

Дигитално процесирање на слика

Тема:

Детекција на движење кај безбедносни камери

Изработил:

Трајче Проданов - 221164

Содржина

Вовед – 3

Проблемот кој го решава системот - 3

Преглед на Системот – 4

Детали за Имплементација – 5

Краток опис на алгоритмите - 6

Конфигурација на Системот - 7

Употреба на Системот - 8

Архитектура на Системот - 9

Инсталација и Поставување на Системот - 11

Потенцијални Предизвици и Решенија - 13

Идни подобрувања - 14

Заклучок - 15

1. Вовед

Оваа документација го опишува системот за безбедност кој користи OpenCV за откривање на луѓе во видео стримови. Системот има три различни имплементации за откривање: машинско учење, хистограми на градиенти (HoG), и позадинска субтракција. Овој систем функционира со процесирање на видео фрејмови и откривање на луѓе во реално време.

2. проблемот што го решава системот

Во денешното брзо развивачко опкружување, безбедноста на домот и имотот станала приоритети за многу луѓе. Системот за безбедност што се развива нуди ефикасно решение за следниве проблеми:

1. Проблеми со Детекција на Движење

- Стандардните системи за детекција на движење често не можат да ги идентификуваат сложените активности или да ги разликуваат лажните аларми. Системот за безбедност користи три различни методи за детекција (машинско учење, хистограм на градиенти и одземување на позадината), што ја зголемува прецизноста и ги намалува лажните аларми.

2. Недостаток на Автоматизирани Известувања

- Многу системи не нудат веднаш известување на корисниците за детектирани настани. Овој систем интегрира известувања преку Pushover и во многу лесно може да се додадат други имплементации за известувањата, како поддршка за е-пошта, СМС, што овозможува брза и ефикасна комуникација со корисниците.

3. Ограничена Модуларност и Скалираност

- Традиционалните системи често се ограничени на еден уред и не можат лесно да се прошируваат. Со разделување на системот на повеќе Raspberry Pi уреди со индивидуални камери, ова решение овозможува поголема модуларност и скалабилност, што го прави лесно да се прилагоди на различни потреби и околина.

4. Недостаток на Интеграција со Други Уреди

- Многу системи не можат да се интегрираат со други IoT уреди. Овој систем планира да вклучи поддршка за интеграција со паметни уреди, како што се светла и аларми, што овозможува целосно автоматизирана безбедносна реакција.

Системот за безбедност обезбедува иновативни решенија за овие проблеми, што го прави идеален избор за современи потреби за безбедност и мониторинг.

3. Преглед на Системот

Системот функционира со процесирање на видео фрејмови и откривање на луѓе. Кога ќе се детектира човек, системот автоматски испраќа известување преку Pushover за мобилни известувања и го зачувува видеото за подоцнежнo прегледување. Корисниците можат да пристапат до снимките и да ги филтрираат според датум, да го гледаат живиот пренос преку Веб интерфејс, да додаваат нови камери, да уредуваат постоечки, како и да ги вклучуваат или исклучуваат известувањата. Системот е флексибилен и може да се прилагоди за различни конфигурации на камерите.

4. Детали за Имплементација

Системот за безбедност користи три различни методи за откривање на луѓе:

1. **Машинско Учење:** Оваа имплементација користи претходно обучен модел за откривање на луѓе во видео стримови. Моделот е трениран на големо податочно множество и може да препознае различни карактеристики на човечка фигура.
2. **Хистограми на Градиенти (HoG):** Оваа техника ги анализира градиентите во фрејмовите на видеото за да открие присуство на луѓе. HoG е познат по својата ефективност во откривање на објекти и е широко користен во компјутерската визија.
3. **Позадинска Субтракција:** Оваа метода ја анализира разликата помеѓу тековниот фрејм и позадината за да открие промени. Кога ќе се детектира значителна промена, системот може да идентификува присуство на човек.

Секој од овие методи се користи според конфигурацијата на камерите и специфичните барања на корисникот. Системот е дизајниран да биде модуларен, што овозможува лесно додавање нови методи за откривање или замена на постоечките.

5. Краток опис на алгоритмите

1. HOGDetection

- **Опис:** Овој модел користи методот на Хистограм на Ориентирани Градиенти (HOG) за детекција на луѓе во рамката. Применува претходно обучен SVM детектор специјално за препознавање на човечки фигури. Откако ќе се откријат луѓето, тие се обележани со зелени правоаголници. Патем, на рамката се прикажуваат статусот на детекцијата и вкупниот број на откриени луѓе. Овој метод е корисен за детектирање на луѓе во различни околии и под различни агли. Овој модел е оптимизиран и за системи со послаба процесирачка моќ.
- **Функција:** Детектира луѓе користејќи HOG карактеристики. Применува зелени правци за обележување на откриените луѓе и покажува информација за статусот на детекцијата и вкупниот број на откриени лица на рамката.

