VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# VYHLEDÁVÁNÍ FOTOGRAFIÍ V DATABÁZI PODLE PŘÍKLADU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MATÚŠ DOBROTKA
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# VYHLEDÁVÁNÍ FOTOGRAFIÍ V DATABÁZI PODLE PŘÍKLADU

IMAGE DATABASE QUERY BY EXAMPLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MATÚŠ DOBROTKA
AUTHOR

VEDOUCÍ PRÁCE — Ing. MARTIN VEĽAS
SUPERVISOR

BRNO 2015

# Abstrakt

Táto práce se zabývá vyhledáváním fotografií v databázi podle příkladu. Práce byla vytvořena s cílem vyvinout aplikaci, která bude porovnávat různé přístupy vyhledávání fotografií v databázi. Jedná se o základní přístup, který spočívá v detekci klíčových bodů, extrakci lokálních příznaků a tvorbě vizuálního slovníku algoritmem shlukování - k-means. Pomocí vizuálního slovníku je spočítán histogram četnosti výskytu vizuálních slov - Bag of Words (BoW), který reprezentuje fotografii jako celek. Po aplikování vhodné metriky dojde k vyhledání podobných fotografií. Druhý přístup představují hluboké konvoluční neuronové sítě (DCNN), které jsou využity k extrakci příznakových vektorů. Tyto vektory jsou použity na tvorbu vizuálního slovníku, který slouží opět k výpočtu BoW. Postup je pak podobný jako v prvním přístupu. Třetí přístup počítá s extrahovanými vektory z DCNN jako s BoW vektory. Následuje aplikace vhodné metriky a vyhledání podobných fotografií. V závěru práce jsou popsány použity přístupy, uvedeny experimenty a závěrečné vyhodnocení.



# Abstract

This thesis deals with content-based image retrieval. The objective of the thesis is to develop an application, which will compare different approaches of image retrieval. First basic approach consists of keypoints detection, local features extraction and creating a visual vocabulary by clustering algorithm - k-means. Using this visual vocabulary is computed histogram of occurrence count of visual words - Bag of Words (BoW), which globally represents an image. After applying an appropriate metrics, it follows finding similar images. Second approach uses deep convolutional neural networks (DCNN) to extract feature vectors. These vectors are used to create a visual vocabulary, which is used to calculate BoW. Next procedure is then similar to the first approach. Third approach uses extracted vectors from DCNN as BoW vectors. It is followed by applying an appropriate metrics and finding similar images. The conclusion describes mentioned approaches, experiments and the final evaluation.



# Klíčová slova

vyhledávání fotografií, klíčový bod, SIFT, lokální příznaky, vizuální slovník, bag of words, shlukování, k-means, neuronové sítě, hluboké konvoluční neuronové sítě, konvoluce

# Keywords

image retrieval, keypoint, SIFT, local features, visual vocabulary, bag of words, clustering, k-means, neural networks, deep convolutional neural networks, convolution

# Citace

Matúš Dobrotka: Vyhledávání fotografií v databázi podle příkladu, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Vyhledávání fotografií v databázi podle příkladu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Martina Veľase. Veškeré zdroje, ze kterých jsem při tvorbě této práce čerpal, jsem uvedl v seznamu použitých zdrojů.

.....
Matúš Dobrotka
20. května 2015

## Poděkování

Chci se poděkovat svému vedoucímu, Ing. Martinu Veľasovi, za odbornou pomoc, cenné rady, věnovaný čas i trpělivost při řešení této bakalářské práce a také Ing. Michalovi Hradišovi za pomoc a vstřícnost při práci s neuronovými sítěmi.

© Matúš Dobrotka, 2015.
*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah


**1 Úvod** **2**

**2 Vyhľadávanie v obrazovej databáze** **3**
2.1 Detekcia kľúčových bodov a extrakcia príznakov z obrazu . . . . . 3
2.2 SIFT . . . . . 4
2.3 Vizuálny slovník - Visual Vocabulary . . . . . 4
2.4 Histogram početnosti výskytu slov - Bag of Words (BoW) . . . . . 9
2.5 Metriky podobností fotografií . . . . . 10
2.6 Hlboké konvolučné neurónové siete . . . . . 11
2.7 Databáza fotografií . . . . . 14

**3 State of the art** **17**
3.1 Konvenčný postup a metódy vyhľadávania v databázach fotografií . . . . . 17
3.2 Deep Convolutional Neural Networks . . . . . 18

**4 Návrh aplikácie** **23**
4.1 Ciele . . . . . 23
4.2 Návrh riešenia . . . . . 23

**5 Implementácia** **26**
5.1 Knižnica OpenCV . . . . . 26
5.2 Popis aplikácie . . . . . 27

**6 Vyhodnotenie** **29**
6.1 Základné riešenie . . . . . 29
6.2 Deep Convolutional Neural Networks . . . . . 30
6.3 Návrh testovania . . . . . 31

**7 Záver** **34**

# Kapitola 1

# Úvod

Táto práca sa zaoberá problematikou vyhľadávania v obrazových databázach podľa príkladu na základe podobnosti. Príklad reprezentuje vzorová fotografia. Cieľom je navrhnúť a implementovať systém na vyhľadávanie obrázkov (*Image Retrieval System*). Tento systém si môžme na úvod predstaviť ako čiernu skrinku, ktorej vstupom bude vzorová fotografia a výstupom zase fotografia z databáze. Výstupná fotografia je zo všetkých fotografií v databáze najviac podobná vzorovej fotografii na vstupe.

Takýto systém je možné v dnešnej dobe vytvoriť rôznymi spôsobmi. V tejto práci bude prezentovaná jedna z najzákladnejších metód, ktoré sa v tejto problematike využívajú – metóda využívajúca vizuálny slovník (visual vocabulary) a histogramy početnosti (Bag of Words) na výpočet a zistenie podobností fotografií. Druhá metóda využíva konvolučné neurónové siete (Deep Convolutional Neural Networks) na extrakciu lokálnych príznakov fotografií. Prostredníctvom extrahovaných príznakov totiž môžeme spočítať, ktoré fotografie sú si ako podobné. Konvolučné neurónové siete používané v tejto oblasti sú oproti spomínanej prvej metóde aktuálnejšie. Hlavným cieľom celej práce je porovnať rôzne prístupy vyhľadávania fotografií. Výsledky, ktoré tieto spomínané metódy dosiahnu, budú navzájom porovnávané. Do porovnania budú zahrnuté nielen vlastné výsledky tejto práce, ale aj výsledky state-of-the-art experimentov. Výsledky vo forme priemernej presnosti (anglicky *average precision*) budú vyhodnotené voľne dostupným C++ programom.

Nasledujúca kapitola *Vyhľadávanie v obrazovej databáze* obsahuje teoretický základ zvolených riešení – tak popis rôznych techník dôležitých pre vytvorenie aplikácie, ako aj zhrnutie faktov a ich súvis a zapracovanie do daných techník. V kapitole 3 - *State of the art* je možné nájsť informácie ohľadom súčasných trendov, aké postupy a metódy sa využívajú v dnešnej dobe v oblasti vyhľadávania obrázkov v databáze. Kapitola 4 - *Návrh aplikácie* zahŕňa informácie o architektúre celého navrhovaného systému. Veľmi podstatná časť, ktorá je tiež v tejto kapitole obsiahnutá, je i návrh testovacích dat a programov, na základe ktorých je overená činnosť samotnej aplikácie. Čo sa týka použitých technológií, algoritmov, knižníc, či konkrétnej implementácie, tie sa nachádzajú v kapitole 5 - *Implementácia.* Kapitola 6 - *Vyhodnotenie* informuje čitateľa o dosiahnutých výsledkoch z experimentovania a ukazuje rozdiely a odchýlky v jednotlivých prístupoch riešenia. Záverečná kapitola 7 - *Záver* sumarizuje výsledky všetkých vykonaných experimentov, testovaní i optimalizácií a takisto sa venuje zhodnoteniu celej práce, aké zistenia priniesla a aké závery z toho plynú.

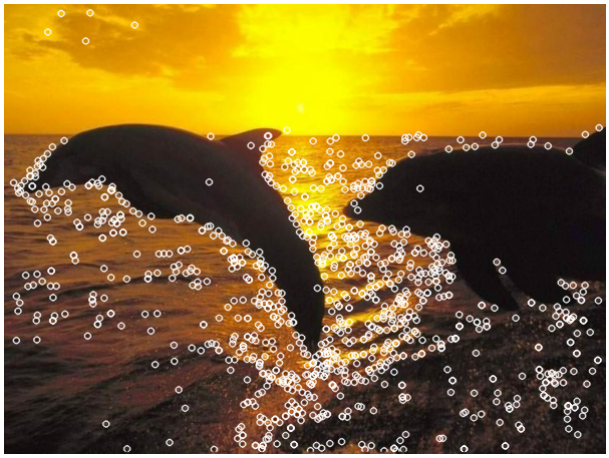
# Kapitola 2

# Vyhľadávanie v obrazovej databáze

Proces vyhľadávania fotografií v obrazovej databáze spočíva v niekoľkých krokoch, ktoré budú postupne predstavené v tejto kapitole.

## 2.1 Detekcia kľúčových bodov a extrakcia príznakov z obrazu

Na to, aby sme mohli vyhľadávať fotografie v obrazovej databáze podľa príkladu, je najprv nutné detekovať kľúčové body (keypoints) každej fotografie. Aj vstupnej (vzorovej) ako aj tých, ktoré sú v databáze. Na detekciu v tejto práci bude využívaný detektor Difference of Gaussians (DoG) v rámci metódy SIFT[18]. Zameriava sa na detekciu rohov a blobov v obraze. Pri detektore je veľmi dôležité, aby spĺňal vlastnosť opakovateľnosti (repeatability). To znamená, aby dokázal detekovať rovnaké kľúčové body u dvoch alebo viacerých obrázkov, ktoré zachytávajú rovnakú scénu, avšak navzájom sa líšia uhľom pohľadu na túto scénu.



Obr. 2.1: Ukážka detekovaných kľúčových bodov prostredníctvom SIFT detektoru DoG

Po tom, čo sú v obrázku detekované kľúčové body, nasleduje popis každého z týchto bodov a jeho okolia prostredníctvom lokálnych príznakov metódy SIFT (podrobný popis tejto metódy zahŕňa nasledujúca podkapitola). Okrem príznakov SIFT poznáme viac druhov

lokálnych príznakov, napr. SURF, ORB, FAST, BRIEF, BRISK, ... [19] Tieto lokálne príznaky popisujú body obrazu a tým sa odlišujú od globálnych, ktoré popisujú obrázok ako celok. Medzi globálne príznaky patrí napr. histogram početnosti slov vo fotografii, známy aj pod názvom Bag of Words (BoW) vektor. O histogramoch početnosti však bude reč v jednej z nasledujúcich častí dokumentu. Každý kľúčový bod spolu so svojim okolím teda bude reprezentovaný *príznakovým vektorom,* čo je číselný vektor. Tiež sa nazýva pojmom *deskriptor.*

Je veľmi dôležité, aby bol detektor spolu s deskriptorom čo najviac invariantný k základným lineárnym transformáciam ako napr. zmena uhlu zachytenia objektu, natočenie, či zmena vzdialenosti pozorovateľa od objektu. Taktiež invariantnosť k fotometrickým zmenám sa podpisuje pod kvalitu popisu príznaku.

## 2.2 SIFT

SIFT (*Scale Invariant Feature Transform*) sa radí medzi lokálne obrazové príznaky. Na rozdiel od metódy SURF, ktorá v sebe zahŕňa tak detektor ako aj deskriptor, SIFT vystupuje iba v roli deskriptora. Ako bolo ale spomenuté vyššie, metóda SIFT využíva detektor DoG[18].

SIFT deskriptory počítajú histogram lokálne orientovaných gradientov okolo kľúčových bodov. Jednotlivé intervaly ukladajú do 128-dimenzionálneho vektora - 8 orientačných intervalov pre každý zo 4x4 lokálnych intervalov [3].

Táto metóda transformuje obrázok na veľkú množinu príznakových vektorov (označujú sa tiež ako *SIFT keys*), ktoré sú invariantné voči posunu a otočeniu obrázku a zmene mierky. Invariantnosť voči zmene mierky je špeciálnou vlastnosťou SIFT deskriptorov, ktorá sa nevyskytuje pri iných príznakoch (ako napr. SURF). Taktiež výhodou SIFT deskriptorov je aj to, že nie sú tak citlivé na projektívnu deformáciu a zmenu osvetlenia ako je to pri iných deskriptoroch. To znamená, že sú čiastočne invariantné aj voči zmenám osvetlenia a afinnej či 3D projekcii. Čiastočná invariancia je dosiahnutá rozmazávaním prechodových oblastí obrázka [18].

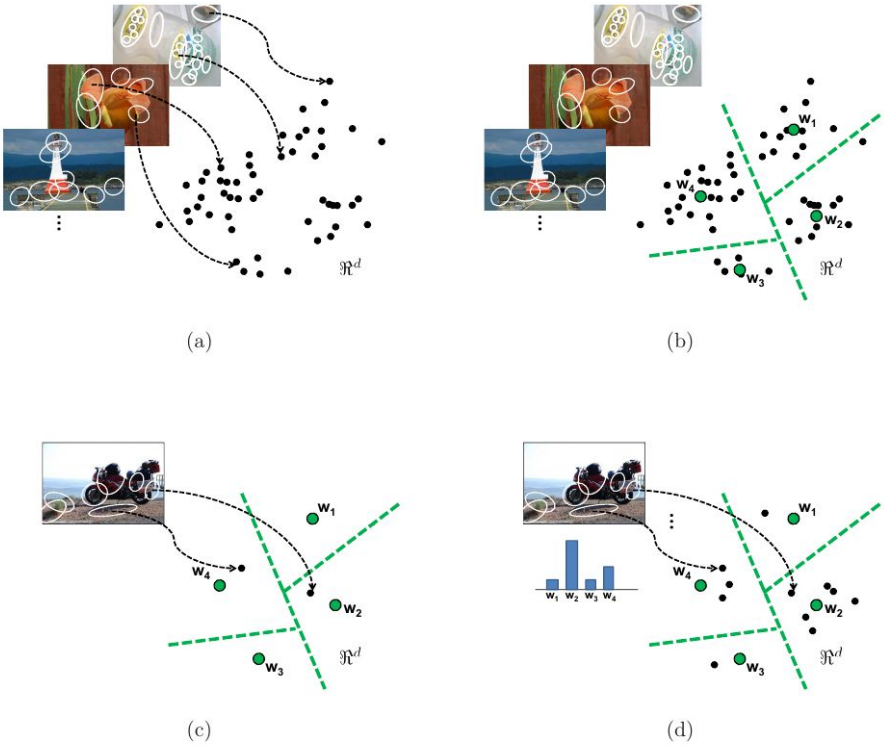
## 2.3 Vizuálny slovník - Visual Vocabulary

V tejto kapitole bude predstavená koncepcia vizuálneho slovníka - stratégia, ktorá čerpá inšpiráciu z prehľadávania textu. Najskôr popíšem tvorbu vizuálnych slov a potom v nasledujúcej kapitole reprezentáciu obrazu.

