

Abstrakt

Odborný článok na tému „Implementácia a využívanie modelov Počítačového videnia v praxi“ pozostáva z 4 kapitol. Prvá kapitola je venovaná používaniu počítačového videnia s využitím programovacích jazykov ako Python a C++ a knižnice OpenCV. Druhá kapitola je venovaná druhom a trénovaniu modelov počítačového videnia. Tretia kapitola bude venovaná výhodám a nevýhodám počítačového videnia. Obsahom štvrtej kapitoly je využitie počítačového videnia v praxi.

1 Počítačové videnie

Počítačové videnie je automatizovaná extrakcia informácií z obrázkov. Informáciu môže predstavovať čokoľvek od 3D modelov, videa alebo obrázka. Niekedy sa počítačové videnie snaží napodobňovať ľudské videnie, inokedy využíva dáta alebo štatistiky. Praktické počítačové videnie je mix programovania, modelovania a matematiky. Matematická časť pomáha pochopiť špecifické algoritmy ktoré sú používané.

2 Programovacie jazyky a knižnice počítačového videnia

2.1 Počítačové videnie a Python

Jeden z najpopulárnejších ak nie najpopulárnejší programovací jazyk pre prácu s počítačovým videním je Python. Python je interpretovaný, interaktívny programovací jazyk s dobrou podporou na spravovanie obrazu. Od verzie 2.6 máme pri práci s Pythonom k dispozícii väčšinu balíkov ktoré pri práci potrebujeme. Od verzie 3.x.x sa zmenila syntax pre ale kompatibilita s balíkmi pre počítačové videnie ostala rovnaká.

2.2 NumPy

NumPy je knižnica pre programovací jazyk Python. Je užitočná na prácu s vektormi a maticami a ich operáciami na vysokej úrovni, ktoré sú potrebné na fungovanie počítačového videnia. Dané typy sú reprezentované pomocou polí.

2.3 Ďalšie knižnice

V počítačovom videní sa stretneme aj s inými knižnicami jazyka Python. Knižnica Matplotlib je využívaná pre vizualizáciu výsledkov. Knižnica SciPy je využívaná keď potrebujeme využívať pokročilejšie matematické funkcie alebo algoritmy.

2.4 OpenCV

OpenCV je open-source knižnica napísaná v C++ ,ktorá obsahuje algoritmy a funkcie zamerané na počítačové videnie.Najčastejšie sa používa s programovacími jazykmi Python,C a C++ ,ale v novších verziach už bola vydaná podpora aj na JavaScript.Má modulárnu štruktúru o znamená že obsahuje niekoľko hlavných knižníc.

- Hlavné knižnice OpenCv sú:
 1. Image Processing-modul na spracovávanie obrázkov
 2. Video Analysis-modul na spracovávanie videí
 3. Core functionality-modul definujúci základné dátové štruktúry a funkcie ,ktoré využívajú ostatné moduly
 4. Object Detection-detekcia objektov a instance preddefinovaných tried (napr. tvár, oči, ľudia, autá...)
 5. High-level GUI – ľahko použiteľné rozhranie na jednoduché UI funkcie.
 6. Camera Calibration and 3D Reconstruction – kalibrácia kamery, odhad polohy objektu, 3D rekonštrukcia.
 7. ...

3 Modely a ich trénovanie

3.1 Typy modelov

Rozne typy modelov nám pomáhajú riešiť rôzne typy problémov.Aké predmety sú na obrázku? Kde sú predmety na obrázku?

- Klasifikácia obrázkov: Snaží sa identifikovať najvýznamnejšiu triedu objektu.Triedu môžeme chápať ako označenie,napr. pri identifikácii topánok(bežecká obuv,...)
- Detekcia objektov: Používa sa keď je dôležitá poloha objektu.Vracia množstvo suradníc nazývaných ohraničujúci rámček.Príklad takéhoto modelu môže byť model na detekciu osôb(AlwaysAI/mobilenet).
- Segmentácia obrazu: Keď je dôležitý presný tvar objektu využijeme segmentáciu obrazu.Pre presnú segmentáciu klasifikuje každý pixel.
- Detekcia orientačných bodov: Využíva sa na určenie dôležitých bodov,ktoré zachytávajú dôležité prvky objektu.Môžeme použiť model na odhad pózy(AlwaysAI/human-pose).

