

1. Въведение

Този проект цели автоматизирано разпознаване на български регистрационни номера на автомобили чрез комбинация от YOLO за детекция на номера и CRNN за разпознаване на текста (OCR). Системата извлича кадри от камера, детектира номера, разпознава текста и го записва в база данни (Supabase).

2. Архитектура на системата

2.1 Общ workflow

1. **Заснемане на кадри от камерата** – поток от видео кадри се поставя в опашка (Queue).
2. **YOLO модел за детекция** – всеки кадър се анализира за обекти с форма на автомобилен номер.
3. **ROI (Region of Interest)** – частта на изображението, съдържаща номера, се изрязва.
4. **OCR с CRNN** – CRNN моделът разпознава текста от изрязаното изображение.
5. **Проверка на валидност** – регексп проверява дали текстът съответства на формат на български номер.
6. **Запис в база данни** – ако номерът е потвърден (видян поне 2 пъти), се записва в Supabase.

Камера -> YOLO -> ROI -> CRNN OCR -> Проверка -> Supabase

2.2 Компоненти

- **YOLO (You Only Look Once)**
 - Модел за детекция на обекти в реално време.
 - Използва convolutional neural networks (CNN) за feature extraction.
- В проекта: детектира номера на автомобила.
- **CRNN (Convolutional Recurrent Neural Network)**
 - Комбинира CNN за извличане на визуални признаци и RNN (LSTM/GRU) за последователно предсказване на символи.
 - Използва CTC (Connectionist Temporal Classification) loss за обучение, което позволява предсказване на текст с променлива дължина.
 - Особено подходящ за регистрационни номера, защото номерът е линейна последователност от символи, а CRNN разпознава целия низ директно от изображението без ръчно сегментиране.

- **Supabase**

- Онлайн база данни за съхранение на открити и потвърдени регистрационни номера.

- **Python и OpenCV**

- Управление на видео потока и предварителна обработка на изображенията.

3. Подробно описание на кода

3.1 Capture Frames

```
cap = cv2.VideoCapture(CAMERA_SOURCE)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
```

- Настройва камерата и започва заснемане. - Поставя кадрите в `Queue`, за да бъдат обработвани асинхронно.

3.2 YOLO за детекция

```
results = model(frame, conf=0.4, verbose=False)
```

- Всеки кадър се подава на YOLO. - `conf=0.4` означава минимална увереност за маркиране. - Връща bounding boxes за откритите номера.

3.3 OCR с CRNN

```
text = crnn_model.predict(plate_roi)
```

- `plate_roi` е изрязаната област около номера. - CRNN извлича признаци чрез CNN и използва RNN (LSTM) за разпознаване на последователността. - Използва се CTC loss за обучение с променлива дължина на текста. - Връща низ, който се почиства от нежелани символи:

```
clean_text = re.sub(r'^A-Z0-9', '', crnn_text.upper())
```

3.4 Валидиране на номера

```
if BG_PLATE_REGEX.match(clean_text):  
    save_plate_db(clean_text)
```

- Регексп проверява дали низът съответства на български регистрационен номер. - Ако е валиден и е потвърден (видян поне 2 пъти), се записва в базата данни.

4. Обучение на CRNN за български номера

1. Събиране на данни

2. Снимки от cars.bg.

3. Отрязване на номера и съхраняване на изображения (напр. 128x32 px).

4. Всеки файл съдържа ground truth текст.

5. Аугментация на данни

6. Добавяне на шум, промяна на контраста, завъртане и размазване за устойчивост на модела.

7. Обучение

8. Използва се CRNN с CTC loss.

9. Метрики: Character Accuracy и Sequence Accuracy.

10. Съхраняване на теглата

11. След обучение моделът се записва като `crnn_weights.pth` за inference.

5. Предимства на CRNN спрямо EasyOCR

Характеристика	EasyOCR	CRNN
Контрол върху обучението	Не	Да
Поддръжка на специфичен шрифт	Ограничена	Пълен контрол върху шрифта
Скорост	Бърз	Бърз при GPU
Устойчивост на шум	Средна	Висока (може да се аугментира)

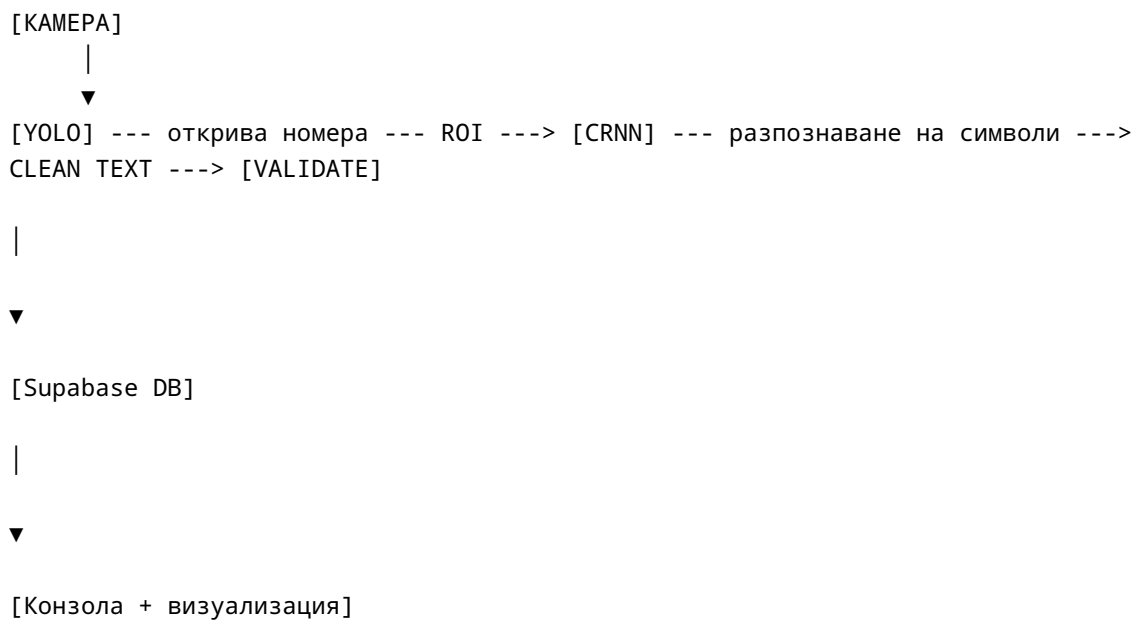
6. Интеграция с текущия проект

1. Заменя EasyOCR с CRNN:

```
crnn_model = load_crnn('crnn_weights.pth')  
text = crnn_model.predict(plate_roi)
```

2. Всички останали стъпки – валидация, съхраняване в Supabase, маркиране в кадъра – остават същите.

7. Визуална схема на workflow с CRNN



- YOLO: открива номера и дава bounding box - CRNN: разпознава текста от изрязаното изображение - Clean & Validate: премахва шум и проверява валиден формат - DB & визуализация: записва и показва резултата