Vizuálne slovníky možno vnímať ako paralelu k slovníkom lexikografickým. Sú však tvorené vizuálnymi slovami, ktoré sa od tradičných lexikografických slov odlišujú. Sú to číselné vektory obsahujúce informácie o obsahu fotografie. Deskriptory kľúčových bodov sa prostredníctvom algoritmu (kap. 2.3.1) nazhlukujú. Tým sa vytvoria zhluky deskriptorov, pričom každý zhluk má svoj vlastný stred. Vizuálne slová predstavujú práve tieto centrá zhlukov.[7].

Postup práce s vizuálnymi slovníkmi je podobný ako u textových dokumentov. Ak je potrebné analyzovať text v rámci textového dokumentu, tak je nutné slová nachádzajúce sa v slovníku nájsť v dokumente a sústrediť sa napríklad na ich výskyt, resp. počet.

Ak analyzujeme fotografiu a jej obsah, sledujeme výskyt vizuálnych slov z vizuálneho slovníka v tejto fotografii. Niektoré vizuálne slová sa v danej fotografii môžu nachádzať viackrát, iné zase vôbec. Obsah dvoch porovnávaných fotografií sa vo väčšine prípadov




Obr. 2.2: **Schéma tvorby vizuálneho slovníka a priraďovania slov.** (a) Väčšie množstvo reprezentatívnych obrázkov obsadí priestor príznakov s inštanciami deskriptorov. Biele elipsy vyznačujú oblasti lokálnych príznakov a čierne body predstavujúce SIFT príznaky. (b) Navzorkované príznaky sú zhlukované za účelom kvantovania priestoru do diskrétneho počtu vizuálnych slov. Vizuálne slová sú centrá zhlukov, na obrázku označené ako väčšie zelené krúžky. (c) Teraz je daný nový obrázok. Pre každý z jeho príznakov je identifikované najbližšie vizuálne slovo. Toto mapuje obrázok z množiny vysoko-dimenzionálnych deskriptorov na zoznam počtov slov. (d) Histogram početnosti výskytu vizuálnych slov môže byť použitý na sumarizáciu celého obrázka. Výpočet spočíva v tom, koľkokrát sa každé z vizuálnych slov vyskytuje v obrázku.[10]

navzájom aspoň do istej miery odlišuje, z toho dôvodu sa líšia aj deskriptory týchto fotografií.

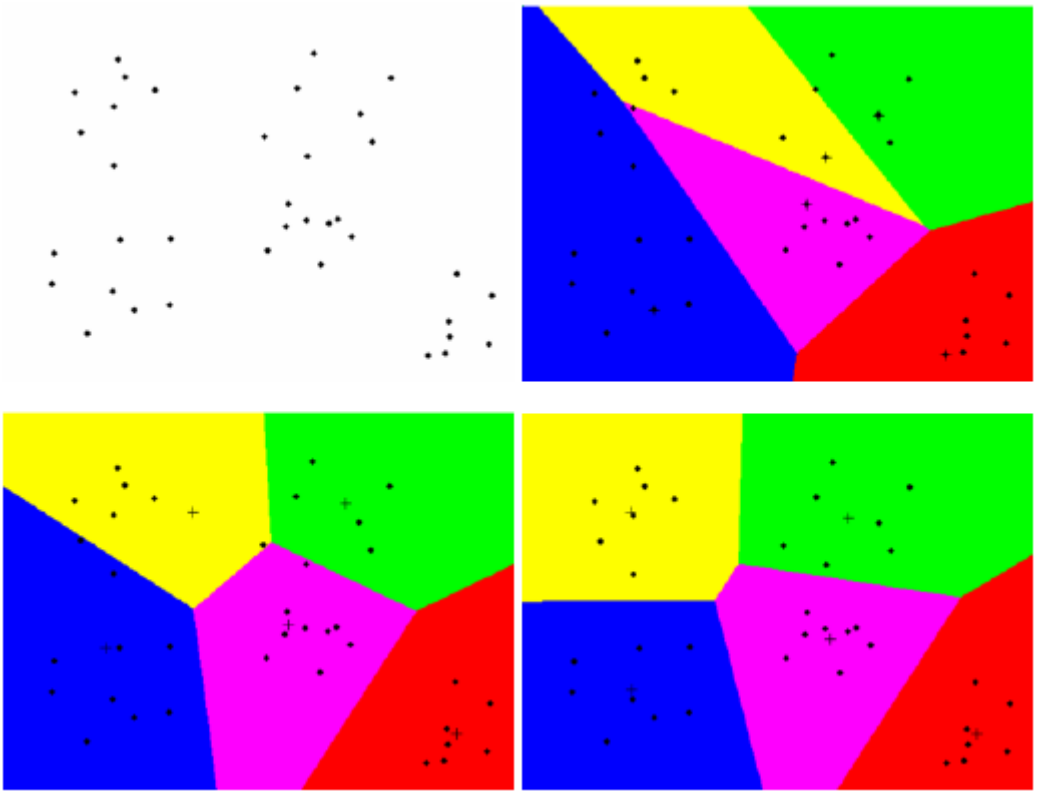
Bolo by neprípustné vytvárať vizuálny slovník zo všetkých vizuálnych slov získaných zo všetkých fotografií. Deskriptory podobných fotografií môžu byť podobné. Je teda rozumné vytvoriť vizuálny slovník takých vizuálnych slov, ktoré by zachytávali dostatočné množstvo obrazových dat, aby bola dosiahnutá dostatočná rozlíšiteľnosť.

Pri tvorbe vizuálneho slovníku sa využívajú algoritmy pre zhlukovanie, ktorých princípom je to, že pre každú skupinu veľmi podobných deskriptorov sa vypočíta jeden centrálny stred zhluku, ktorý bude reprezentovať všetky deskriptory tejto skupiny vo vizuálnom slovníku, čiže v konečnom dôsledku to bude vizuálne slovo zastupujúce všetky tieto veľmi podobné deskriptory. Zhlukovaniu sa bude podrobnejšie venovať nasledujúca kapitola.

### 2.3.1 K-means clustering

Algoritmus zhlukovania *k-means clustering* [24] spadá do kategórie strojového učenia bez učiteľa (anglicky *unsupervised learning*). Toto učenie spočíva v hľadaní podobností medzi príkladmi trénovacej množiny a v zaraďovaní (klasifikácii) príkladov s podobnými charakteristikami do skupín - zhlukov. Umelý systém pritom nedostáva žiadnu informáciu o správnosti klasifikácie. A tým pádom ani o priebehu svojho učenia. Jedinou informáciou môže byť počet zhlukov, do ktorých sa majú príklady z trénovacej množiny klasifikovať. Tieto príklady trénovacej množiny sú reprezentované číselnými vektormi.

Algoritmus *k-means clustering* patrí medzi najznámejšie a najpoužívanejšie algoritmy. Klasifikuje príklady (číselné vektory) trénovacej množiny do vopred daného počtu $k$ zhlukov. Algoritmus zaradí každý vektor do toho zhluku, ktorého stred je k danému vektoru najbližšie. Samotné učenie spočíva v nasledujúcich krokoch (obr. 2.3):



Obr. 2.3: **Učenie algoritmom k-means clustering.** *Vľavo hore* je zobrazená trénovacia množina dvojrozmerných číselných vektorov, *vpravo hore* možno vidieť počiatočnú náhodnú voľbu piatich stredov a následné rozdelenie vektorov do zhlukov, *vľavo dole* sa nachádza stav počas učenia a *vpravo dole* je výsledok po ukončení zhlukovania. [24]

- Algoritmus spočiatku náhodne vyberie $k$ vektorov z trénovacej množiny a považuje ich za stredy zhlukov. Keďže náhodný výber stredov nepatrí medzi výhody tohto algoritmu, napríklad autori *Arthur* a *Vassilvitskii* prišli s alternatívnym algoritmom *k-means++,* ktorý náhodný výber nevyužíva.
- Potom priradí všetky vektory trénovacej množiny k príslušným zhlukom.

- Nasleduje prepočítanie stredov zhlukov a znovupriradenie všetkých vektorov trénovacej množiny k príslušným zhlukom.
- Učenie končí vtedy, keď všetky vektory priraďuje do stále rovnakých zhlukov. Ďalšou variantou ukončenia algoritmu môže byť aj dosiahnutie požadovaného maximálneho počtu iterácií alebo určitej konkrétnej presnosti.[24, 1]

**Algoritmus zhlukovania[24]:**

1. Inicializuj $k$ stredov (náhodne vybratých, ale rôznych vektorov z trénovacej množiny o mohutnosti $P$: $\overrightarrow{w}_j = \overrightarrow{x}_p$, $j \in \langle 1, k \rangle$, $p \in \langle 1, P \rangle$).
2. Každý vektor $\overrightarrow{x}_p$ z trénovacej množiny priraď do zhluku $C_j$ $j \in \langle 1, k \rangle$, ktorého stred $\overrightarrow{w}_j$ má od vektoru $\overrightarrow{x}_p$ najmenšiu vzdialenosť:
$$|\overrightarrow{x}_p - \overrightarrow{w}_j| \leq |\overrightarrow{x}_p - \overrightarrow{w}_i| \qquad i, j \in \langle 1, k \rangle \tag{2.1}$$
3. Pre každý zhluk $C_j$ $j \in \langle 1, k \rangle$ prepočítaj stred $\overrightarrow{w}_j$ tak, aby bol ťažiskom koncových bodov všetkých vektorov, ktoré sú k tomuto zhluku práve priradené (nech $n_j$ je počet týchto vektorov):
$$\overrightarrow{w}_j = \frac{\sum_{\overrightarrow{x}_i \in C_j} \overrightarrow{x}_i}{n_j} \tag{2.2}$$
4. Spočítaj „chybu" aktuálneho stavu zhlukovania (je to súčet„chýb" všetkých zhlukov, ktoré sú dané súčtami druhých mocnín vzdialeností všetkých vektorov jednotlivých zhlukov od stredov týchto zhlukov):
$$E = \sum_{j=1}^{k} \sum_{\overrightarrow{x}_i \in C_j} |\overrightarrow{x}_i - \overrightarrow{w}_j|^2 \tag{2.3}$$
5. Pokiaľ „chyba" $E$ klesla, alebo pokiaľ bol niektorý vektor priradený k inému zhluku, vráť sa späť na bod 2. Ďalším kritériom ukončenia býva dosiahnutie určitej presnosti zhlukovania, čiže dostatočne malá „chyba". Iným ukončovacím kritériom môže byť špecifikovaný maximálny počet iterácií, ktoré má algoritmus vykonať.

**Výhody(+) a nevýhody(-) algoritmu:**[13, 23]

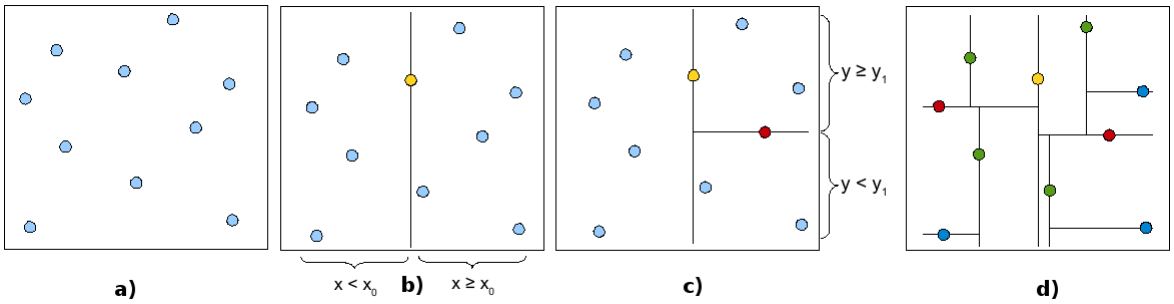
+ Rýchly a jednoduchší na porozumenie.
+ Relatívne rýchly: $O(tknd)$, pričom $n$ je počet objektov, $k$ je počet zhlukov, $d$ je dimenzia každého objektu a $j$ je počet iterácií. Spravidla $k, t, d \ll$ n.
+ Najlepšie výsledky dáva vtedy, keď sú dáta rozdielne alebo dobre oddelené od seba navzájom.

- Algoritmus vyžaduje špecifikovať počet centier jednotlivých zhlukov.
- Ak dochádza k vysokému prekrytiu medzi dvoma dátami, tak k-means nie je schopný rozlíšiť, že sa jedná o dva zhluky.
- Algoritmus nie je invariantný k nelineárnym transformáciam, napr. s odlišnými reprezentáciami dát dostávame odlišné výsledky (dáta reprezentované formou karteziánskych súradníc a polárnych súradníc dávajú odlišný výsledok).
- Náhodný výber centier zhlukov taktiež nemôže viesť k ideálnym výsledkom.
- Algoritmus je neúspešný pre nelineárne dáta.