3.2 Typy údajov pre školenie počítačového videnia

Generovanie množiny údajov:Na učenie modelu počítačového videnia potrebujeme kvalitné údaje.Za kvalitný údaj sa považuje ten čo je podobný údajom z reálneho sveta,ktoré budú použité na trénovanie modelu.

- Typy generovania množín údajov: Podľa toho či chceme nech model vykonáva jednu konkrétnu úlohu alebo chceme všeobecný model potrebujeme aj rôzne typy údajov. Ak chceme konkrétny model budeme potrebovať na jeho tréning špecifické dáta a obrázky ,ktoré sú čo najviac podobné danej špecifikácii.
- Použitie existujúcej anotovanej množiny údajov: Výrazne môžu skrátiť čas potrebný na trénovanie modelu ale zároveň nemáme takú kontrolu nad kvalitou údajov. Populárne verejné súbory údajov sú napr. PASCAL Visual Object Classes (VOC), ImageNet, ...
- Použitie vlastných údajov : vlastný súbor údajov tvorený z voľne dostupných videí a obrázkov online. Na rozdiel od existujúcej anotovanej množiny ich treba anotovať pred použitím.
- Použitie digitálne generovaného súboru: Používa sa ak niesme sami schopný zhromaždiť dostatok údajov. Vedie k lepšiemu výkonu.

3.3 Trénovanie modelov

Zhromaždené a anotované údaje sa použijú ako vstup pre tréning modelu. Prechádzajú a porovnávajú sa údaje až kým nedosiahneme dobrý výkon, na základe špecifikácií pre každý typ modelu.

- Transfer Learning: Využíva poznatky zo všeobecného tréningu a aplikuje ich na iné špecifickejšie možnosti.
- Testovacia aplikácia: Po vytvorení modelu na ňom otestujeme aj nové údaje, a zistíme či model reaguje a pracuje ako očakávame. Nejde o automatizovaný test ale jeho výhoda je že umožňuje rýchlo identifikovať nedostatky alebo prípady kedy model nefunguje spoľahlivo. Potom na model použijeme situácie prípadov kedy nefungoval úplne správne a opravíme to.

Aj text môže byť prezentovaný ako obrázok. Stane sa z neho označný plávajúci objekt. Po vytvorení diagramu zrušte znak % pred príkazom `\includegraphics` označte tento riadok ako komentár (tiež pomocou znaku %).

Obr. 1: Rozhodujúci argument.

4 Výhody a nevýhody počítačového videnia

Ako každá ina technológia aj počítačové videnie má svoje výhody a nevýhody.

- Porovnanie s ľudským videním
 - ľudské videnie: zdroj obrazu → oko → kognitívny systém
 - počítačové videnie: zdroj obrazu → kamera → počítač
- Výhody
 1. Uľahčuje množstvo procesov

2. Úplné frekvenčné spektrum pre získanie obrazov
3. Jednoduchý a rýchly spôsob získavania údajov
4. Generovanie presných a presných údajov
5. Finančne efektívne

- Nevýhody

1. Treba spravovať veľmi veľké množstvo informácií
2. Citlivé na osvetlenie a rôzne iné javy
3. Realny svet transformujeme 3D na 2D
4. Je potrebné veľké množstvo pamäte
5. Veľké množstvo nadbytočných údajov

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 2.4), a potom na ešte nejaké (časť 2.4).¹

Môže sa zdať, že problém vlastne nejestvuje [Cop99], ale bolo dokázané, že to tak nie je [CHE05, CK05]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [SEI]. Dôležité veci možno *zdôrazniť kurzívou*.

Veľmi dôležitá poznámka. Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

5 Využitie počítačového videnia

6 Záver

Literatúra

- [CHE05] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [CK05] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.
- [Cop99] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [SEI] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. http://www.sei.cmu.edu/productlines/frame_report/.

¹Niekedy môžete potrebovať aj poznámku pod čiarou.