

Semestrální práce z předmětu KKY/ZDO - 2014

Rozpoznávání dopravních značek

Autoři - Pavel Rak, Jana Hejdová, Václav Jůna

Zadání práce

- Cílem práce je napsat stroj na automatické rozpoznávání dopravních značek.
 - a. Cílem vaší práce je skript v jazyce python, který bude posouzen automatickým vyhodnocením. Proto je potřeba dbát na předepsanou formu. Hlavní skript musí obsahovat třídu Znacky. Ta má povinnou funkci rozpoznajZnacku().
 - b. Nač je potřeba dát pozor:
 - class Znacky
 - funkce rozpoznajZnacku()
 - případné datové soubory je nutné adresovat relativně pomocí __file__ (viz init v navazujícím příkladu)
 - demo režim

Specifikace úlohy

V jazyku Python vytvořit algoritmus, který vyhodnotí vstupní fotografie, obrázky, a přiřadí jim název značky, respektive číselný kód zastupující danou značku.

Návrh řešení

- Načtení obrázku,
- převedení barevného obrázku na šedotónový obrázek,
- segmentace, zesvětlení, ztmavení obrázku
- převod na jednotnou velikost obrázku,
- volba vektoru příznaků:
 - průměrný jas
 - nekompaktnost
 - výstřednost
 - délka
 - velikost
- natrénování klasifikátoru pomocí zjištěných vektorů příznaků z vzorových dat,
- rozpoznání vstupního obrazu pomocí klasifikátoru.

Popis řešení

Cílem je vytvořit takové příznaky, aby na jejich základě mohlo dojít k rozpoznání dopravních značek s co nejvyšší pravděpodobností shody.

Pomocí nadefinovaných příznaků a trénování množiny obrázků dopravních značek se vytvoří klasifikátor. Následně se pomocí klasifikátoru určí neznámá dopravní značka.

Vytvoření vektoru příznaků

Vstupní obraz je nejprve převeden na šedotónový. Následně se vytvoří vektor příznak pomocí histogramu orientovaných gradientů (HOG).

Histogram orientovaných gradientů (HOG)

Histogramy orientovaných gradientů vycházejí z výpočtu gradientu obrazové funkce (například pomocí Sobelova operátoru). Gradient obrazové funkce je vždy charakterizován dvěma složkami, kterými mohou být směr a velikost gradientu. Klasický HOG pak vznikne spočítáním histogramu gradientů v určité oblasti obrazu. Histogram vždy zachycuje sílu gradientu v dané oblasti a určitém směru - pro každý bod obrazu se akumuluje příspěvek odpovídající velikosti gradientu do binu histogramu odpovídajícímu směru gradientu. Gradienty bývají velmi silné na siluetách objektů a podobných výrazných hranách, a proto jsou Histogramy orientovaných gradientů vhodné pro objekty jako jsou chodci, dopravní značky a podobně.

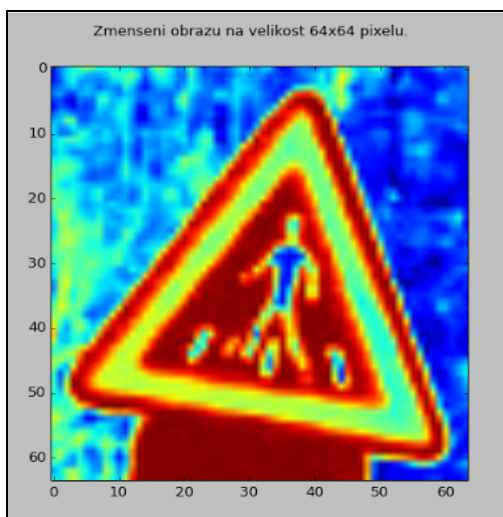
1. Je zjištěn rozměr obrazu, vypočítán poměr výšky a šířky a velikost obrazu (hodnota velikosti je přibližně velikost obrázku 64x64 pixelů).
2. Podle velikosti a poměru je obraz transformován na jeden z formátů dle následující tabulky:

VELIKOST	ORIENTACE	CÍLOVÝ FORMÁT
větší než 4100 pixelů	na výšku	64x32 pixelů
větší než 4100 pixelů	na šířku	32x64 pixelů
větší než 4100 pixelů	čtvercová	64x64 pixelů
menší než 4100 pixelů	na výšku	32x16 pixelů
menší než 4100 pixelů	na šířku	16x32 pixelů
menší než 4100 pixelů	čtvercová	32x32 pixelů