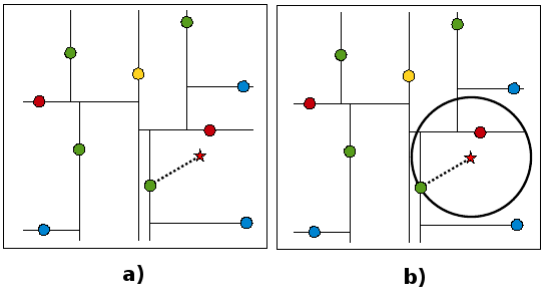
### 2.3.2 Kd-tree

V momente, keď sú extrahované lokálne príznaky z obrázkov, a taktiež je vytvorený vizuálny slovník z vizuálnych slov prostredníctvom algoritmu zhlukovania, možno pristúpiť k ďalšej fáze vyhľadávania fotografií. Vizuálny slovník je treba prehľadávať a zisťovať, či sa to-ktoré vizuálne slovo v danej fotografii nachádza alebo nie, aby bolo možné túto fotografiu korektne popísať.

Tento slovník predstavuje datová štruktúra - *k-dimenzionálny strom.* Jedná sa o multidimenzionálny binárny vyhľadávací strom (angl. $k$-d tree)[4], kde $k$ označuje dimenzionalitu priestoru, ktorý sa prehľadáva. Slúži na uloženie informácií, ktoré vyžadujú asociatívne vyhľadávanie v rámci určitej kolekcie záznamov. Takýto záznam si možno predstaviť ako usporiadanú $k$-ticu $(v_1, v_2, ..., v_k)$ hodnôt. Tieto hodnoty môžu byť buď kľúčmi alebo atribútmi záznamu.




Obr. 2.4: Vytvorenie *kd*-tree zo zbierky dátových bodov v $k$-dimenzionálnom priestore pre $k = 2$ a . *Pozn.:* Žltý bod je koreňový uzol, uzly 1. úrovne zanorenia sú červené, 2. úroveň - zelené uzly, 3. úroveň - modré uzly. [6].




Obr. 2.5: Vyhľadávanie najbližšieho suseda k testovaciemu bodu, ktorý je označený hviezdičkou. Je daný kd-strom a bod v priestore (nazývaný testovací bod). Ktorý bod je najbližšie k testovaciemu? Najbližším susedom rozumieme bod v datasete, ktorý je najbližšie testovaciemu bodu. Predpokladajme, že tušíme, ktorý bod by mohol byť najbližším susedom k testovaciemu bodu. Predpokladajme, že to je bod spojený s testovacím bodom úsečkou. Na overenie môžme použiť úvahu, že ak sa v datasete nachádza bod, ktorý je k testovaciemu bližšie ako náš aktuálny tip, tak tento najbližší sused musí ležať v kruhu so stredom v testovacom bode, ktorý prechádza našim odhadovaným najbližším susedom. V prípade 2D je daná oblasť kruhom, v 3D priestore by to bola guľa. [6].

Požiadavka na vyhľadávanie v kolekcii záznamov sa nazýva *dopyt - query*[4]. Dopyt presne stanovuje určité podmienky, ktorých splnenie by mal zabezpečiť vyhľadávaný záznam. Významnou výhodou $k$-dimenzionálneho stromu je fakt, že jediná dátová štruktúra môže veľmi efektívne pracovať s viacerými typmi dopytov. Podľa toho, aké podmienky ten-ktorý dopyt určuje, môžme uvažovať o dopytoch na prienik (*intersection queries*), do ktorých spadajú aj dopyty na určitú oblasť (*region queries*), ďalej ich delíme na dopyty na presnú alebo čiastočnú zhodu (*exact/partial match queries*) a dopyty na najbližšieho suseda (*nearest neighbor queries*).

V tejto práci sa budem zaujímať práve o posledný uvedený typ dopytov. Budú hľadaní najbližší susedia v $k$-dimenzionálnom strome - vo vizuálnom slovníku. Našou úlohou bude nájsť najbližšie, čiže zároveň i najpodobnejšie slovo k extrahovanému príznakovému vektoru fotografií. Ako spomína J.L.Bentley [4], pre $n$ záznamov má hľadanie najbližšieho suseda pomocou $k$-d tree empiricky vyjadrenú priemernú časovú zložitosť $O(\log n)$.

Ako každý binárny strom, aj $k$-dimenzionálny strom obsahuje uzly. Každý uzol obsahuje nasledujúce časti[5]:

- 2 ukazatele na synovské uzly
- vyhľadávací kľúč - buď jedno desatinné číslo reprezentujúce hodnotu súradnice, alebo viacero takýchto čísel reprezentujúcich umiestnenie záznamu v priestore
- sprievodné informácie (napr. meno záznamu)

Na obr. 2.4 možno vidieť postup tvorby $k$-dimenzionálneho stromu pre $k = 2$. Konštrukcia tohto $k$-dimenzionálneho stromu spočíva v niekoľkých krokoch. Najskôr dôjde k výberu základu (2.4 a) - jedného z rozmerov priestoru vybratého bodu, ktorý rozdelí ostatné body na 2 skupiny. Takže napríklad pre koreňový uzol vyberieme $x$-ovú os ako základ (2.4 b). Všetky body, ktoré sa nachádzajú vľavo od koreňového uzlu, majú $x$-ovú súradnicu menšiu ako koreň. Naopak, všetky tie body, ktoré sa nachádzajú vpravo od koreňového uzlu, majú svoju $x$-ovú súradnicu väčšiu alebo rovnú ako koreň. Pokračujeme výberom základu pre synovské uzly koreňa. Tentokrát vyberieme $y$-ovú os ako základ (2.4 c). Obdobným spôsobom postupujeme aj ďalej, takže pre synovské uzly koreňových synovských uzlov vyberieme ako základ znovu $x$-ovú os atď. (2.4 d).

Majme funkciu vzdialenosti $D$, kolekciu záznamov $B$ v $k$-dimenzionálnom priestore a bod $P$ v tomto priestore. Pre najbližšieho suseda $Q$ bodu $P$ potom bude platiť

$$(\forall R \in B): \quad \{(R \neq Q) \Rightarrow [D(R,P) \geq D(Q,P)]\} \tag{2.4}$$

## 2.4 Histogram početnosti výskytu slov - Bag of Words (BoW)

Pri vyhľadávaní podobných fotografií je potrebné každú fotografiu reprezentovať číselným vektorom, ktorý bude zachytávať informáciu, koľkokrát sa každé vizuálne slovo z vizuálneho slovníka v danej fotografii vyskytuje (obr. 2.2 d). Tento číselný vektor vyjadruje histogram početnosti výskytu vizuálnych slov v danej fotografii, ktorý je vo svete počítačového videnia známy pod pojmom *Bag of Words* (*BoW*).

O tejto reprezentácii fotografie je vhodné spomenúť, že prekladá (obvykle veľmi veľkú) množinu vysoko-dimenzionálnych lokálnych deskriptorov do jedného riedkeho vektoru fixnej dimenzionality naprieč všetkými fotografiami [10].

Nedostatok geometrie v reprezentácii Bag of Words môže byť potenciálne aj výhodou, ale taktiež i nevýhodou. Na jednej strane, zakódovaním iba výskytu vzhľadu lokálnych oblastí, nie ich relatívnej geometrie, dostávame významnú flexibilitu na bod pohľadu a zmeny pozície. Na druhej strane, geometria medzi príznakmi môže sama byť dôležitým rozhodovacím faktorom, ktorý však v BoW reprezentácii chýba. Príznaky sa v skutočnosti extrahujú takým spôsobom, že sa navzájom do istej miery prekrývajú. Toto poskytuje prinajmenšom isté implicitné geometrické závislosti medzi deskriptormi.

Keď je BoW vektor extrahovaný z celej fotografie, príznaky z popredia sa zmiešajú s príznakmi z pozadia. Toto môže byť v konečnom dôsledku problematické, pretože príznaky z pozadia „znečistia" skutočný vzhľad objektu. Na zmiernenie tohto aspektu možno napríklad vytvoriť jeden BoW z každej segmentovanej oblasti fotografie (viacnásobné segmentácie).

Napriek viacerým výhodám, ktoré prinášajú vizuálne slová ako prostriedky pre rozpoznávanie, optimálna tvorba vizuálneho slovníka zostáva nejasná. Reálne slová sú diskrétne a človekom definované konštrukty, ale vizuálny svet je spojitý, kontinuálny a prináša komplexné prirodzené fotografie. Reálne vety majú 1D štruktúru, zatiaľ čo fotografie sú 2D projekcie 3D sveta. Je preto potrebný ďalší výskum v oblasti, aby sme lepšie porozumeli vykonaným voľbám pri konštrukcii slovníkov pre lokálne príznaky [10].

## 2.5 Metriky podobností fotografií

Na to, aby sme zistili, ktoré fotografie sú si navzájom ako podobné, musíme použiť určitú metriku. Fotografie sú reprezentované číselnými vektormi (BoW), ako bolo spomenuté v predošlej kapitole, takže hľadať podobnosť medzi fotografiami znamená v konečnom dôsledku hľadať podobnosť medzi číselnými vektormi.

V oblasti vyhľadávania fotografií sa podľa [25] na základe efektivity a presnosti vyhľadávania najlepšie prejavili tieto metriky (pre všetky metriky platí, že $Q = \{Q_0, Q_1, \ldots, Q_{N-1}\}$ a $T = \{T_0, T_1, \ldots, T_{N-1}\}$ predstavujú dopytovacie (*query*) a cieľové (*target*) vektory príznakov):

- ***Chí-kvadrát test dobrej zhody*** - ($\chi^2$-*statistics distance*):

$$d_{\chi^2}(Q,T) = \sum_{i=0}^{N-1} \frac{(Q_i - m_i)^2}{m_i} \tag{2.5}$$

  pričom $m_i = \frac{Q_i + T_i}{2}$.

- *Manhattanská metrika* ($L_1$), ktorá je tiež známa pod názvom *vzdialenosť mestských blokov* - (*City block distance*):

$$d_1(Q,T) = \sum_{i=0}^{N-1} |Q_i - T_i| \tag{2.6}$$

- Okrem týchto dvoch metrík sa relatívne dobre osvedčila aj *Euklidovská vzdialenosť* ($L_2$) - (*Euclidean distance*):

$$d_2(Q,T) = \sqrt{\sum_{i=0}^{N-1} (Q_i - T_i)^2} \tag{2.7}$$

Ďalšie metriky, ktoré D. Zhang a G. Lu v článku testovali, sú:

- *Kosínová vzdialenosť* - (*Cosine distance*)

$$d_{cos}(Q,T) = 1 - cos\theta = 1 - \frac{Q^t T}{|Q|.|T|} \tag{2.8}$$

- *Kvadratická vzdialenosť* - (*Quadratic distance*):

$$d_{qad}(Q,T) = [(Q-T)^t A(Q-T)]^{\frac{1}{2}} \tag{2.9}$$

  pričom $A = [a_{ij}]$ je $N$x$N$ matica a $a_{ij}$ je koeficient podobnosti medzi indexami $i$ a $j$, ďalej platí, že $a_{ij} = 1 - d_{ij}/d_{max}$ a tiež $d_{ij} = |Q_i - T_j|$. Rovnicu 2.9 teda môžme prepísať na tento tvar:

$$d_{qad}(Q,T) = (\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} a_{ij}Q_iQ_j + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} a_{ij}T_iT_j + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} Q_iT_j)^{\frac{1}{2}} \tag{2.10}$$

- *Prienik histogramov* - (*Histogram intersection*):

$$d_{hi}(Q,T) = 1 - \frac{\sum_{i=0}^{N-1} min(Q_i,T_i)}{min(|Q|,|T|)} \tag{2.11}$$

- *Mahalanobisova vzdialenosť* - (*Mahalanobis distance*)

V tejto práci sa budeme zaujímať o *Chí-kvadrát test dobrej zhody* vzhľadom na dosiahnuté výsledky v [25]. A túto metriku aj aplikujeme v aplikácii.

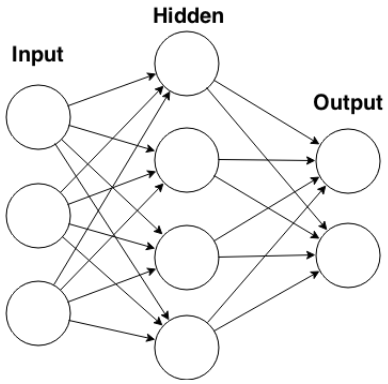
## 2.6 Hlboké konvolučné neurónové siete

*Hlboké konvolučné neurónové siete* (*Deep convolutional neural networks - DCNN*) sú odvodené od *dopredných neurónových sietí* (*feed-forward neural networks*) s úplným prepojením (obr. 2.6).

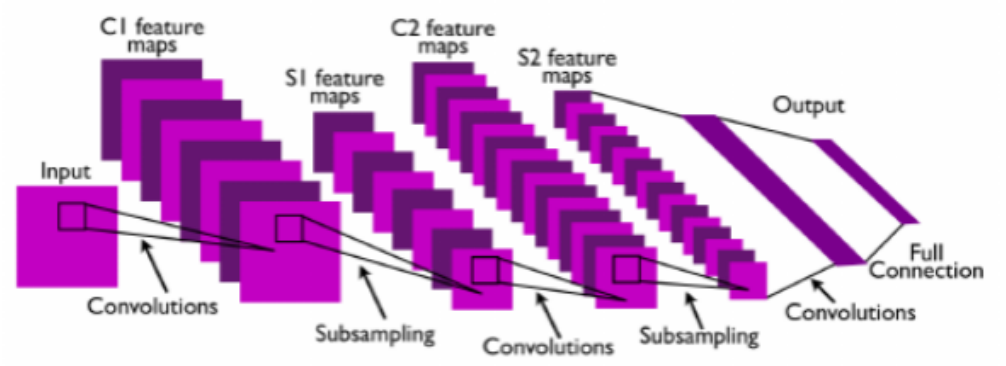
Obsahujú páry konvolučnej a podvzorkovacej vrstvy (v anglickej literatúre ich nájdeme pod názvami convolutional layer a pooling/subsampling layer). Fakt, že je neurónová sieť hlboká, hovorí o tom, že táto sieť má viac ako jednu vrstvu skrytých neurónov (anglicky *layer of hidden units*) medzi svojimi vstupmi a výstupmi[11], čím sa líši od klasických neurónových sietí.