2. MLDetection

- **Опис:** Овој модел користи MobileNet модел со Caffe рамка за детекција на објекти. Сликата од влезот се обработува преку длабока невронска мрежа за идентификација на луѓе базирана на нивните уверливости. Откриените луѓе се обележани со зелени правоаголници на рамката. Моделот е ефикасен за детектирање на луѓе со уверливост над 0.5, што го прави многу прецизен за ситуациите каде што се бара висока точност. Овој модел бара повеќе процесирачка моќ.
- **Функција:** Детектира луѓе користејќи MobileNet модел. Рамката се означува со зелени правци околу детектираните поединци, ако уверливоста е над границата.

3. MogDetection

- **Опис:** Овој модел користи позадински одземач (MOG2) за идентификација на движење во рамката. Применува разлика помеѓу сегашната рамка и позадинскиот модел за да се

детектираат движечките објекти. Движењето се флагува ако бројот на движечки пиксели го надмине поставениот праг. Откриеното движење се обележува со зелени правоаголници околу контурите, а минималната големина на контурите што се земаат предвид е 500 пиксели. Овој метод е особено корисен за детектирање на непланирани движења или активности во просторот.

- **Функција:** Детектира движење преку споредба на сегашната рамка со позадинскиот модел. Откриеното движење се обележува со зелени правоаголници околу областите со значително движење, што помага во визуелизацијата на активностите во сцената.

6. Конфигурација на Системот

Кога апликацијата се стартува, таа ги чита податоците за камерите од SQLite базата на податоци. Базата содржи информации за секоја камера, вклучувајќи го и нејзиниот „source“, кој може да биде било каков видео стрим, дури и постоечко видео, тип на детекција што треба да се користи, интервали за снимање, и поставки за нотификации. Откако камерите ќе се вчитаат, системот ги иницијализира со соодветните конфигурации.

Поставки за Камерите:

- **Source:** Било каков видео стрим (rtmp, rstp, hls...) од кој се примаат видео стримовите.
- **Тип на Детекција:** Корисникот може да избере кој метод на детекција да се користи за секоја камера.
- **Нотификации:** Опции за вклучување или исклучување на мобилните нотификации преку Pushover.

Иницијализација на Камерите: Системот автоматски ги иницијализира камерите при стартување. Секоја камера е врзана за посебна нишка кој ја контролира нејзината обработка. Ова овозможува паралелно обработување на повеќе видео стримови, без да дојде до блокирање или загуба на податоци.

Flask endpoints: Корисниците можат да ги активираат или деактивираат камерите преку endpoints кои се поставени во Flask преку API или директно преку веб интерфејсот. Овие endpoints овозможуваат брзо и лесно управување со камерите, како и промена на нивните конфигурации во реално време.

7. Употреба на Системот

Системот е дизајниран да биде лесен за користење и управување преку frontend интерфејс кој е развиен со React SPA. Корисниците можат да пристапат до различни функционалности на системот преку интуитивен и кориснички ориентиран интерфејс.

Главни Функционалности:

1. **Гледање пренос во живо:** Корисниците можат да гледаат видео стримови од сите камери кои се поврзани со системот. Ова е корисно за моментален надзор и проверка на ситуации во реално време.
2. **Прегледување логови:** Сите настани кои се детектирани од камерите се зачувуваат во логови. Корисниците можат да ги прегледуваат логовите, да филтрираат според датум, и да ги анализираат снимките од настаните.
3. **Додавање и уредување камери:** Системот овозможува лесно додавање на нови камери, како и уредување на постоечките. Корисниците можат да го изменат изворот, типот на детекција, и другите поставки за секоја камера.
4. **Управување со нотификации:** Корисниците можат да вклучат или исклучат нотификации според нивните потреби. Нотификациите се испраќаат преку Pushover и можат да бидат прилагодени за секоја камера поединечно.
5. **Армирање и дезармирање:** Преку специјален дел од интерфејсот, корисниците можат да го армираат (вклучат) или дезармираат (исклучат) системот. Оваа функција е особено корисна кога корисникот сака да го паузира системот за одреден временски период.

Кориснички Интерфејс: Интерфејсот е дизајниран да биде едноставен и јасен, со интуитивни контроли за управување со сите аспекти на системот. Сите функции се достапни преку главното мени, а логовите и преносите се прикажани во реално време.

