



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

MUL

MULTIMÉDIA

Dokumentace

Aplikace pro komunikaci s webkamerou

Autor:
Petr Buchal

Login:
xbucha02

17. dubna 2019

1 Úvod

Jako zadání projektu jsem si vybral implementaci aplikace pro komunikaci s webkamerou. Mým cílem je vytvořit aplikaci, která bude mít dva módy. První mód bude demonstrovat algoritmus zpracování obrazu pro detekci pohybu. Druhý mód bude hlídat, zdali na snímcích kamery nedošlo k pohybu. Pokud k detekci pohybu dojde, aplikace upozorní uživatele vybraným způsobem. Projekt jsem se rozhodl vypracovat sám.

2 Zadání

Vytvořte aplikaci, která cyklicky komunikuje s kamerou a periodicky z ní snímá obrázky. Tyto obrázky vyhodnotí a v případě, že detekuje podezřelý pohyb/změnu, obrázek uloží a volitelně informuje uživatele zvukovým signálem. Případně upozorní na oblast, kde došlo k detekci pohybu. Aplikace musí upozorňovat na nalezení pohybu/změny vhodným, například zvukovým signálem.

Možnosti:

- výběr regionu
- ukládání na vzdálený počítač
- notifikace uživatele e-mailem
- detekce osob

3 Implementace

Aplikaci jsem se rozhodl implementovat v jazyce Python. K detekci pohybu v obraze poté využívám výhradně knihovnu OpenCV.

3.1 Zpracování obrazu

Zpracování každého snímku probíhá následovně. Nejdříve použiji metodu Background-SubtractorMOG2, která používá GMM na modelování pozadí snímku [4]. Tato metoda je výhodná, protože se dokáže adaptovat na změny v osvětlení a nové statické

objekty. Takže například při posunutí kamery je po krátké době detektor opět schopen fungovat. Obraz získaný z této metody obsahuje relativně hodně nežádoucího šumu, který je podobný typu šumu Sůl a pepř. Ten se nejčastěji odstraňuje pomocí mediánového filtru [5], na obraz je tedy aplikován mediánový filtr. Mediánový filtr ale dále způsobuje výskyt oblastí s různými stupni šedi, což je také nežádoucí. Stupně šedi se odstraní pomocí thresholdingu. Dále se ještě jednou aplikuje mediánový filtr.



Obrázek 1: GUI demo módu aplikace.

Cílem předchozích operací je segmentace pohybujících se objektů. Při segmentaci se pak často využívají různé morfologické operace [2] a ty jsou dále využity rovněž i v tomto projektu. Na obraz je aplikována dilatace za účelem nalezení popředí, dále pak mediánový filtr opět na odstranění vzniklého šumu a následně morfologická operace uzavírání, která zvýrazňuje nalezené objekty. V takto upraveném obrazu se hledají kontury. Následně jsou za objekt uznány kontury, které obsahují počet bodů překračující dynamicky stanovenou hranici. Ta je stanovena na základě procentuálního zastoupení bílých pixelů v segmentovaném obraze. Efekty jednotlivých operací na obraz jsou k dispozici na obrázku 1.

3.2 Technologie potřebné pro spuštění programu

- Standardní knihovny jazyka Python 3.5.4
- Knihovna **OpenCV**
- API **CVUI** nad knihovnou OpenCV
- API **imutils** nad knihovnou OpenCV

4 Obsluha programu

Aplikace je spustitelná z příkazové řádky. Pro využívání emailových notifikací a komunikace se vzdáleným serverem je třeba nakonfigurovat soubory *email_settings.json* a *scp_settings.json*, tomu se věnují podkapitoly 4.2 a 4.3.

4.1 Argumenty a příklady spuštění

<code>-mode {demo, time_capsule}</code>	slouží pro výběr módu aplikace
<code>-source VIDEO.MP4</code>	je vybrán demo mód aplikace a argument není zadán, zpracovává se obraz z webkamery, je vybrán demo mód aplikace a argument je zadán, zpracovává se soubor VIDEO.MP4, jinak nemá efekt
<code>-time_jump SECONDS</code>	udává po kolika sekundách se kontroluje, zdali došlo k pohybu
<code>-email_notification</code>	slouží pro zapnutí emailových notifikací
<code>-copy_to_server</code>	slouží pro zapnutí kopírování fotek s pohybem na vzdálený server
<code>-sound_notification</code>	slouží pro zapnutí zvukového upozornění
<code>-save_records</code>	slouží pro zapnutí ukládání fotek s pohybem na disk
<code>-no_display</code>	slouží pro zapnutí běhu pouze v příkazovém řádku, tedy bez displeje

- **python main.py -m demo** - spustí demo detekce pohybu nad živým videem z webkamery
- **python main.py -m demo -source example.mp4** - spustí demo detekce pohybu nad videem uloženým v počítači
- **python main.py -m time_capsule -time_jump 1 -email_notification -sound_notification** - spustí detekci pohybu nad živým videem z webkamery v intervalech 1 sekundy a při detekci pohybu, odešle emailovou notifikaci a přehraje výstražný zvuk
- **python main.py -m time_capsule -time_jump 5 -save_records -copy_to_server** - spustí detekci pohybu nad živým videem z webkamery v intervalech 5 sekund a při detekci pohybu, uloží obraz s pohybem a zkopíruje ho dále na vzdálený server

- **python main.py -m time_capsule --time_jump 5 --no_display** - spustí detekci pohybu nad živým videem z webkamery v intervalech 5 sekund, přičemž se nezobrazuje snímáný obraz na displeji

4.2 Nastavení emailových notifikací

Jako hodnotu pole *sender_email* nastavte gmail adresu ze které se budou notifikační zprávy odesílat. Jako hodnotu pole *sender_password* nastavte heslo gmail adresy ze které se budou notifikační zprávy odesílat. Do pole *recievers* zadejte email adresy na které se budou notifikační zprávy odesílat. Dále je třeba v gmail účtu odesílatele povolit přístup méně zabezpečených aplikací, viz [1].

4.3 Nastavení komunikace se vzdáleným serverem

Jako hodnotu pole *user* nastavte vaše uživatelské jméno. Jako hodnotu pole *server* nastavte adresu vzdáleného serveru. Do pole *path* zadejte cestu, kam se mají snímky zachycující pohyb ukládat. Dále je třeba mít nastavené přihlašování na server pomocí RSA klíče, tedy bez hesla, viz [3].

Reference

- [1] Povolení přístupu k účtu méně zabezpečeným aplikacím - nápověda Účet google. <https://bit.ly/2V9Htpu>.
- [2] Afzal Ansari. Image segmentation using morphological operations in python. <https://bit.ly/2IqY1mU>.
- [3] Václav Blažej. Přihlášení přes ssh bez hesla. <https://bit.ly/2V9Htpu>.
- [4] Ph.D. Hamad Odhabi. Motion detection using opencv in python. <https://bit.ly/2ICZwxV>, November 2017.
- [5] Wikipedia. Salt-and-pepper noise — Wikipedia, the free encyclopedia. <https://bit.ly/2IoKs7z>, 2019. [Online; accessed 17-April-2019].