Konvolučné neurónové siete boli spočiatku navrhnuté pre určitú špecifickú oblasť - rozpoznávanie rukou písaného písma[15](kap. 2.6.4). Tieto siete však disponujú štruktúrou a vlastnosťami, ktoré ich predurčujú k širšiemu nasadeniu, keď sa jedná o úlohy klasifikačného charakteru dvojrozmerných obrazových dát.

Konvolučné siete sú učenlivé viacstupňové architektúry vytvorené viacnásobnými stupňami. Vstup a výstup každého stupňa sú množiny polí nazývané *príznakové mapy* (*feature maps*)[16].




Obr. 2.6: Typická architektúra doprednej neurónovej siete, z ktorej je odvodená aj konvolučná neurónová sieť. Obsahuje 3 základné vrstvy neurónov - vstupnú, skrytú a výstupnú.




Obr. 2.7: Typická architektúra konvolučných sietí s dvomi príznakovými stupňami[16].

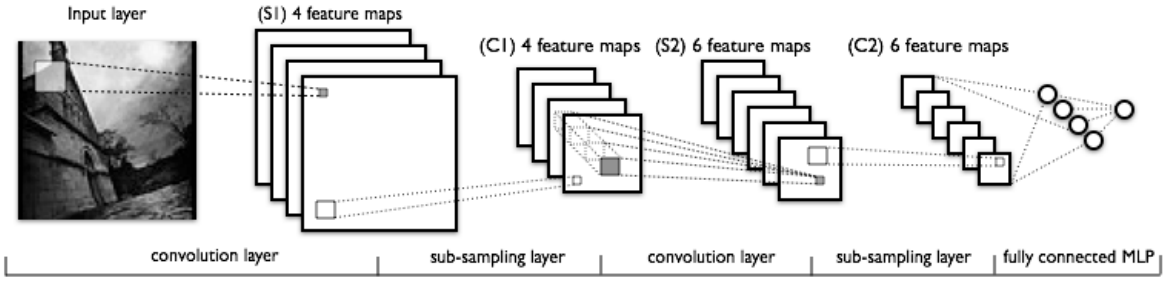
### 2.6.1 Architektúra

Každý stupeň hlbokých konvolučných neurónových sietí pozostáva z 3 vrstiev:

- vrstva filtrov - *filter bank layer*
- nelineárna vrstva - *non-linearity layer*
- vrstva delenia príznakov - *feature pooling layer*

Prvá zo spomenutých vrstiev - *vrstva filtrov* obsahuje sadu máp, pričom každá z týchto máp je naplnená výstupmi jedného príznakového detektoru. Tieto mapy úplne pokrývajú vstupný priestor a ich výstup reprezentuje extrahované príznaky na všetky miesta vstupu. Príznakový detektor je neurón so zdieľanými váhami s jeho súrodencami v jednej mape. Dobrá vlastnosť zdieľania váh je tá, že výrazne redukuje počet voľných parametrov na prácu v sieti s nimi. Obvykle je veľmi drahé používať filtre pod odlišnými veľkosťami a orientáciami v mape. Oveľa jednoduchšie je transformovať vstupné dáta.

*Nelineárna vrstva* kóduje príznaky aplikovaním nelineárnej funkcie. Bežná funkcia je logistický sigmoid, ale experimenty z [9] demonštrujú, že rektifikačný sigmoid môže byť




Obr. 2.8: Grafické zobrazenie LeNet modelu konvolučnej siete[17].

lepší. V súvislosti s rektifikačným sigmoidom alebo nejakou inou neohraničenou funkciou sa používa normalizácia odozvy (response normalization). Táto normalizácia spôsobuje konkurenciu medzi príznakmi na rovnakom mieste [12].

Deliaca pooling vrstva pracuje s každou mapou oddelene. Vezme oblasť aktivity príznakových detektorov a zakóduje ich do komplexnejších príznakov. Počas tohto procesu taktiež redukuje rozlíšenie vstupu. Deliaci proces (podvzorkovanie) poskytuje jemnú priestorovú invariantnosť. Nevýhodou tejto techniky je, že sa stráca informácia o presnej pozícii detekovaných príznakov. Možno tomu predísť delením prekrývajúcich sa oblastí.

Na vrchole tejto stupňovanej architektúry je bežné mať niekoľko úplne prepojených vrstiev (fully connected layers) ako možno vidieť na obrázku 2.8.

### 2.6.2 Konvolúcia

*Konvolúcia*[22] je matematický výraz, definovaný ako použitie funkcie opakovane cez výstup inej funkcie. V tomto kontexte to znamená použiť filter nad obrázkom vo všetkých možných offsetoch. Filter pozostáva z vrstvy váh so vstupom o veľkosti 2D políčka obrázku a výstupom je jedna jednotka (celok). Keď sa filter aplikuje opakovane, výsledné prepojenie vyzerá ako rad prekrývajúcich sa polí, ktoré sa mapujú do matice filtračných výstupov (alebo do niekoľkých takýchto matíc, v prípade, že bolo použitých niekoľko filtrov).

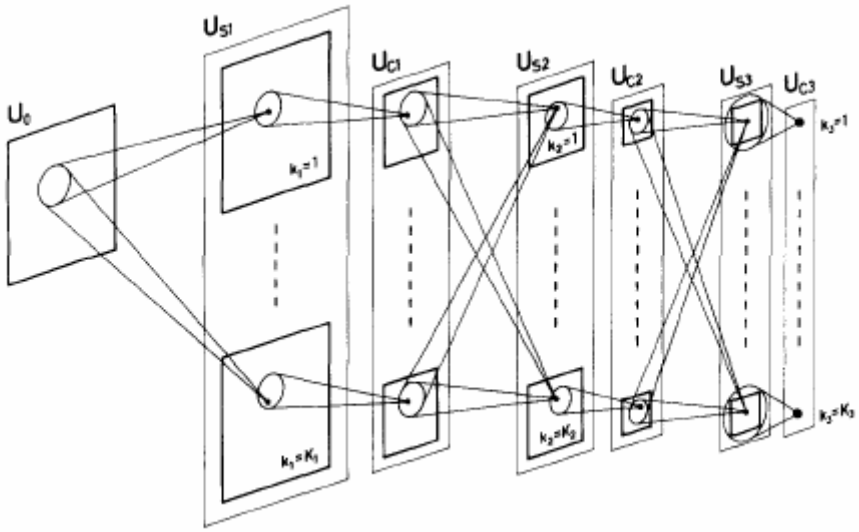
### 2.6.3 Podvzorkovanie

Podvzorkovanie, v anglickej terminológii známe ako *subsampling* alebo *down-sampling*[22], predstavuje zmenšovanie celkovej veľkosti signálu. V mnohých prípadoch, ako je napríklad kompresia zvuku pre hudobné súbory, sa podvzorkovanie vykonáva jednoducho pre zmenšenie veľkosti. Avšak čo sa týka výstupov 2D filtrov, podvzorkovanie môže byť myslené ako zvýšenie invariantnosti pozície filtrov. Špecifická metóda podvzorkovania použitá v konvolučných neurónových sieťach je známa ako *max pooling*. To zahŕňa rozdelenie matice výstupov filtru na malé neprekrývajúce sa mriežky (čím väčšia mriežka to je, tým väčšie zmenšenie signálu predstavuje), pričom z každej mriežky sa vezme maximálna hodnota do zmenšenej matice.

### 2.6.4 Neocognitron

*Neocognitron* je model neurónovej siete pre mechanizmus rozpoznávania rukou písaných znakov a vizuálnych vzorov[8]. Sieť je samo-organizovaná prostredníctvom učenia bez učiteľa

a nadobúda schopnosť rozpoznávať stimulačné vzory založené na geometrickej podobnosti ich tvarov bez ovplyvnenia ich pozíciami. Po dokončení samo-organizácie má sieť štruktúru podobnú hierarchickému modelu vizuálneho nervového systému, ktorý navrhli Hubel a Wiesel. Sieť pozostáva zo vstupnej vrstvy (pole fotoreceptorov), nasledovanej kaskádovitým spojením množstva modulárnych štruktúr, pričom každá z nich je zložená z dvoch vrstiev buniek spojených do kaskády (obr. 2.9).




Obr. 2.9: Schematický diagram zachytávajúci prepojenia medzi vrstvami v neocognitrone[8].

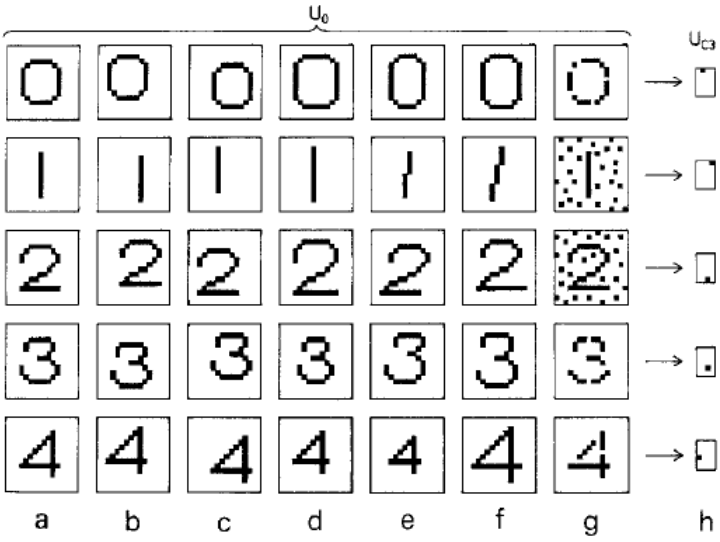
Keďže má sieť schopnosť učenia bez učiteľa (*unsupervised learning*), nie je potrebný žiadny „učiteľ" počas procesu samo-organizácie. Avšak potrebné je iba opakovane poskytovať množinu stimulačných vzorov do vstupnej vrstvy siete. Po opakovanom poskytovaní tejto množiny stimulačných vzorov, každý stimulačný vzor vyvoláva výstup z jednej C-bunky (komplexné bunky) poslednej vrstvy a obrátene, táto C-bunka reaguje iba na ten stimulačný vzor. Z toho vyplýva, že žiadna z C-buniek poslednej vrstvy neodpovedá viac ako jednému stimulačnému vzoru. Odozva C-buniek poslednej vrstvy nie je vôbec ovplyvnená pozíciou vzoru. Nie je ovplyvnená ani malou zmenou v tvare ani vo veľkosti stimulačného vzoru. Na obr. 2.10 môžme vidieť, aké zdeformované stimulačné vzory dokázal neocognitron správne rozpoznať.

## 2.7 Databáza fotografií

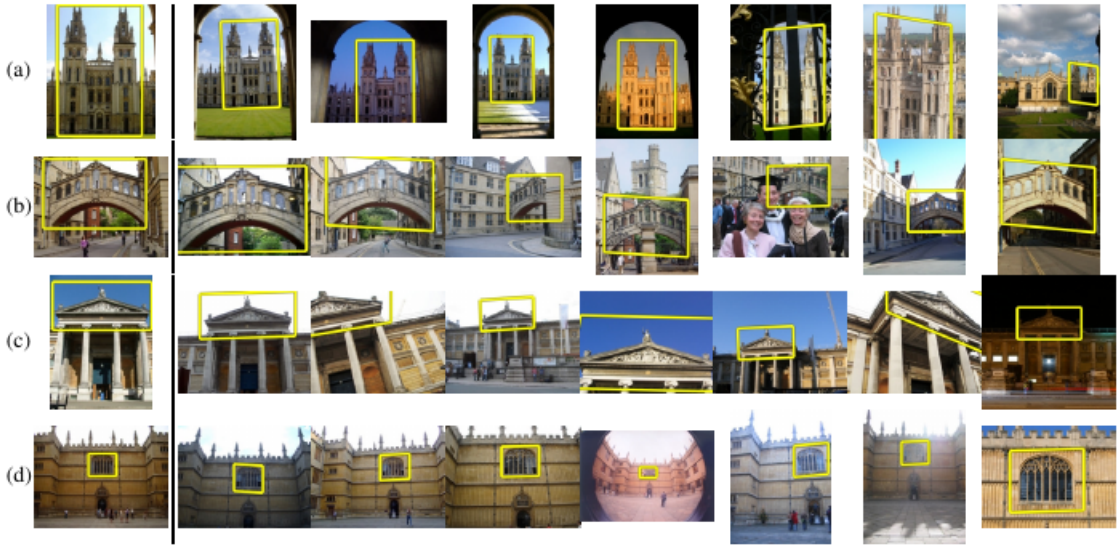
Na počiatku riešenia tejto práce bolo nevyhnutnosťou zvoliť si určitú databázu fotografií, s ktorou budeme ostatný čas pracovať. Spomedzi viacerých datasetov fotografií, ktoré sú k dispozícii na internete, som sa rozhodol pre oxfordskú databázu fotografií - *The Oxford Buildings Dataset*[1].

Tento dataset je na internete známy, o čom svedčia viaceré odborné články vzťahujúce sa práve na túto databázu fotografií. Hlavným dôvodom výberu tejto databázy bol však fakt, že databáza je anotovaná. To znamená, že k daným fotografiam existuje popis, na základe ktorého môžme určiť, ktoré fotografie sú si navzájom podobné a ktoré zase nie.

[1]http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/




Obr. 2.10: Príklady zdeformovaných stimulačných vzorov, ktoré neocognitron správne rozpoznal, a odozva konečnej vrstvy siete[8].




Obr. 2.11: Ukážka fotografií z oxfordského datasetu. Na obrázku vľavo môžeme vidieť 4 dopytové fotografie z tohto datasetu. Vpravo pri každej dopytovej fotografii je zobrazený zoznam najlepšie vyhľadaných fotografii podľa článku [20]. Každý riadok predstavuje inú pamiatku Oxfordu: (a) All Soul's College. (b) Bridge of sighs, Hertford College. (c) Ashmolean museum. (d) Bodleian window.

