S. a JAIN, A. A real-time matching system for large fingerprint databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1996, sv. 18, č. 8, s. 799–813. DOI: 10.1109/34.531800.
- [29] RAVI, J., K B, R. a K R, V. Fingerprint Recognition Using Minutia Score Matching. *CoRR*. Leden 2010, abs/1001.4186.
- [30] ROLÍNEK, M., SWOBODA, P., ZIETLOW, D., PAULUS, A., MUSIL, V. et al. Deep Graph Matching via Blackbox Differentiation of Combinatorial Solvers. *CoRR*. 2020, abs/2003.11657.
- [31] SANCHEZ LENGELING, B., REIF, E., PEARCE, A. a WILTSCHKO, A. B. A Gentle Introduction to Graph Neural Networks. *Distill*. 2021. DOI: 10.23915/distill.00033. <https://distill.pub/2021/gnn-intro>.
- [32] SOCHER, R., CHEN, D., MANNING, C. D. a NG, A. Y. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In: *NIPS*. 2013, s. 926–934. Dostupné z: <http://dblp.uni-trier.de/db/conf/nips/nips2013.html#SocherCMN13>.
- [33] THAKKAR, D. *Minutiae Based Extraction in Fingerprint Recognition* [online]. Bayometric, 2010 [cit. 2021-12-23]. Dostupné z: <https://www.bayometric.com/minutiae-based-extraction-fingerprint-recognition/>.
- [34] VAŽAN, R. *ISO/IEC 19794-2:2005/Cor.1:2009 Summary* [online]. 2005 [cit. 2021-12-23]. Dostupné z: <https://templates.machinezoo.com/iso-19794-2-2005>.
- [35] WANG, T., LIU, H., LI, Y., JIN, Y., HOU, X. et al. Learning Combinatorial Solver for Graph Matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.
- [36] WU, Z., PAN, S., CHEN, F., LONG, G., ZHANG, C. et al. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*. Institute of Electrical and Electronics Engineers (IEEE). Jan 2021, sv. 32, č. 1, s. 4–24. DOI: 10.1109/tnnls.2020.2978386. ISSN 2162-2388. Dostupné z: <http://dx.doi.org/10.1109/TNNLS.2020.2978386>.
- [37] XIAO GUANG, C., LIN, W. a MINGYAN, L. An Efficient Graph-Based Algorithm for Fingerprint Representation and Matching. Listopad 2013. DOI: 10.2991/icmt-13.2013.125.
- [38] YANG, W., HU, J. a WANG, S. A Delaunay Quadrangle-Based Fingerprint Authentication System With Template Protection Using Topology Code for Local Registration and Security Enhancement. *IEEE Transactions on Information Forensics and Security*. 2014, sv. 9, s. 1179–1192.

- [39] ZACHARIAS, G. C., NAIR, M. S. a SOJAN LAL, P. Pre- and Post-fingerprint Skeleton Enhancement for Minutiae Extraction. In: RAMAN, B., KUMAR, S., ROY, P. P. a SEN, D., ed. *Proceedings of International Conference on Computer Vision and Image Processing*. Singapore: Springer Singapore, 2017, s. 453–465. ISBN 978-981-10-2104-6.

Příloha A

Obsah příloženého paměťového média

Adresářová struktura příloženého paměťového média:

- `src/` - adresář se zdrojovými soubory navrhovaného řešení
 - `BB_GM/` - zdrojové soubory metody *Deep Graph Matching*
 - `data/` - skripty pro načtení a zpracování dat
 - `experiments/` - spouštěcí konfigurace metody *DGM*
 - `results/` - výstup metody *DGM*
 - `utils/` - pomocné skripty metody *DGM*
 - `gnn/` - zdrojové soubory metody *SimGNN*
 - `models/` - natrénované modely *DGM* a *SimGNN*
 - `README.md` - podrobnější popis struktury adresáře zdrojových textů, včetně instalace a spuštění jednotlivých trénovacích a vyhodnocovacích skriptů
- `video/` - adresář s videem
- `text/` - adresář obsahující text práce včetně zdrojových souborů