

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ВИКОРИСТАННЯ OPENCV ДЛЯ РЕЄСТРАЦІЇ РУХУ НА ВІДЕОСТРІМІ

МЕТОДИЧНІ ВКАЗІВКИ

**до виконання лабораторної роботи № 3
з дисципліни «Штучний інтелект в ігрових застосунках»
для студентів бакалаврського рівня вищої освіти спеціальності 121
"Інженерія програмного забезпечення"**

Львів -- 2025

Використання OpenCV для реєстрації руху на відеостримах: методичні вказівки до виконання лабораторної роботи №3 з дисципліни "Штучний інтелект в ігрових застосунках" для студентів першого (бакалаврського) рівня вищої освіти спеціальності 121 "Інженерія програмного забезпечення" . Укл.: О.Є. Бауск. -- Львів: Видавництво Національного університету "Львівська політехніка", 2025. -- 10 с.

Укладач: Бауск О.Є., к.т.н., асистент кафедри ПЗ

Відповідальний за випуск: Федасюк Д.В., доктор техн. наук, професор

Рецензенти: Федасюк Д.В., доктор техн. наук, професор

Задорожний І.М., асистент кафедри ПЗ

Тема роботи: Використання OpenCV для реєстрації руху на відеостримах.

Мета роботи: Ознайомитись з основами функціонування системи OpenCV, навчитися використовувати її для реєстрації руху на відеостримах.

Теоретичні відомості

Теоретичні відомості

Що таке OpenCV?

OpenCV (Open Source Computer Vision Library) — це бібліотека з відкритим вихідним кодом, яка містить понад 2500 алгоритмів для комп'ютерного зору та машинного навчання. Вона широко використовується для обробки зображень та відео, розпізнавання об'єктів, відстеження руху, аналізу сцен та інших задач, пов'язаних з комп'ютерним зором.

Відстеження особливих точок (Feature Tracking)

Відстеження особливих точок — це процес визначення та слідування за рухом певних ключових точок (features) на послідовних кадрах відео. Це дозволяє аналізувати рух об'єктів, стабілізувати відео, розпізнавати жести та багато іншого.

Одним з популярних алгоритмів для відстеження особливих точок є алгоритм Лукаса-Канаде (Lucas-Kanade), який реалізований в OpenCV у функції `cv2.calcOpticalFlowPyrLK()`.

Алгоритм Лукаса-Канаде

Алгоритм Лукаса-Канаде базується на припущенні, що рух точок між двома послідовними кадрами є малим і приблизно однаковим у локальному околі точки. Він використовує піраміду зображень для ефективного відстеження руху точок на різних масштабах.

Основні етапи алгоритму Лукаса-Канаде:

- Вибір ключових точок на першому кадрі (наприклад, за допомогою алгоритму Ші-Томасі або Харріса).
- Відстеження цих точок на наступних кадрах за допомогою оптичного потоку.
- Оновлення позицій точок та повторення процесу для наступних кадрів.

Висновок

Бібліотека OpenCV є потужним інструментом для обробки зображень та відео. Вона дозволяє ефективно відстежувати рух об'єктів на відеостримах, що є важливим для багатьох задач, таких як відстеження руху, розпізнавання жестів, аналіз сцен та ін.

Хід роботи

1. Підготовка середовища

1.1. Клонування репозиторію з кодом для лабораторної роботи:

Для початку роботи необхідно клонувати репозиторій з вихідним кодом. Відкрийте термінал (командний рядок) та виконайте наступну команду:

```
git clone git@github.com:bausk/ar-practice-2025.git
cd ar-practice-2025/lab-2025-03-01
```

1.2. Встановлення miniconda і залежностей.

Miniconda - це мінімальний інсталятор для Conda, який включає тільки Conda, Python та кілька інших необхідних пакетів. Це дозволяє створювати ізольовані середовища для різних проектів.

Встановіть miniconda за допомогою [офіційного сайту](https://www.anaconda.com/docs/getting-started/miniconda/install).

<https://www.anaconda.com/docs/getting-started/miniconda/install>

1.3. Створення віртуального середовища та встановлення залежностей

Після встановлення Miniconda та клонування репозиторію, необхідно створити віртуальне середовище та встановити всі необхідні залежності. Переконайтеся, що ви знаходитесь у директорії проекту `lab-2025-03-01` і що нема активованих інших середовищ (командна строка не має показувати `(base)` чи щось подібне).

```
conda env create -n lab-2025-03 -f environment.yml
```

```
conda activate lab-2025-03
```

2. Запуск програми

Після налаштування середовища можна запустити програму для відстеження руху на відеострімі. Програма підтримує різні режими роботи та параметри.