*The Oxford Buildings Dataset* pozostáva z 5062 fotografií rôznych rozmerov zozbieraných z webovej lokality pre zdieľanie fotografií a videa *Flickr*[2], hľadaním konkrétnych

[2] www.flickr.com

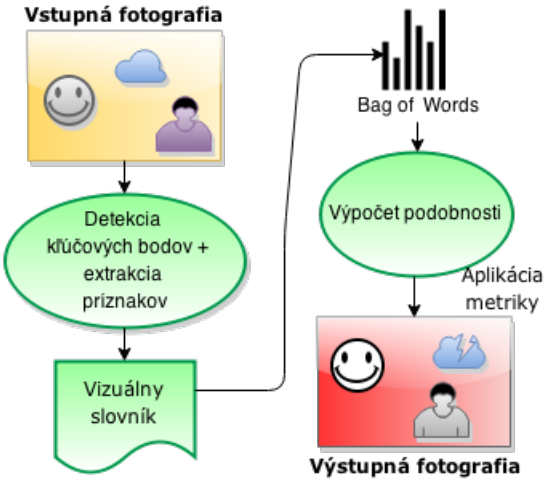
oxfordských pamiatok. Zbierka fotografií je manuálne anotovaná na generovanie podrobných „ground truth" pre 11 odlišných pamiatok, každá je reprezentovaná 5 možnými dopytovými fotografiami (*queries*). Toto dáva množinu 55 dopytových fotografií, nad ktorými môže byť vyhodnotený vyhľadávací systém.

# Kapitola 3

# State of the art

Táto kapitola vysvetľuje základný postup štandardne používaný pri vyhľadávaní podobných fotografií, tiež metódy, ktoré sa v dnešnej dobe využívajú na zlepšenie výsledkov. Okrem toho v tejto kapitole bude popísané alternatívne riešenie danej problematiky prostredníctvom hlbokých konvolučných neurónových sietí.

## 3.1 Konvenčný postup a metódy vyhľadávania v databázach fotografií




Obr. 3.1: Schematický náčrt systému na vyhľadávanie obrázkov (*Image Retrieval System*)

Systém na vyhľadávanie obrázkov (*Image retrieval system*) je možné v dnešnej dobe implementovať rôznymi spôsobmi a technikami. Schematický náčrt 3.1 zachytáva základný konvenčný postup pri vyhľadávaní fotografií. Vstupom systému je vzorová fotografia, ku ktorej chce používateľ systému nájsť najpodobnejšiu fotografiu, prípadne zoznam najpodobnejších fotografií z určitej databázy. Na druhej strane, výstupom tohto systému bude práve spomínaná najpodobnejšia fotografia, resp. zoznam najpodobnejších fotografií. Vyhľadávanie podobných fotografií spočíva v nasledujúcich krokoch:

1. Po tom, čo sa načíta fotografia, nasleduje *detekcia* a *extrakcia lokálnych príznakov* (deskriptorov kľúčových bodov) fotografie.

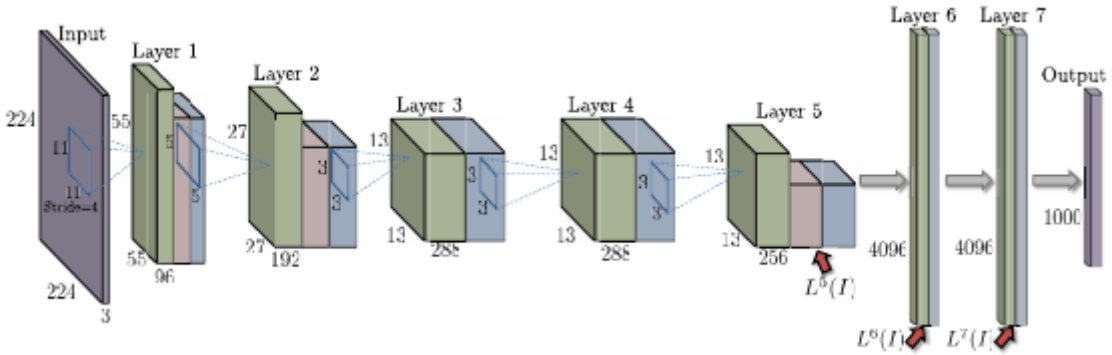
2. Každý príznakový vektor (deskriptor) sa následne priradí k najbližšiemu vizuálnemu slovu definovanému vo *vizuálnom slovníku.*

3. Nasleduje vytvorenie histogramu početnosti výskytu vizuálnych slov vo fotografii - *Bag of Words (BoW)*, čiže fotografiu bude reprezentovať histogram (číselný vektor).

4. Aplikácia vhodnej *metriky*, pomocou ktorej bude možné vypočítať najkratšiu vzdialenosť medzi BoW vektorom vzorovej fotografie a BoW vektormi trénovacích fotografií z databáze. Takto sa dopracujeme k najpodobnejšej fotografii, resp. k zoznamu najpodobnejších fotografií k danej vzorovej fotografii.

## 3.2 Deep Convolutional Neural Networks

Hlboké konvolučné neurónové siete (DCNN) [2] v poslednej dobe výrazne pokročili v klasifikácii obrazu a patrične preto vzbudili veľký záujem v komunite počítačového videnia. S problematikou klasifikácie obrazu súvisí problematika vyhľadávania v obrazových databázach, t.j. úloha hľadania obrázkov, ktoré obsahujú rovnaký objekt alebo scénu ako dopytový obrázok. Príznaky objavujúce sa vo vrchných vrstvách konvolučných neurónových sietí naučených klasifikovať obrázky, môžu slúžiť ako dobré deskriptory pre vyhľadávanie obrázkov. Takéto príznaky budeme označovať ako neurónové kódy (*neural codes*).




Obr. 3.2: Architektúra konvolučnej neurónovej siete použitej na experimenty, ktorým sa táto kapitola bližšie venuje. Experimenty vykonal A. Babenko spolu s ďalšími[2]. Tmavo modré uzly predstavujú vstup (RGB obrázok o veľkosti 224 x 224) a výstup (označenia 1000 tried). Zelené jednotky reprezentujú výstupy konvolúcií, červené zodpovedajú výstupom z max poolingu a modré predstavujú výstupy ReLU (*Rectified Linear Unit*). Vrstvy 6, 7 a 8 (výstup) sú plne prepojené k predošlým vrstvám. Jednotky, ktoré zodpovedajú neurónovým kódom sú znázornené červenou šípkou.

Na základe experimentov, ktoré vykonal A. Babenko a ďalší [2] s neurónovými kódmi, môžme porovnať výsledky vyhľadávania fotografií získané rôznymi prístupmi - tak prostredníctvom konvolučných neurónových sietí ako aj metódami, ktoré neurónové siete nevyužívajú. Z výsledkov je patrné, že konvolučné neurónové siete použité pri vyhľadávaní fotografií prinášajú zaujímavé výsledky. Architektúra konvolučnej neurónovej siete, ktorá bola použitá

na tieto experimenty, je zobrazená na obr. 3.2. V tejto kapitole teda nájdeme porovnanie viacerých prístupov vychádzajúcich z DCNN pre vyhľadávanie fotografií, taktiež ale aj prístupy, ktoré sa netýkajú výhradne DCNN, avšak patria medzi state-of-the-art.

### 3.2.1 Predtrénované neurónové kódy - Pretrained neural codes

V tejto kapitole bude vyhodnotený výkon neurónových kódov získaných prechodom obrázkov cez hlbokú konvolučnú sieť, ktorá je natrénovaná na klasifikovanie 1000 Image-Net tried[14]. To znamená, že táto konvolučná sieť pracuje ako extraktor deskriptorov - predtrénovaných neurónových kódov. Sieť je použiteľná na obrázky o veľkosti 224 x 224. Obrázky iných veľkostí sú upravené na túto veľkosť (bez orezania). Architektúra konvolučnej siete je dopredná, pre daný obrázok $I$ sieť produkuje sekvenciu vrstvových aktivácií. Prostredníctvom $L^5(I), L^6(I)$, a $L^7(I)$ označujeme aktivácie (výstup) príslušných vrstiev pred ReLU transformáciou. Každý z týchto vysoko dimenzionálnych vektorov reprezentuje hlboký deskriptor (*neurónový kód*) vstupného obrázka.

Výkon neurónových kódov bude vyhodnotený na 4 štandardných datasetoch:

- *Oxford Buildings Dataset (Oxford)* - ako bolo už spomenuté, tento dataset obsahuje 5062 fotografií, z toho 55 dopytových fotografií. Výkon bude vyhodnotený prostredníctvom strednej priemernej presnosti (*mean average precision - mAP*) nad danými dopytovými fotografiami.

- *Oxford Buildings Dataset+100K (Oxford 105K)* - rovnaký dataset, ktorý má o 100 000 fotografií viac.

- *INRIA Holidays Dataset (Holidays)* - obsahuje 1491 prázdninových fotografií prislúchajúcich do 500 skupín založených na rovnakej scéne alebo objekte. Jedna fotografia z každej skupiny slúži ako dopytová. Výkon sa udáva taktiež ako stredná priemerná presnosť (mAP) nad 500 dopytovými fotografiami. Dataset je však modifikovaný. Hlboké architektúry sú trénované na fotografiách s prirodzenou orientáciou. Keďže niektoré fotografie tohto datasetu nemajú prirodzenú orientáciu (sú otočené o $\pm 90$ stupňov), bolo ich treba pred zahájením experimentu upraviť, aby boli orientované prirodzene. Všetky výsledky budú uvádzané vzhľadom na túto modifikáciu (na obr. 3.3 označené hviezdičkou).

- *University of Kentucky Benchmark Dataset (UKB)* - obsahuje 10200 vnútorných fotografií 2550 objektov (4 fotografie na objekt). Každá fotografia je použitá na dopyt v rámci zvyšku datasetu. Výkon sa v tomto prípade zaznamenáva ako priemerný počet fotografií s rovnakým objektom v rámci top 4 výsledkov, je to číslo medzi 0 a 4.

Výsledky pre neurónové kódy produkované sieťou trénovanou na ILSVRC[1] triedach sa nachádzajú v strednej časti obr. 3.3. Výsledky boli získané použitím *L2* vzdialenosti na L2-normalizovaných neurónových kódoch. Výsledky sú uvedené pre každú z vrstiev 5, 6, 7. Spomedzi týchto vrstiev, 6. vrstva mala najlepšie výsledky.

### 3.2.2 Pretrénované neurónové kódy - Retrained neural codes

Priamočiara myšlienka na zlepšenie výkonu neurónových kódov je pretrénovať konvolučnú architektúru na dataset s obrazovými štatistikami a triedami, ktoré sú relevantnejšie pre datasety použité v testovaní.

[1] ImageNet Large Scale Visual Recognition Challenge - http://www.image-net.org/challenges/LSVRC/

Najprv je teda potrebné zozbierať dataset, ktorý bude relevantný pre datasety pamiatkového typu (Oxford Buildings a Holidays). Potom môžme tento zozbieraný dataset použiť na trénovanie konvolučnej neurónovej siete s rovnakou architektúrou ako pre ILSVRC (líši sa len v počte výstupov, namiesto pôvodných 1000 je to teraz len 672 na základe použitého datasetu). Inak je tréning rovnaký ako pri pôvodnej sieti.

Výsledky pre neurónové kódy produkované sieťou pretrénovanou na pamiatkové triedy sú uvedené na obr. 3.3. Pre datasety, ktoré sú založené na fotografiách pamiatok - Oxford a Oxford 105K, nastalo veľké zlepšenie. Zlepšenie pre Holidays dataset je menšie, ale stále značné. Výkon upravených $L^6(I)$ príznakov na datasete Holidays je lepší ako pre prv publikované systémy založené na holistických príznakoch.

| Descriptor | Dims | Oxford | Oxford 105K | Holidays | UKB |
|---|---|---|---|---|---|
| Fisher+color | 4096 | — | — | **0.774** | 3.19 |
| VLAD+adapt+innorm | 32768 | 0.555 | — | 0.646 | — |
| Sparse-coded features | 11024 | — | — | 0.767 | **3.76** |
| Triangulation embedding | 8064 | **0.676** | **0.611** | 0.771 | 3.53 |
| **Neural codes trained on ILSVRC** | | | | | |
| Layer 5 | 9216 | 0.389 | — | 0.690* | 3.09 |
| Layer 6 | 4096 | 0.435 | 0.392 | 0.749* | 3.43 |
| Layer 7 | 4096 | 0.430 | — | 0.736* | 3.39 |
| **After retraining on the Landmarks dataset** | | | | | |
| Layer 5 | 9216 | 0.387 | — | 0.674* | 2.99 |
| Layer 6 | 4096 | 0.545 | 0.512 | **0.793*** | 3.29 |
| Layer 7 | 4096 | 0.538 | — | 0.764* | 3.19 |
| **After retraining on turntable views (Multi-view RGB-D)** | | | | | |
| Layer 5 | 9216 | 0.348 | — | 0.682* | 3.13 |
| Layer 6 | 4096 | 0.393 | 0.351 | 0.754* | 3.56 |
| Layer 7 | 4096 | 0.362 | — | 0.730* | 3.53 |

Obr. 3.3: Dosiahnuté výsledky experimentov pre neurónové kódy a porovnanie s ďalšími state-of-the-art metódami. Pre datasety Oxford, Oxford 105K a Holidays je výkon vyhodnotený prostredníctvom strednej priemernej presnosti (*mean average precision - mAP*). Pre dataset UKB sa výkon udáva ako priemerný počet fotografii s rovnakým objektom v rámci top 4 výsledkov, je to číslo medzi 0 a 4.[2].

Po pretrénovaní na dataset pamiatok výkon na UKB datasete poklesol. To reflektuje fakt, že triedy v UKB datasete, ktoré korešpondujú s viacnásobnými vnútornými pohľadmi odlišných malých objektov, sú podobnejšie niektorým triedam v pôvodnom datasete ILSVRC ako k pamiatkovým fotografiám. Potvrdzuje to aj 2. experiment pretrénovania, kde bol použitý Multi-view RGB-D dataset[2], ktorý obsahuje rôzne pohľady 300 objektov domácnosti. Každý objekt bol spracovaný ako samostatná trieda, do jednej triedy spadá 200 obrázkov. Sieť po pretrénovaní na tomto 60 000 obrázkovom datasete priniesla výsledky, ktoré sú uvedené taktiež na obr. 3.3. Tento experiment zvýšil výkon na súvisiacom datasete, presnosť na UKB sa zvýšila z 3.43 na 3.56. Avšak, ako sme mohli očakávať, tento experiment vzhľadom na špecifickosť použitého datasetu na pretrénovanie, znížil výkon nesúvisiacich datasetov (Oxford, Oxford 105K).