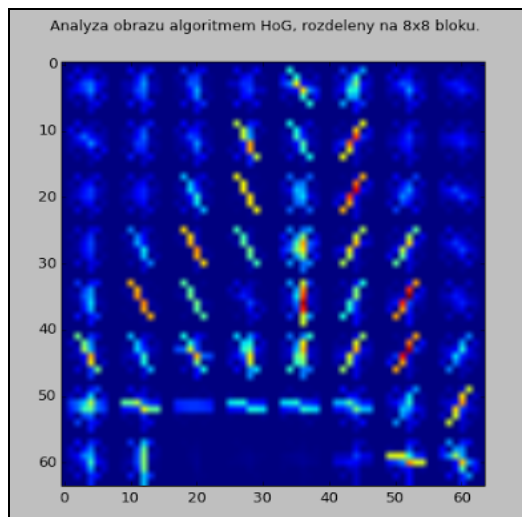
Tímto bylo dosaženo menší deformace než kdyby se všechny obrazy bez ohledu na jejich velikost transformovaly na jednotný formát.

3. Nad transformovaným obrazem je vytvořen HOG. Jeho výstup je přidán do vektoru příznaků.

Na následujících obrázcích je dopravní značka "Pozor, přechod pro chodce". Na 1. obrázku je zmenšený vstupní obraz. Na druhém obrázku je obraz po aplikaci HOG algoritmu. Na posledním obrázku je vstupní obrázek ve formátu *.jpg (70x70 px).



1. obrázek



2. obrázek



3. obrázek

Podmínky funkčnosti skriptu

- Algoritmus nebude dokonale funkční u velmi nekvalitních snímků dopravních značek (př. snímek pořízený bez osvětlení v noci)
- Aby byla zajištěna funkčnost algoritmu je nutné, aby byl dostatek dat, která budou použita k natrénování klasifikátoru.

Jak spustit program

Pro spuštění je nutné nejprve otevřít algoritmus ve vhodném programu (Spyder, Python (x,y)). Dále je nutné programu navolit cesty ke zdrojovým datům určeným k trénování klasifikátoru a k testovacím datům (v poslední části kódu). Uživatel musí mít nainstalovány knihovny matplotlib, skimage, skimage.feature.hog, glob, os, numpy, pickle, sklearn, time a logging. Poté stačí skript spustit v aplikaci pomocí GUI nebo spustit ve složce, kde se nachází skript.

Uživatel může povolit metodu demo a zobrazit si jednotlivé kroky klasifikace obrázku (demo:False X demo:True - 65 řádka kódu).

Celkové zhodocení

Pomocí výše popsaného postupu bylo dosaženo shody s 76,45% pravděpodobností při trénování na sadě zdo2014-training3 a otestování na sadě zdo2014-training1. Shody s větší pravděpodobností by mohlo být dosaženo, kdyby se nám podařilo z obrazu vysegmentovat dopravní značku. Následně bychom mohli vypočítat příznaky jako nekompaktnost, výstřednost nebo velikost značky. Pro tvorbu příznakového vektoru bylo použito metody, která nebyla předem predikována v navrhovaném řešení. Metody, které jsou v navrhovaném řešení, dostatečně nezaručovaly tak vysokou úspěšnost, jako použití histogramu orientovaných gradientů.

Podle zkušeností se příliš nevyplatí pro předzpracování použít prahování ani filtraci např. mediánem, neboť se tím sníží úspěšnost určení značky. Nevhodnost vychází nejspíše z různých kvalit vstupních obrázků, jejich ostrosti a velikostí.

Závěr

Díky předmětu ZDO a této semestrální práci jsme se více seznámili s jazykem Python. Tento jazyk se používá např. při programování nástrojů pro ArcGIS. Vzhledem k oboru, který studujeme, věříme, že by se nám znalosti ohledně jazyka Python mohly někdy v budoucnu hodit.

Zdroje

- [1] Zpracování digitálního obrazu (ZDO). Katedra Kybernetiky [online]. 2014 [cit.2014-05-29]. Dostupné z: <http://www.kky.zcu.cz/cs/courses/zdo>.
- [2] Object detection overview. *MEDUSA* [online]. 2011 [cit. 2014-05-29]. Dostupné z: http://medusa.fit.vutbr.cz/wiki/index.php?title=Object_detection_overview&oldid=254.