8. Архитектура на Системот

Архитектурата на системот е дизајнирана да биде модуларна и скалабилна, со јасно дефинирани компоненти кои соработуваат за да го обезбедат целокупното функционирање на системот. Главните компоненти се:

1. **Видео обработка и детекција:**
 - **OpenCV:** Јадрото на системот за обработка на видео. OpenCV ги прима фрејмовите од камерите, ги обработува, и ги подготвува за детекција.
 - **Модул за Детекција:** Трите различни имплементации за детекција (машинско учење, HoG, позадинска субтракција) се дел од овој модул. Секој фрејм се праќа на анализирање преку избраниот алгоритам со помош на техника наречена “Dependency injection”.
2. **Складирање на податоци:**
 - **SQLite База на Податоци:** Системот користи SQLite за складирање на информации за камерите, конфигурациите, и логовите. Базата е избрана заради нејзината леснотија на користење и брзина.
 - **Логирање:** Сите настани и аларми се складираат во логови кои можат да се прегледуваат подоцна. Секој запис вклучува време, камера, и тип на детекција.
3. **Фронтенд интерфејс:**
 - **React SPA:** Овој интерфејс обезбедува интуитивен начин за управување со системот. Корисниците можат да го гледаат преносот во живо, да ги прегледуваат логовите, и да управуваат со камерите преку овој интерфејс.
 - **Интеракција со Бекендот:** React SPA комуницира со Flask преку API повици за да добие информации од базата и да ги прати командите за управување со камерите.
4. **Бекенд сервис:**
 - **Flask:** Овој микро-фрејмворк се користи за управување со HTTP барањата, управување со камерите, и процесирање на командите од фронтендот. Flask исто така служи како

интерфејс помеѓу базата на податоци и корисничкиот интерфејс.

- **API endpoints:** Системот обезбедува сет од **API endpoints** кои овозможуваат комуникација со фронтендот и управување со различни аспекти на системот.

5. Известувања:

- **Pushover:** Оваа услуга се користи за испраќање мобилни нотификации кога системот ќе детектира човек. Нотификациите можат да бидат прилагодени и испратени до специфични корисници.

6. Паралелизам и Скалирање:

- **Мултитрединг:** Системот користи мултитрединг за да обработува повеќе видео стримови паралелно, овозможувајќи поефикасно и брзо обработување на големи количини на податоци.
- **Модуларност:** Системот е дизајниран да биде лесно скалабилен, што овозможува додавање на нови камери и нови модули за детекција без значителни промени во основната архитектура.

9. Инсталација и Поставување на Системот

Зависности: За да го поставите системот, потребно е да ги инсталирате следните модули:

- **Python 3.8+:** Основниот програмски јазик во кој е напишан системот.
- **OpenCV:** За обработка на видео фрејмовите и детекција на објекти.
- **Flask:** Микро-фрејмворк за бекенд управување и API крајни точки.
- **SQLite:** Лесна и брза база на податоци за складирање на информации за камерите и логовите.
- **Pushover:** За испраќање на мобилни нотификации (потребна е регистрација и API клуч).

Чекори за Инсталација:

1. Клонирање на Репозиториумот:

```
git clone https://github.com/prodanov17/opencv-security-system.git
cd opencv-security-system
cd backend
```

2. Инсталација на зависности: Користете `pip` за да ги инсталирате потребните библиотеки:

```
pip install -r requirements.txt
mv .env.example .env
```

3. Конфигурација на Камерите: камерите се конфигурираат преку веб интерфејсот, а дополнително во `config.json` фајлот има конфигурации

4. Поставување на Pushover API: Внесете го вашиот Pushover API клуч во `.env` фајлот:

```
PUSHOVER_API_KEY=your_api_key

PUSHOVER_USER_KEY=your_user_key
```

5. Стартување на Системот: Започнете го Flask серверот:

```
python app.py
```

Ова ќе го стартува бекендот и ќе ја иницијализира комуникацијата со камерите.

6. Стартување на фронтендот:

```
cd frontend  
npm install && npm run dev  
mv .env.example .env
```

7. Пристапување до Фронтендот: Отворете го вашиот веб прелистувач и одете на:

```
http://localhost:5174
```

Тука ќе можете да го користите корисничкиот интерфејс за управување со системот.

Тестирање на Системот: За да го тестирате системот, можете да вклучите тест камери со снимање на видео или преку поставување на IP камери. Проверете дали системот правилно ги детектира објектите и дали се испраќаат нотификациите преку Pushover.