[2]http://rgbd-dataset.cs.washington.edu/

### 3.2.3 Skomprimované neurónové kódy - Compressed neural codes

Zatiaľ čo sú neurónové kódy v týchto experimentoch vysoko-dimenzionálne (napr. 4096 pre $L^6(I)$), aj keď menej vysoko-dimenzionálne ako ostatné state-of-the-art holistické deskriptory, vynára sa otázka ich vhodnej kompresie. Bude preskúmané, ako sa zníži účinnosť neurónových kódov s bežnou kompresiou založenou na $PCA$. Vyhodnotenie bude zamerané na $L^6(I)$, pretože výkon neurónových kódov asociovaný so 6.vrstvou bol lepší ako s kódmi z ostatných vrstiev.

**PCA kompresia:** PCA tréning bol vykonaný na 100 000 náhodných obrázkoch z datasetu pamiatok. PCA kompresia bola vykonaná pre rôzne rozmery. Kvalita neurónových kódov $L^6(I)$ pre rôzne PCA kompresie je uvedená na obr. 3.4. PCA pracuje prekvapujúco dobre. Neurónové kódy môžu byť teda komprimované do 256 alebo dokonca aj 128 rozmerov takmer bez žiadnej straty kvality. Na obr. 3.5 sa porovnávajú rozličné holistické deskriptory skomprimované do 128 rozmerov. Pre datasety Oxford a Holidays, neurónové kódy pretrénované na pamiatkach poskytujú nový state-of-the-art medzi nízko-dimenzionálnymi globálnymi deskriptormi.

| Dimensions | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|
| **Oxford** | | | | | | |
| Layer 6 | 0.328 | 0.390 | 0.421 | 0.433 | 0.435 | 0.435 |
| Layer 6 + landmark retraining | 0.418 | 0.515 | 0.548 | 0.557 | 0.557 | 0.557 |
| Layer 6 + turntable retraining | 0.289 | 0.349 | 0.377 | 0.391 | 0.392 | 0.393 |
| **Oxford 105K** | | | | | | |
| Layer 6 | 0.260 | 0.330 | 0.370 | 0.388 | 0.392 | 0.392 |
| Layer 6 + landmark retraining | 0.354 | 0.467 | 0.508 | 0.523 | 0.524 | 0.522 |
| Layer 6 + turntable retraining | 0.223 | 0.293 | 0.331 | 0.348 | 0.350 | 0.351 |
| **Holidays** | | | | | | |
| Layer 6 | 0.591 | 0.683 | 0.729 | 0.747 | 0.749 | 0.749 |
| Layer 6 + landmark retraining | 0.609 | 0.729 | 0.777 | 0.789 | 0.789 | 0.789 |
| Layer 6 + turntable retraining | 0.587 | 0.702 | 0.741 | 0.756 | 0.756 | 0.756 |
| **UKB** | | | | | | |
| Layer 6 | 2.630 | 3.130 | 3.381 | 3.416 | 3.423 | 3.426 |
| Layer 6 + landmark retraining | 2.410 | 2.980 | 3.256 | 3.297 | 3.298 | 3.300 |
| Layer 6 + turntable retraining | 2.829 | 3.302 | 3.526 | 3.552 | 3.556 | 3.557 |

Obr. 3.4: Výkon neurónových kódov (pôvodných aj pretrénovaných) pre rozličné hodnoty PCA-kompresie. Výkon deskriptorov je takmer neovplyvnený do dimenzionality 256 [2].

| Descriptor | Oxford | Oxford 105K | Holidays | UKB |
|---|---|---|---|---|
| Fisher+color | — | — | 0.723 | 3.08 |
| VLAD+adapt+innorm | 0.448 | 0.374 | 0.625 | — |
| Sparse-coded features | — | — | 0.727 | **3.67** |
| Triangulation embedding | 0.433 | 0.353 | 0.617 | 3.40 |
| **Neural codes trained on ILSVRC** | | | | |
| Layer 6 | 0.433 | 0.386 | 0.747* | 3.42 |
| **After retraining on the Landmarks dataset** | | | | |
| Layer 6 | **0.557** | **0.523** | **0.789*** | 3.29 |
| **After retraining on turntable views (Multi-view RGB-D)** | | | | |
| Layer 6 | 0.391 | 0.348 | 0.756* | 3.55 |

Obr. 3.5: Porovnanie PCA-skomprimovaných neurónových kódov (128 rozmerov) so state-of-the-art holistickými obrazovými deskriptormi rovnakej dimenzionality. Neurónové kódy pretrénované na datasete pamiatok a skomprimované prostredníctvom PCA vytvorili nový state-of-the-art na datasetoch Holidays, Oxford a Oxford 105K [2].

# Kapitola 4

# Návrh aplikácie

V tejto kapitole sa zameriame podrobnejšie na návrh samotnej aplikácie - systému pre vyhľadávanie fotografií (*image retrieval system*). Spomenieme taktiež algoritmy a metódy použité pri riešení tejto aplikácie.

## 4.1 Ciele

Vyhľadávanie fotografií v databáze podľa príkladu je téma, na ktorú sa dá pozerať z viacerých uhlov pohľadu. Existuje niekoľko cieľov pri implementácií. Jedným je implementovať aplikáciu, ktorá bude primárne zameraná na rýchlosť vyhľadávania fotografií. Obdobným cieľom je implementácia aplikácie, ktorej pôjde primárne o presnosť spracovania požiadavku na vyhľadávanie.

V tejto práci nám ide hlavne o porovnanie rôznych prístupov. To je hlavným cieľom. Vedľajšími cieľmi môže byť rýchlosť aj presnosť aplikácie. Budú použité rôzne metódy a prístupy, ktorými sa dá dosiahnuť vyhľadávanie fotografií a v závere budú tieto prístupy navzájom porovnané medzi sebou a taktiež s výsledkami uvedenými v kapitole 3. V rámci vyhodnotenia budú taktiež popísané problémy a komplikácie, ktoré nejakým spôsobom ovplyvnili samotnú implementáciu.

## 4.2 Návrh riešenia

Riešenie sa skladá z dvoch hlavných fáz:

1. *Vytvorenie potrebných vstupov* - pre úspešný beh aplikácie na vyhľadávanie fotografií je nevyhnutné najskôr pripraviť vstupné dáta, aby bolo možné hľadať podobné fotografie k zadaným vzorovým fotografiám. Prvú fázu znázorňuje obr. 4.1.

   Charakteristiky 1. fáze:

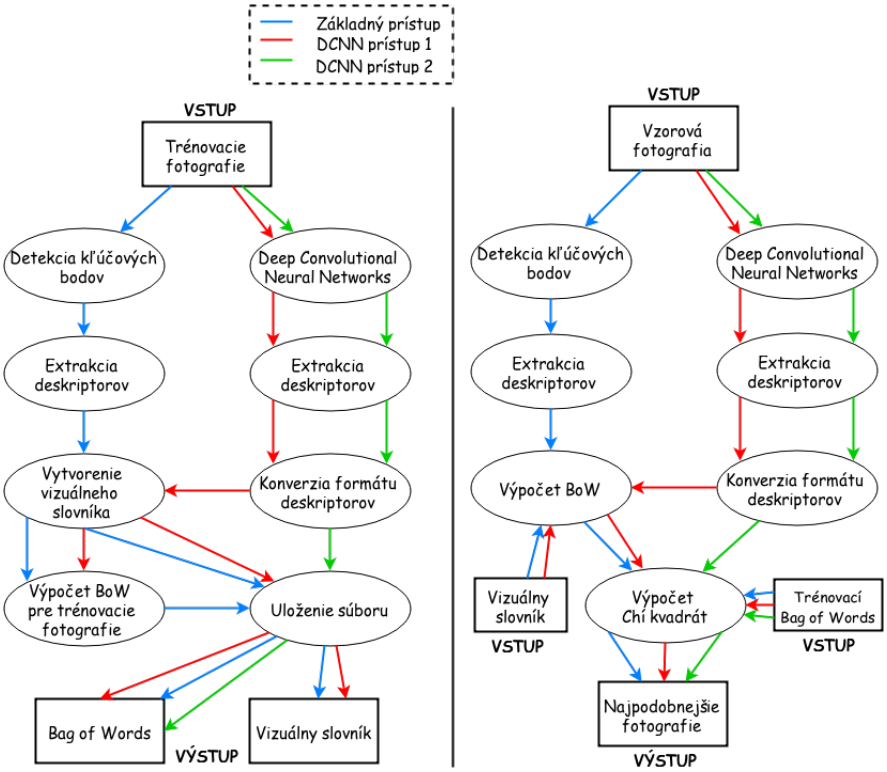
   - *vstup:* sada trénovacích fotografií
   - *výstup:* vizuálny slovník, BoW pre trénovacie fotografie

2. *Výpočet dát pre vzorovú dopytovú fotografiu a vyhľadávanie na základe podobnosti* - po tom, čo sú potrebné vstupy vytvorené, môžme prejsť k výpočtu dát reprezentujúcich dopytovú (query) fotografiu. Keď disponujeme týmito dátami, môžme zahájiť proces vyhľadávania podobných fotografií k zadanej dopytovej fotografii prostredníctvom

zvolenej metriky porovnávania histogramov BoW. Činnosť tejto fázy zachytáva obr. 4.1.

Charakteristiky 2.fáze:

- *vstup:* vizuálny slovník, BoW vektory trénovacích fotografií, vzorová dopytová fotografia
- *výstup:* najpodobnejšie fotografie




Obr. 4.1: Vľavo = schéma vytvorenia potrebných vstupov (fáza 1). Vpravo = schéma výpočtu dát pre vzorovú fotografiu a vyhľadávania na základe podobnosti (fáza 2). DCNN prístup 1 predstavuje prístup, pri ktorom sú extrahované deskriptory z DCNN použité namiesto SIFT deskriptorov pri základnom prístupe (nasleduje tvorba vizuálneho slovníka, BoW, atď.). DCNN prístup 2 predstavuje iný prístup, extrahované deskriptory z DCNN sú použité namiesto BoW deskriptorov, takže nasleduje výpočet podobnosti histogramov metrikou chí kvadrát.

### 4.2.1 Použité metódy a algoritmy

Lokálne príznaky pri základnom prístupe sú extrahované prostredníctvom algoritmu SIFT. Každý príznak predstavuje 128-dimenzionálny vektor. Príznakom z DCNN sa venuje nasledujúca kapitola 4.2.2. Zhlukovanie využíva algoritmus k-means, ktorý je podrobne popísaný v kapitole 2.3.1. Vizuálny slovník využíva štruktúru k-dimenzionálneho stromu a algoritmus hľadania najbližšieho suseda. Fotografie sú reprezentované pomocou histogramu Bag

of Words. Ako metrika porovnania histogramov je použitá metrika Chí-kvadrát test dobrej zhody (kapitola 2.5).

### 4.2.2 Hlboké konvolučné neurónové siete DCNN

Pre oba prístupy platí, že aby sme mohli extrahovať deskriptory z DCNN, je potrebné predložiť danej neurónovej sieti fotografie, z ktorých dané deskriptory budú extrahované. Vstupom pre neurónovú sieť by mali byť fotografie rovnakých rozmerov. Keďže v použitom oxfordskom datasete fotografie majú rôzne rozmery, bolo nutné tieto fotografie najprv upraviť. V tomto prípade sa jedná o rozmery 256x256 a 525x525. Na extrakciu príznakov bol použitý 16 vrstvový model konvolučnej neurónovej siete, o ktorej pojednáva článok [21].

Tieto výstupy predstavujú odozvy po 3., 4. a 5. pooling vrstve (môžme si ich pre jednoduchosť označiť ako L3, L4 a L5), z toho dôvodu majú tieto výstupy rôzne rozmery. Ich prehľad sa nachádza v tabuľke 4.1. Každý výstup je vo forme 3D matice, pričom jednotlivé rozmery predstavujú šírku a výšku regiónu fotografie, ktorý daný príznak popisuje a tretí rozmer predstavuje rozmer deskriptoru (v našom prípade 256 a 512 v porovnaní so 128 dimenzionálnym SIFT príznakom).

| Rozmer fotografie | Vrstva | Rozmery |
|---|---|---|
| 256x256 | L03 | 28x28x256 |
| 256x256 | L04 | 12x12x512 |
| 256x256 | L05 | 4x4x512 |
| 525x525 | L03 | 61x61x256 |
| 525x525 | L04 | 28x28x512 |
| 525x525 | L05 | 12x12x512 |

Tab. 4.1: Prehľad rôznych druhov príznakov extrahovaných z DCNN.

# Kapitola 5

# Implementácia

Na základe návrhu riešenia z predošlej kapitoly je realizovaná implementácia tejto práce. V tejto kapitole sa budeme venovať implementačným detailom aplikácie a rôznym aspektom, ktoré s tým súvisia. Budú tu taktiež rozobraté problémy, ktorým bolo treba počas samotnej implementácie čeliť.

Aplikácia ako celok pozostáva z niekoľkých menších aplikácií, ktoré riešia čiastkové úlohy vyhľadávania fotografií. Väčšina týchto aplikácií je napísaných v jazyku C++. Dôvodom výberu tohto jazyka bol v nemalej miere určite fakt, že knižnica *OpenCV*, ktorá je bohato zastúpená v zdrojových kódoch tejto aplikácie, poskytuje rozhranie pre C++ (okrem toho má rozhranie ešte pre C, Python, Java a MATLAB). Ďalším dôvodom bola tiež skutočnosť, že s jazykom C++ som zatiaľ do kontaktu veľmi neprišiel, tak som si chcel prehĺbiť znalosti aj v tomto smere.

Na úvod tejto kapitoly by som sa chcel ešte poďakovať pánovi Ing. Michalovi Hradišovi za to, že mi spočítal potrebné výstupy z DCNN.

