

Image Histogram

Popis projektu

Cílem projektu je vytvoření aplikace, která měří podobnost vstupního obrázku s databází obrázků na základě analýzy jejich histogramů barev.

Další Informace

Obrázek je možno reprezentovat jako množinu bodů a jejich barev. Četnosti jednotlivých barev pak tvoří histogram. Podobnost obrázků lze definovat tak, že 2 obrázky prohlásíme jako podobné, pokud mají podobnou barevnost – histogram. Tímto způsobem dokážeme identifikovat podobné barevné obrázky, nicméně vnitřní sémantika v takové reprezentaci schází (neuvažujeme-li i prostorovou informaci, tj. jaký je vztah z hlediska barevnosti mezi jednotlivými regiony). Histogramy lze chápat jako n -dimenzionální vektory celých čísel a lze na nich tedy definovat podobnost. U libovolné ze zvolených měr je třeba také aplikovat nějakou normalizační techniku pro omezení vlivu velikosti obrázku.

Způsob řešení

- Výsledná aplikace je napsána v C# nad ASP.NET Core a Razor Web Pages.
- Aplikace závisí na dvou balíčcích třetích stran
 - `MathNet.Numerics` pro práci s vektory a maticemi.
 - `SixLabors.ImageSharp` pro práci s obrázky.
- Aplikaci lze kontejnerizovat (docker).

Aplikace obsahuje tyto části:

- Modul extrakce histogramu (složka histogram)
 - Lze konfigurovat počet binů. (default je 16)
 - Počítá se histogram pro HSV a RGB reprezentaci barev.
 - Normalizace probíhá přes dělení celkovým počtem pixelů.
- Modul podobnostní míra pro porovnání dvojice obrázků, tj. jejich histogramů (složka similarity)
 - Implementoval jsem tyto typy měr:
 - KullbackLeiblerDivergence dle přednášek.
 - MinkowskiDistance dle přednášek.
 - Qfd dle [paperu pana Skopala](#).
- Modul databaze obrázku (složka database)
 - Aplikace na začátku nahraje z nakonfigurované lokace obrázky, spočítá jejich histogramy a uloží do paměti.
- Modul identifikace podobných databázových obrázků vzhledem ke vstupnímu obrázku (složka evaluation)
 - Aplikace seřadí podobnosti dotazovaného obrázku vůči databázi podle různých měr a vrátí nejlepšího kandidata včetně informací o delce porovnání.
- Webový interface (složka Pages)
 - Jedna stránka, kde se vloží dotazovaný obrázek a vrátí se kandidátní obrázky včetně vizualizace.

Běh programu

Program běží na adrese `http://imagehistogram.northeurope.azurecontainer.io` (kvůli HW nárokům ale s omezenou databází obrázků). Zde se dá vše vyzkoušet. Deploy skript je `containerize.ps1` (funguje ale jen pro rymeskar, neboť jsou potřeba login credentials..)

Pokud máte zájem o lokální build, pak se musí udělat tři kroky.

1. Build aplikace
 - `docker build -t image_histogram ImageHistogram/.`
2. Zajištění fotografií pro inicializace databáze.
 - Je možné se podívat na můj sdílený disk (skripty connectShare pro namountování disku do systému)
3. Spustit aplikaci s namountovanými obrázky ve složce `"images\train"`
 - Příklad `docker run -p 1234:80 -v c://Users/karymes/source/repos/ImageHistogram/photos:/images -i image_histogram`
 - Docker image očekává obrázky ve složce `/images/train`

Zdrojová data

Databázi obrázků jsem získal na [IEEE dataport](#) hledáním veřejného datasetu. Našel jsem dataset objektů. Ten jsem lehce promázl, ať nemám tisíc obrázků, ale pouze kolem stovky.

V druhé fázi jsem si vybral testovací data, na kterých budu hodnotit přesnost algoritmu. Pro tyto data jsem pokaždé manuálně vybral nejpresnější objekt z databáze. Výsledná reference je uložena v `reference_test.json`. Tyto informace jsou též vidět v projektu `ExperimentRunner`.