9. Потенцијални Предизвици и Решенија

Во процесот на развој и користење на системот за безбедност, можат да се појават различни предизвици. Оваа секција ги опишува најчестите предизвици со кои корисниците може да се соочат и ги предлага соодветните решенија.

1. Проблеми со Детекција во Слабо Осветлување:

- **Предизвик:** Камерите може да имаат проблеми со детекција на луѓе во услови на слабо осветлување.
- **Решение:** Користете камери со подобра сензитивност на светлина или инфрацрвени камери. Исто така, може да се прилагоди алгоритмот за детекција да биде поосетлив на промени во осветлувањето.

2. Лажни Позитиви:

- **Предизвик:** Системот може да генерира лажни позитивни детекции поради движење на објекти кои не се луѓе, како што се животни или сенки.
- **Решение:** Оптимизирајте го алгоритмот за детекција, користете повеќе критериуми за да се намали бројот на лажни позитиви, и применете дополнителни филтри за да ги елиминирате несаканите детекции.

3. Зголемена оптовареност на системот:

- **Предизвик:** Обработката на повеќе видео стримови паралелно може да предизвика оптовареност на серверот и намалување на перформансите.
- **Решение:** Користете сервер со повеќе јадра и поголема меморија за да го распределите оптоварувањето. Дополнително, имплементирајте load balancing ако користите повеќе сервери.

4. Проблеми со поврзување на камери:

- **Предизвик:** Камерите може да изгубат конекција со системот поради проблеми со мрежата.
- **Решение:** Вклучете механизам за автоматско повторно поврзување на камерите при губење на конекцијата. Исто така, обезбедете стабилна мрежна инфраструктура.

5. Доцнење во нотификациите:

- **Предизвик:** Може да се појави доцнење при испраќање на нотификациите, што ќе резултира со задоцнување во известувањето на корисникот.
- **Решение:** Оптимизирајте ја мрежната врска и осигурајте се дека системот за нотификации е правилно конфигуриран за брза испорака.

6. Складирање на големи количини на видео податоци:

- **Предизвик:** Долготрајно складирање на видео податоци може да го оптовари просторот на диск.
- **Решение:** Имплементирајте механизми за автоматско бришење на стари видеа или архивирање на податоците на надворешни уреди за складирање.

10. Идни подобрувања

1. Додавање на е-пошта известувања

- Во иднина, ќе се имплементираат функции за испраќање на е-пошта известувања, заедно со постоечките известувања преку Pushover. Ова ќе овозможи корисниците да добиваат известувања директно на нивната е-пошта, што може да биде корисно за оние кои го преферираат овој канал на комуникација или кога Pushover не е достапен.

2. Сепарирање на системот на Raspberry Pi уреди со индивидуални камери

- Планирано е да се раздели системот на повеќе Raspberry Pi уреди, секој со своја камера. Ова ќе овозможи подобрување на перформансите и намалување на оптоварувањето на еден поединечен уред. Секој Raspberry Pi ќе работи автономно и ќе испраќа податоци до централниот сервер или база на податоци, што ќе ја зголеми стабилноста и флексибилноста на системот.

3. Подобрена интеграција со IoT уреди

- Идните подобрувања ќе вклучат интеграција со други IoT уреди, како што се паметни светла или аларми, кои ќе можат да бидат активирани автоматски кога ќе се детектира непожелна активност. Ова ќе го подобри безбедносниот аспект на системот и ќе овозможи комплетно автоматизирање на безбедносната реакција.

4. Напредно управување со корисници

- Ќе се развијат нови функции за управување со корисници, вклучувајќи можност за дефинирање на различни нивоа на пристап и права за различни корисници. Ова ќе овозможи подобра контрола и управување со системот од страна на администраторите.

11. Заклучок

Системот за безбедност кој е развиен со користење на OpenCV и други напредни технологии нуди робусно и флексибилно решение за детекција на луѓе и управување со видеонадзор. Со три различни имплементации за детекција, можност за конфигурација и управување преку интуитивен кориснички интерфејс, и опција за интеграција со мобилни нотификации, овој систем е идеален за различни безбедносни апликации. Дополнително, системот е модуларен и лесно може да се надгради со нови функционалности, како што се AI-базирани аларми и поддршка за облачни технологии.

Во иднина, системот може да биде дополнително подобрен со воведување на нови аналитички способности, поддршка за повеќе видеоформати, и интеграција со други безбедносни системи. Со тоа, ќе обезбеди уште поголема ефикасност и сигурност во обезбедувањето на различни средини.

Корисниците можат да го користат овој систем за континуирано следење на настаните и брзо реагирање на потенцијални закани, со што значително ќе ја зголемат безбедноста и заштитата на имотот и лицата.