## 5.1 Knižnica OpenCV

OpenCV (Open source Computer Vision) je voľne dostupná knižnica, ktorá ponúka množstvo algoritmov z oblasti počítačového videnia a strojového učenia. Tieto algoritmy môžu byť použité na detekciu a rozpoznávanie tvárí, identifikáciu objektov, klasifikáciu ľudskej činnosti vo videách, extrakciu 3D modelov objektov, hľadanie podobných obrázkov v obrazových databázach a tak ďalej.

Knižnica OpenCV podporuje viaceré operačné systémy, ako Windows, Linux, Android a Mac OS. Podrobné informácie o všetkých triedach a funkciách sa dajú nájsť v oficiálnej dokumentácii[1]. Tak ako aj rôzne návody, tutoriály a vzorové zdrojové kódy určené na demonštráciu algoritmov.

### 5.1.1 Detekcia kľúčových bodov a extrakcia lokálnych príznakov

OpenCV ponúka možnosti pre prácu s obrazovými dátami. Hned po spustení aplikácie teda dôjde k načítaniu vstupných fotografií prostredníctvom funkcie `imread()`, ktorá načíta fotografiu do matice bodov, predstavujúcej triedu `Mat`. Na matici reprezentujúcej načítanú fotografiu následne pomocou triedy `SiftFeatureDetector` detekuje kľúčové body a prostredníctvom triedy `SiftDescriptorExtractor` extrahuje príznaky. Ako už tieto 2 triedy napovedajú, v tejto práci sa budeme zaujímať o SIFT lokálne príznaky vzhľadom na viaceré výhody (napr. sú invariantné voči posunu a otočeniu obrázku a zmene mierky).

### 5.1.2 Zhlukovanie

Extrahované príznaky je treba združiť do jednej matice `Mat`, aby mohli byť nazhlukované. Zhlukovanie zabezpečuje trieda `BOWKmeansTrainer`, ktorá v rámci funkcie `cluster()` volá funkciu `kmeans()`. Tá sa už postará o samotné nazhlukovanie. Najskôr určí stredy výsledných zhlukov, ich počet je definovaný parametrom. Taktiež určí príslušnosť príznakov (vektorov) ku konkrétnym zhlukom. To, ako sa nainicializujú centrá zhlukov je možné určiť pomocou príznaku (flag-u). V našom prípade je použitá voľba `KMEANS_PP_CENTERS`, ktorá zabezpečí vhodné rozmiestnenie centier do priestoru vďaka algoritmu k-means++. Štruktúra `TermCriteria` definuje ukončujúcu podmienku - maximálny počet iterácií. Ten je v tejto práci určený na základe experimentov na 4.

## 5.2 Popis aplikácie

V tejto kapitole budú popísané všetky programy, ktoré sa podieľajú na funkčnosti celkovej aplikácie. Niektoré prgramy pracujú v 2 módoch - *basic* a *dcnn*. Mód basic je mód, ktorý je využívaný v základnom prístupe riešenia aplikácie. Mód dcnn sa využíva v prístupoch, ktoré využívajú DCNN, t.j. nepracuje sa so SIFT príznakovými vektormi v tom prípade, ale s extrahovanými príznakmi z DCNN.

Okrem nasledujúcich programov sa na finálnej aplikácií podieľa ešte niekoľko skriptov v *BASH*-i, ktoré automatizujú celý proces vyhľadávania.

### 5.2.1 trainer

Pracuje v 2 módoch. *Mód Basic:* na vstupe očakáva adresár s fotografiami, ktoré postupne načíta, detekuje kľúčové body, extrahuje deskriptory a z týchto deskriptorov vytvorí vizuálny slovník. *Mód Dcnn:* na vstupe očakáva adresár so súbormi vo formáte **.yaml*, ktoré predstavujú extrahované deskriptory (1 súbor = 1 fotografia). Tieto súbory taktiež postupne načíta a z deskriptorov vytvorí vizuálny slovník.

### 5.2.2 get_bow

Program vytvorí prostredníctvom načítaného vizuálneho slovníka BoW histogram pre všetky fotografie. Pracuje v 2 módoch. *Basic:* Načíta každú fotografiu (**.jpg*), detekuje kľúčové body a vypočíta patričný BoW vektor. *Dcnn:* Načíta každú fotografiu (**.yaml*) a vypočíta jej BoW vektor.

### 5.2.3 get_ranked_list

Program vypočítava ranked list pre vstupnú dopytovú fotografiu prostredníctvom zadaného vizuálneho slovníka. Najprv načíta všetky fotografie v databáze cez vstupný BoW súbor, ktorý obsahuje všetky BoW vektory všetkých fotografií. Potom načíta slovník a dopytovú fotografiu, ktorej detekuje kľúč. body, vypočíta deskriptory a z nich vypočíta BoW vektor pre túto dopytovú fotografiu. Následne porovnáva dopytový BoW vektor s ostatnými BoW vektormi a hľadá najbližší vektor k nemu pomocou metriky chí kvadrát. Potom vytvorí ranked list, ktorý uloží do súboru špecifikovanému vstupným parametrom.

### 5.2.4 get_bowfile_from_yamls

Program načíta všetky **.yaml* súbory, ktoré predstavujú extrahované deskriptory z DCNN a z týchto súborov vytvorí jeden veľký spoločný BoW súbor (*.yaml*). Tento program sa využíva pri prístupe, keď sa výstupy z DCNN používajú priamo na výpočet BoW.

### 5.2.5 get_rl_from_dcnn

Tento program nadväzuje na predošlý a vypočítava ranked list pre zadanú dopytovú fotografiu vo formáte (*.yaml*), pričom používa súbor zo vstupu so všetkými BoW vektormi.

Vzhľadom na to, že výstupné deskriptory z DCNN sú vo formáte **.npz* a C++ ich nemôže načítať, tak je potrebné ich konvertovať prostredníctvom Pythonu do formátu, ktorému C++ rozumie. Na konverziu slúžia nasledujúce 2 programy:

### 5.2.6 convert_npz_to_csv

Program načíta pomocou argumentu príkazového riadku *.npz* súbor (čiže výstup z DCNN), a keďže je to 3D tenzor, tak ho upraví na 2D a vytvorí z neho *.csv* súbor, ktorý sa už dá prepojiť s C++.

### 5.2.7 csv2yaml

Program skonvertuje vstupný *.csv* súbor na *.yaml* súbor, ktorý je vhodnejší na uloženie 2D vektoru a následnú prácu s ním.

# Kapitola 6

# Vyhodnotenie

Táto kapitola obsahuje popis všetkých použitých metód, postupov, algoritmov i prístupov a zahŕňa taktiež zistené výsledky. Spomenuté budú takisto aj prípadné komplikácie a prekážky pri získavaní výsledkov, ktoré zabránili ďalšiemu experimentovaniu so systémom, príp. celý experiment nasmerovali určitým vhodnejším smerom. Jednotlivé metódy vyhľadávania fotografií budú vyhodnocované prostredníctvom priemernej presnosti (*average precision*), na základe ktorej sa bude dať relatívne zhodnotiť, či je daný systém úspešný, alebo nie.

Celá aplikácia bola vyvíjaná lokálne, vypracovaná na notebooku HP ProBook 4535s, ktorý disponuje procesorom AMD A6-3420M so 4 jadrami s frekvenciou 1.5GHz. Na tomto počítači boli taktiež vykonávané všetky experimenty, merania, testy a výpočty.

Prvá časť experimentov sa venuje základnému riešeniu problematiky vyhľadávania fotografií v databáze podľa príkladu. Druhá časť sa zaoberá experimentami, v ktorých vystupujú hlboké konvolučné neurónové siete. Tie sa v oblasti počítačového videnia a teda aj v úlohách klasifikácie a vyhľadávania obrázkov stali veľmi populárne. Záverečná časť tejto kapitoly sa venuje otázkam budúceho vývoja aplikácie.

## 6.1 Základné riešenie

Základné riešenie bolo implementované podľa návrhu na obr. 4.1. Veľkosť vizuálneho slovníka je 1000 slov a zhlukovanie sa bude opakovať 4 krát. Tabuľka 6.1 ukazuje porovnanie pri vytváraní vizuálneho slovníka deskriptormi extrahovanými z fotografie pri základnom prístupe a deskriptormi extrahovanými z DCNN pri DCNN prístupe 1.

| Dataset | Prístup | Počet fotografií | Počet slov | Doba vytvárania [min.] | Veľkosť [MB] |
|---|---|---|---|---|---|
| Oxford 5K | basic | 1000 | 1000 | 76 | 2.2 |
| Oxford 5K | dcnn | 5062 | 1000 | 40 | 8.4 |

Tab. 6.1: Tvorba vizuálneho slovníka pri základnom prístupe a DCNN prístupe 1. Slovník pre dcnn má viac dát, no doba vytvárania je kratšia ako pri 'basic' slovníku. To je celkom podozrivé. Pravdepodobne nastal problém pri zhlukovaní. Viac v kapitole 6.2.1.

Výsledky, ktoré môj naimplementovaný systém dosiahol sú znázornené v grafe 6.3. Z tohto grafu môžme vyčítať, že pre niektoré dopytové fotografie bol veľmi úspešný, pre iné zase nie. Je zaujímavé, že systém dokázal pri niektorých dopytových fotografiách vyhľadať

patričné podobné fotografie temer ideálne, a pri iných dopytových fotografiách zase neuspel skoro vôbec. Na obr.6.1 vidíme príklad najúspešnejších a najmenej úspešných dopytových fotografií. Obrázok 6.2 ukazuje najlepšie vyhľadané fotografie pre najúspešnejšiu dopytovú fotografiu. Priemerná hodnota presnosti je 0.205516. V porovnaní so state-of-the-art metódami je to vcelku nízka hodnota a bolo by potrebné do budúcna zvážiť optimalizácie a rozšírenia tohto základného prístupu, aby dosahoval vyššie výsledky.

Obr. 6.1: Najúspešnejšie dopytové fotografie (horný riadok) a najmenej úspešné dopytové fotografie (dolný riadok).

## 6.2 Deep Convolutional Neural Networks

Pre hlboké konvolučné neurónové siete boli navrhnuté 2 prístupy riešenia vyhľadávania fotografií v databáze podľa príkladu. Oba tieto prístupy budú bližšie popísané v nasledujúcich podkapitolách.

### 6.2.1 Prístup 1

Tento prístup spočíva v extrahovaní deskriptorov z DCNN, ktoré následne nahradia SIFT deskriptory v základnom riešení. To znamená, že po extrakcii týchto deskriptorov z DCNN sa pokračuje vytvorením vizuálneho slovníka a prostredníctvom slovníka sa vypočítajú BoW histogramy pre každú fotografiu.

Pri výpočte Bag of Words histogramov došlo k zisteniu, že hodnoty, ktoré sa v jednotlivých histogramoch nachádzajú, nevyzerajú príliš optimisticky. Po analýze a uvažovaní, kde by mohla byť chyba, sa napokon ukázalo, že problém nastal pri zhlukovaní algoritmom k-means. Počas vytvárania vizuálneho slovníka došlo k zisteniu, ktoré zhluky boli priradené ktorým deskriptorom. Pre ukážku uvádzam histogram početnosti priradených zhlukov k deskriptorom (6.4).

Z histogramu môžme usúdiť, že viac ako 70% deskriptorov je priradených k jednému rovnakému zhluku. Preto by bolo vhodné zaoberať sa v tomto prípade inou metódou zhlukovania, pretože k-means algoritmus vyzerá, že nedokáže správne tieto deskriptory nazhlukovať. Nevýhodou ale zostáva, že algoritmy zhlukovania sú časovo náročné. Inou alternatívou môže byť použitie druhej metriky vzdialenosti.

Obr. 6.2: Prvých 6 fotografií vyhľadaných pre najúspešnejšiu vzorovú fotografiu. Vzorová fotografia je najvrchnejšia, vyhľadané fotografie sú v poradí od najlepšej zľava doprava zhora dole.

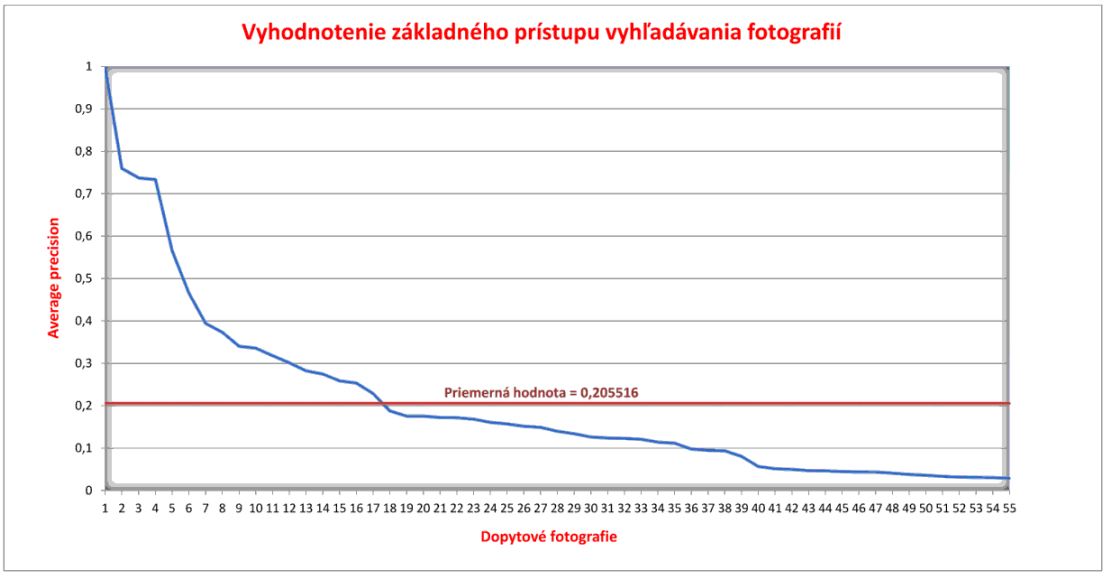
### 6.2.2 Prístup 2

Prístup 2 spočíva v tom, že extrahované deskriptory z DCNN sa ďalej používajú ako náhrada reprezentácie fotografií pomocou BoW, takže tvorba slovníka a výpočet BoW v tomto prístupe, narozdiel od predošlých 2 prístupov, nefigurujú. Úspešnosť vyhľadávania fotografií týmto prístupom je znázornená na grafe 6.5. Na obr. 6.6 sú uvedené znovu najúspešnejšie a najmenej úspešné dopytové fotografie.