Experimentální sekce

- Databáze obsahuje 80 obrázků.
- Testuji na 23 obrázcích.
- Měřím rychlost výpočtu histogramů a podobnostních metrik s vlivem reprezentace barev.
- Porovnávám přesnost jednotlivých metrik s referencí.

Vyšly mi následující data pro vytvoření databáze histogramu obrázků.

```
{
  "InitHistogramTime": "00:00:43.9914860",
  "OverallInit": "00:00:51.5400719",
}
```

Též jsem experimentálně potvrdil, že identita funguje a pro obrázky z databáze se vždy zvolí ten samý jako nejpodobnější.

Pro přesnost mi vyšly následující počty správně najitých podobných obrázků při 16 binech.

```
{
  "Qfd:HSV": 18,
  "Qfd:RGB": 16,
  "KullbackLeiblerDivergence:HSV": 18,
  "KullbackLeiblerDivergence:RGB": 18,
  "MinkowskiDistance2:HSV": 13,
  "MinkowskiDistance2:RGB": 16,
}
```

Metriky pak při výpočtu zabrali následující čas.

```
{
  "MinkowskiDistance2:HSV": "00:00:00.0039239",
  "MinkowskiDistance2:RGB": "00:00:00.0041655",
  "Qfd:RGB": "00:02:15.9346277",
  "Qfd:HSV": "00:02:24.3232844",
  "KullbackLeiblerDivergence:HSV": "00:00:00.0039099",
  "KullbackLeiblerDivergence:RGB": "00:00:00.0030172",
}
```

Pro úplnost jsem též zvolil nastavení o 8 binech.

```
{
  "KullbackLeiblerDivergence:RGB": 16,
  "KullbackLeiblerDivergence:HSV": 17,
  "MinkowskiDistance2:HSV": 13,
  "MinkowskiDistance2:RGB": 16,
  "Qfd:RGB": 12,
  "Qfd:HSV": 15
}
```

- Při použití menšího počtu binů je vidět nepatrné zhoršení.
- Z výsledku je tedy patrné, že výpočet histogramu obecně zabírá největší část výpočtu. Toto je pochopitelné, neboť se jedná o největší počet operací.
- Též je patrné, že nejpomalejší měrou je QFD, která při velkém počtu vzorku dokonce trvá déle než samotný výpočet histogramu.
 - Překvapivě to ale není vynahrazeno lepší přesností.
- Mezi Minkowski a Kullback-Leibler Divergence pak není zásadní rozdíl.
- Obecně mezi reprezentacemi barev není vidět rozdílný výsledek.
- Na přesnost je nepatrným vítězem KullbackLeiblerDivergence.

Tento rozbor výsledků je pouze reprezentativní, pro lepší pochopení chování by měli být provedeny statistické metody rozboru výsledků.

Diskuze

Toto řešení slouží jako proof of concept hledání podobných obrázků za použití jejich histogramů. Aplikace by se dala dále rozšiřovat v každé ze svých komponent.

- Extrakce histogramu
 - SIMD, OpenMP zvýšení performance.
- Podobnostní míra
 - Zvýšení performance
 - Další míry
- Databáze obrázků

- Dynamické přidávání a odebrání
- Identifikace nejpodobnějších
 - Využit jiné metody než pouze porovnání histogramů.
 - Využit možnosti nedělat kompletní průchod databáze, ale approximate, či techniky indexování.
- Webový interface
 - Přidat historii hodnocení
 - Přidat zpětnou uživatelskou referenci

Závěr

Ukázali jsme si, že podobnost dle histogramů je rychlé přiblížení se k podobnému obrázku. Dále jsme si ukázali vliv různých měr na přesnost a rychlost.

Přílohy

Matice *A* z metricky QFD při počtu 2 binů na pixel.

1	0,292893	0,292893	0	1	0,292893	0,292893	0
0,292893	1	0	0,292893	0,292893	1	0	0,292893
0,292893	0	1	0,292893	0,292893	0	1	0,292893
0	0,292893	0,292893	1	0	0,292893	0,292893	1
1	0,292893	0,292893	0	1	0,292893	0,292893	0
0,292893	1	0	0,292893	0,292893	1	0	0,292893
0,292893	0	1	0,292893	0,292893	0	1	0,292893
0	0,292893	0,292893	1	0	0,292893	0,292893	1