2.1. Запуск з веб-камери

Для запуску програми з використанням веб-камери виконайте наступну команду:

```
python calculate.py
```

2.2. Запуск з відеофайлом

Для запуску програми з використанням відеофайлу (по бажанню і якщо є власне відео в форматі MP4) виконайте наступну команду:

```
python calculate.py --image_path=video.mp4
```

2.3. Додаткові параметри

Програма підтримує додаткові параметри для налаштування:

- `--step` - інтервал між захопленими кадрами в секундах
- `--crop` - відсоток горизонтального поля зору для збереження (за замовчуванням 100%)
- `--maxwidth` - максимальна ширина для зменшення розміру зображення (за замовчуванням без зміни розміру)

Приклад:

```
python calculate.py --step=50 --crop=0.5 --maxwidth=1024
```

3. Взаємодія з програмою

Під час роботи програми ви можете використовувати наступні клавіші для керування:

- `1` - перемикає на щільний оптичний потік з візуалізацією HSV кольорами (за замовчуванням)
- `2` - перемикає на щільний оптичний потік з візуалізацією лініями
- `3` - перемикає на щільний оптичний потік з деформованим зображенням
- `4` - перемикає на метод Лукаса-Канаде
- `s` - збереження поточного зображення
- `f` - горизонтальне відображення зображення
- `пробіл` - пауза/продовження відтворення
- `ESC` - вихід з програми

4. Аналіз результатів

Під час роботи програми ви побачите візуалізацію руху на відеострімі. Залежно від обраного режиму, це може бути:

1. Кольорове зображення, де колір відповідає напрямку руху, а яскравість - швидкості
2. Лінії, що показують напрямок та величину руху
3. Деформоване зображення, що відображає рух
4. Точки та вектори, що відстежуються методом Лукаса-Канаде.

ХІД ВИКОНАННЯ РОБОТИ

1. Запустіть програму за допомогою команди `python calculate.py`.

- 1.1. Виправте помилку у коді. Додайте правильну ініціалізацію `create_default_lk_flow`.

```
create_default_lk_flow = lambda: optical_flow.LucasKanadeOpticalFlow(  
    feature_params=dict(  
        maxCorners=100,  
        qualityLevel=0.2,  
        minDistance=15,
```

```
        blockSize=7,
        useHarrisDetector=False
    ),
    lk_params=dict(
        winSize=(15, 15),
        maxLevel=2,
        criteria=(
            cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT,
            10,
            0.03
        )
    ),
)
```

2. Виберіть режим роботи за допомогою клавіш `1` , `2` , `3` , `4` .
3. Використовуйте клавіші `s` , `f` , `Пробіл` , `ESC` для керування програмою.
4. Поміняйте код ініціалізації `create_default_lk_flow` вище.

Оберіть інші параметри для відстеження руху.

УМОВА ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

1. Встановити систему розгортання моделей глибокого навчання Ollama.
2. Розгорнути локально LLM модель DeepSeek-R1 (Варіант з 1B параметрів).
3. Протестувати локальне розгортання моделі.
4. Дослідити налаштування моделей при локальному розгортанні, зрозуміти різницю між використанням онлайн- сервісів з LLM моделями та власного деплоймента.

ІНДІВІДУАЛЬНІ ВАРІАНТИ ЗАВДАННЯ

Створіти чат з локальною інсталяцією DeepSeek і використати наступні теми для розмови, залежно від номера в списку. -- див. пункт 3.1

ЗМІСТ ЗВІТУ

1. Тема та мета роботи
2. Теоретичні відомості
3. Постановка завдання
4. Хід виконання роботи:
 - Скріншоти процесу створення локальної інсталяції
 - Код та пояснення для створення моделі
 - Скріншоти інтерфейсу
5. Результати роботи
6. Висновки

КОНТРОЛЬНІ ПИТАННЯ

1. Що таке LLM моделі?
2. Що таке попередньо тренувані моделі (pretrained models)?
3. Що таке дистільовані моделі?
4. Що таке локальне розгортання моделі?
5. Який розмір моделі DeepSeek-R1 на вашому комп'ютері?
6. Які функції і задачі має інструмент розгортання моделей Ollama?
7. Яку роль відіграє кількість параметрів у роботі моделі, якості генерації тексту, та швидкості виконання?
8. Які переваги і недоліки локального розгортання моделі?
9. Які переваги і недоліки онлайн-сервісів з LLM моделями, таких як OpenAI/ChatGPT?
10. Які переваги і недоліки дистільованих моделей?

СПИСОК ЛІТЕРАТУРИ

1. [Ollama](#)
2. [DeepSeek](#)
3. [LLM](#)
4. [Distilled models](#)
5. [DeepSeek-R1](#)