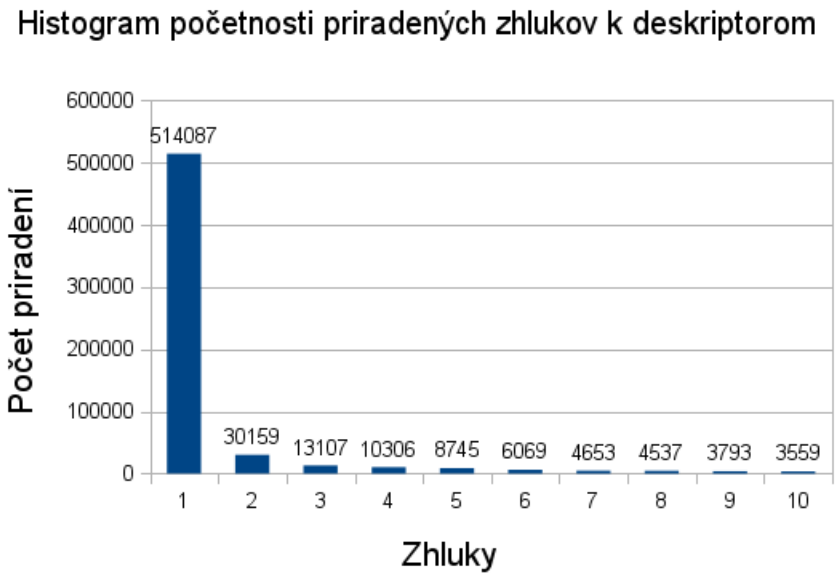
Graf ukazuje, že priemerná hodnota presnosti je 0.092852, čo je prekvapujúco menšie číslo ako pri vyhľadávaní základným prístupom. Dôvodov, prečo je priemerná hodnota nízka oproti state-of-the-art, môže byť viacero. Jedným môže byť napríklad rozdiel v použití metriky. A. Babenko a spol. [2] vo svojich experimentoch používali *L2 vzdialenosť*, zatiaľ čo naša aplikácia využíva *chí kvadrát*.

## 6.3 Návrh testovania

Na účely testovania bude použitý už spomínaný anotovaný dataset The Oxford Buildings Dataset (kap. 2.7). Tento dataset totiž obsahuje tak tréningovú časť s rôznymi mestskými pamiatkami, ako aj dopytovú časť. Príklad dopytových fotografií spolu s príslušnými podobnými tréningovými fotografiami je uvedený v úvodnej časti 2.11.



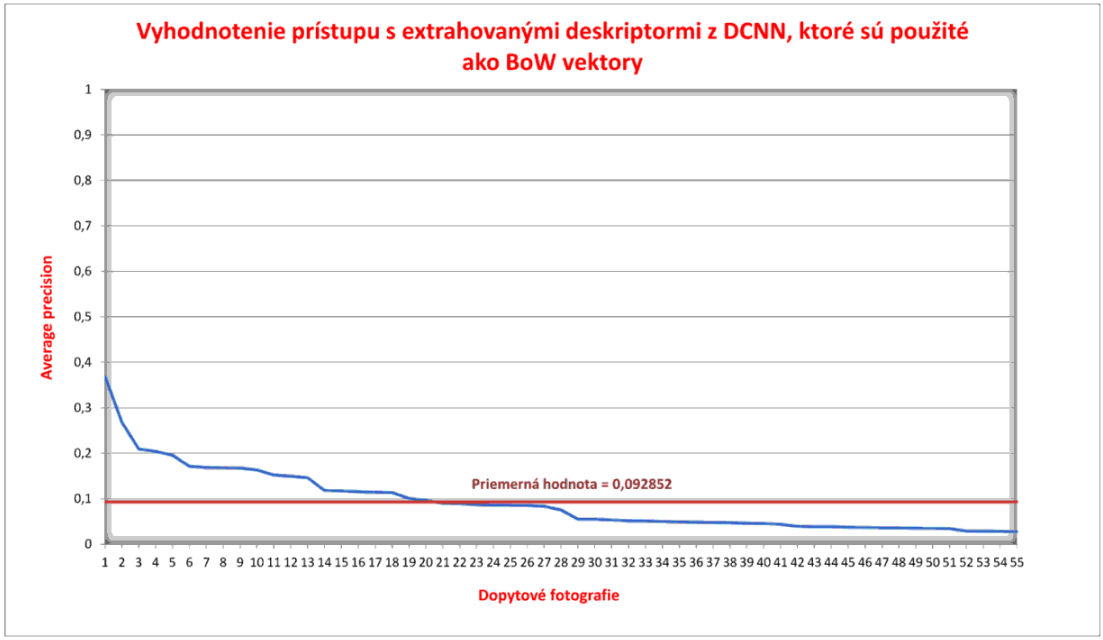

Obr. 6.3: Vyhodnotenie základného prístupu vyhľadávania fotografií v databáze podľa príkladu. Priemerná hodnota presnosti je 0.205516.




Obr. 6.4: Histogram početnosti priradených zhlukov k deskriptorom pri vytváraní vizuálneho slovníka. (Pre prehľadnosť je uvedených len prvých 10 najčastejšie priradených zhlukov.) Počet deskriptorov je 728928.

### 6.3.1 Testy

Testovanie aplikácie bude vykonávané na základe výpočtu priemernej presnosti (*average precision*), čo predstavuje plochu pod krivkou *precision-recall*. Vzorce pre výpočet presnosti




Obr. 6.5: Vyhodnotenie prístupu s extrahovanými deskriptormi z DCNN, ktoré sú použité ako BoW vektory. Priemerná hodnota presnosti je 0.092852.

Obr. 6.6: Najúspešnejšie dopytové fotografie (horný riadok) a najmenej úspešné dopytové fotografie (dolný riadok).

(p) a odozvy (r):

$$p = \frac{correct_results}{all_retrieved_results} \quad r = \frac{correct_results}{all_correct_results_to_the_query_photo} \tag{6.1}$$

Na výpočet priemernej presnosti je použitý voľne dostupný C++ program vytvorený pre oxfordský dataset[1].

[1] http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/compute_ap.cpp

# Kapitola 7

# Záver

Primárnym úmyslom tejto práce bolo navrhnúť a implementovať systém na vyhľadávanie fotografií v databáze podľa príkladu. Jedným z hlavných cieľov bolo porovnať rôzne prístupy vyhľadávania fotografií. Vedľajším cieľom tejto práce bolo naštudovať si problematiku vyhľadávania fotografií, oboznámiť sa s klasickými, ale aj novšími, dnes často používajúcimi sa metódami, ktoré prinášajú (nielen) vo svete počítačového videnia zaujímavé výsledky.

V práci je implementovaný základný prístup k vyhľadávaniu fotografií podľa príkladu. Tento prístup využíva *lokálne príznaky* (konkrétne *SIFT*), *vizuálny slovník*, pri tvorbe ktorého sa využíva algoritmus zhlukovania - *k-means* a tiež histogram početnosti výskytu slov, známy tiež ako *Bag of Words (BoW)*, prostredníctvom ktorého sú reprezentované všetky fotografie v systéme.

Okrem základného prístupu sa práca zaoberá aj dvomi alternatívnymi prístupmi, ktoré využívajú *hlboké konvolučné neurónové siete DCNN* na extrakciu deskriptorov fotografií. Prvý z týchto prístupov pracuje s extrahovanými deskriptormi ako s „náhradou" k SIFT deskriptorom zo základného prístupu. Okrem toho tiež využíva vizuálny slovník a histogram BoW. Druhý prístup, ktorý sa zaoberá tiež extrahovanými deskriptormi z DCNN, pracuje tak, že výstupom z DCNN nahradí BoW reprezentácie fotografií, to znamená, že tento prístup narozdiel od zvyšných nevyužíva ani vizuálny slovník ani BoW.

Celý systém na vyhľadávanie fotografií v databáze podľa príkladu je zložený z viacerých dielčích aplikácií, ktoré navzájom spolupracujú a zdieľajú svoje vstupy i výstupy. Zdrojom fotografií, ktoré som používal v tejto práci na vyhľadávanie, je oxfordský dataset *The Oxford Buildings Dataset*.

Čo sa týka experimentov, základný prístup nášho systému dosiahol priemernú hodnotu presnosti 0.205516. Pri DCNN prístupe 1 sme narazili na problém pri zhlukovaní dát. Algoritmus k-means nedokázal správne nazhlukovať dáta, ukážkou čoho bol uvedený histogram 6.4, ktorý znázorňuje, že 70% všetkých deskriptorov sa nazhlukovalo na jeden a ten istý zhluk. Čo sa výsledkov týka, DCNN prístup 2 bol o niečo bohatší, priemerná hodnota presnosti vyhľadávania fotografií dosiahla úroveň 0.092852. Návrhom, implementáciou a experimentovaním so spomínanými prístupmi bolo splnené zadanie tejto práce.

Do budúcna by bolo určite dobré vytvoriť experimenty komplexne nad viacerými parametrami systému a navzájom ich testovať, ako sa navzájom ovplyvňujú a spolupracujú (napr. viaceré metódy zhlukovania, viaceré metriky vzdialeností, viaceré slovníky rozličnej veľkosti). Budúci vývoj v tejto problematike by sa mohol zameriať tiež hlbšie na širšie využitie DCNN. Príp. porovnať aplikované prístupy s novšími a aktuálnejšími prístupmi/metódami.

# Literatúra


[1] *The OpenCV Reference Manual, v.2.4.9.* Jún 2014, [PDF].

[2] Babenko, A.; Slesarev, A.; Chigorin, A.; aj.: Neural Codes for Image Retrieval. In *Computer Vision - ECCV 2014*, *Lecture Notes in Computer Science*, ročník 8689, editace D. Fleet; T. Pajdla; B. Schiele; T. Tuytelaars, Springer International Publishing, 2014, ISBN 978-3-319-10589-5, s. 584–599, doi:10.1007/978-3-319-10590-1˙38.
URL http://dx.doi.org/10.1007/978-3-319-10590-1_38

[3] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, ročník 110, č. 3, 2008: s. 346 – 359, ISSN 1077-3142, doi:http://dx.doi.org/10.1016/j.cviu.2007.09.014, similarity Matching in Computer Vision and Multimedia.
URL http://www.sciencedirect.com/science/article/pii/S1077314207001555

[4] Bentley, J. L.: Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM*, ročník 18, č. 9, Září 1975: s. 509–517, ISSN 0001-0782, doi:10.1145/361002.361007.
URL http://doi.acm.org/10.1145/361002.361007

[5] Chandran, S.: Introduction to kd-trees. 2002, [PDF].
URL http://www.cs.umd.edu/class/spring2002/cmsc420-0401/pbasic.pdf

[6] Cristian Cibils Bernardes: Assignment 3: KDTree [online].
URL
http://web.stanford.edu/class/cs106l/handouts/assignment-3-kdtree.pdf

[7] Csurka, G.; Dance, C. R.; Fan, L.; aj.: Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, s. 1–22.

[8] Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, ročník 36, č. 4, 1980: s. 193–202, ISSN 0340-1200, doi:10.1007/BF00344251.
URL http://dx.doi.org/10.1007/BF00344251

[9] Glorot, X.; Bordes, A.; Bengio, Y.: Deep Sparse Rectifier Neural Networks.

[10] Grauman, K.; Leibe, B.: Visual Recognition - Excerpt Chapter from Synthesis lecture draft. 2009, [PDF].
URL http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/bag_of_visual_words.pdf

[11] Hinton, G.; Deng, L.; Yu, D.; aj.: Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, ročník 29, č. 6, November 2012: s. 82–97.
URL http://research.microsoft.com/apps/pubs/default.aspx?id=171498

[12] Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; aj.: Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, ročník abs/1207.0580, 2012.
URL http://arxiv.org/abs/1207.0580

[13] Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; aj.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 24, č. 7, Červenec 2002: s. 881–892, ISSN 0162-8828, doi:10.1109/TPAMI.2002.1017616.
URL http://dx.doi.org/10.1109/TPAMI.2002.1017616

[14] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, editace F. Pereira; C. Burges; L. Bottou; K. Weinberger, Curran Associates, Inc., 2012, s. 1097–1105.
URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[15] LeCun, Y.; Bottou, L.; Bengio, Y.; aj.: Gradient-Based Learning Applied to Document Recognition. In *Intelligent Signal Processing*, IEEE Press, 2001, s. 306–351.

[16] LeCun, Y.; Kavukcuoglu, K.; Farabet, C.: Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, May 2010, s. 253–256, doi:10.1109/ISCAS.2010.5537907.

[17] LISA lab.: Convolutional Neural Networks (LeNet) [online].
URL http://deeplearning.net/tutorial/lenet.html

[18] Lowe, D. G.: Object Recognition from Local Scale-Invariant Features. 1999.
URL http://dl.acm.org/citation.cfm?id=850924.851523

[19] Miksik, O.; Mikolajczyk, K.: Evaluation of Local Detectors and Descriptors for Fast Feature Matching.

[20] Philbin, J.; Chum, O.; Isard, M.; aj.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[21] Simonyan, K.; Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] VISTA Lab: An Introduction to Convolutional Neural Networks [online].
URL http://white.stanford.edu/teach/index.php/An_Introduction_to_Convolutional_Neural_Networks

[23] Wu, X.; Kumar, V.; Ross Quinlan, J.; aj.: Top 10 algorithms in data mining. *Knowledge and Information Systems*, ročník 14, č. 1, 2008: s. 1–37, ISSN 0219-1377, doi:10.1007/s10115-007-0114-2.
URL http://dx.doi.org/10.1007/s10115-007-0114-2

[24] Zbořil, F.; Zbořil, F.: *Základy umělé inteligence - studijní opora*. FIT VUT v Brně, 5. 2012, [PDF].

[25] Zhang, D.; Lu, G.: Evaluation of similarity measurement for image retrieval. In *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, ročník 2, Dec 2003, s. 928–931 Vol.2, doi:10.1109/ICNNSP.2003.1280752.