

## QR 코드 스캔

---

- QR 코드 스캔 결과 출력
- QR 코드 스캔 결과 출력 (스캔 위치, 스캔 시간, 스캔 횟수, 스캔 성공 여부, 스캔 실패 사유) 출력
- 8/13 ~ 8/21
- [https://github.com/jaemin6/qr\\_scan](https://github.com/jaemin6/qr_scan)

## 개발 환경

---

- 개발 환경 구성:
  - QR 코드 스캔 라이브러리, QR 코드 생성 라이브러리, QR 코드 스캔 결과 출력 라이브러리
  - QR 코드 스캔 결과 출력 라이브러리
  - QR 코드 스캔 결과 출력 라이브러리
  - QR 코드 스캔 결과 출력 라이브러리
  - QR 코드 스캔 결과 출력 라이브러리
  - QR 코드 스캔 결과 출력 라이브러리
- QR 코드 스캔 결과 출력 라이브러리

## 개발 도구

---

개발 언어: Python

개발 라이브러리/프레임워크: OpenCV, Pillow, Numpy

개발 도구: Git, VS Code

AI 모델: Gemini 2.5 Flash (스캔 결과 출력 라이브러리)

**QR.py:** **QR** **code** **generator** **and** **detector**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url** **(Equalization)**

**Imports:** **cv2.VideoCapture**, **cv2.QRCodeDetector**, **cv2.cvtColor**, **cv2.equalizeHist**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url**

**QR.py:** **QR** **code** **generator** **and** **detector**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url** **(Equalization)**

**Imports:** **OpenCV**, **cv2.polylines**, **cv2.QRCodeDetector**, **cv2.cvtColor** **(cvtColor)**, **cv2.equalizeHist** **(equalizeHist)**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url**

**QR.py:** **QR** **URL** **generator** **and** **detector**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url** **(Equalization)**

**Imports:** **webbrowser**, **time**, **numpy**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url**

- 1: **QR** **code** **generator** **and** **detector**

**Usage:** **python** **QR.py** **generate** **code** **QR** **code** **url**, **python** **QR.py** **detect** **code** **QR** **code** **url**

この章では、QRコードの生成と読み取りの基本的な手順を説明します。QRコードは、2Dバーコードの一種で、デジタルデータを効率的にエンコードするための技術です。

生成と読み取りの手順:

1. 生成: データをバイナリ形式に変換 (Binarization) し、QRコードの形式にエンコードします。この過程で、エラー訂正能力を調整することが可能です。

2. 読み取り: QRコードをスキャンし、データを読み取ります。この過程で、エラー訂正能力を調整することが可能です。読み取りの結果は、元のデータ形式に変換されます。

3. 生成: QRコードの生成 (Black Hat Morphology) は、QRコードの生成に必要です。この過程で、QRコードの生成に必要なパラメータを設定します。QRコードの生成は、QRコードの生成に必要なパラメータを設定します。

4. 読み取り: QRコードの読み取り (Contour) は、QRコードの読み取りに必要なパラメータを設定します。QRコードの読み取りは、QRコードの読み取りに必要なパラメータを設定します。

5. 読み取り: QRコードの読み取り (Gaussian Blur): QRコードの読み取りに必要なパラメータを設定します。QRコードの読み取りは、QRコードの読み取りに必要なパラメータを設定します。

1~5. QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。

この章では、QRコードの生成と読み取りの手順を説明します。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。

この章では、QRコードの生成と読み取りの手順を説明します。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。

この章では、QRコードの生成と読み取りの手順を説明します。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。

この章では、QRコードの生成と読み取りの手順を説明します。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。

この章では、QRコードの生成と読み取りの手順を説明します。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。QRコードの生成と読み取りの手順は、QRコードの生成と読み取りの手順です。

📌 项目目标: 使用 YOLOv11n 模型进行 AI 目标检测并生成 QR 码

📌 所需工具:

1. Roboflow 模型: Roboflow 平台提供预训练的 YOLOv11n 模型, 用于检测目标物体。
2. yolov11n.pt 文件: 从 Roboflow 平台下载 YOLOv11n 模型的权重文件 (yolov11n.pt)。

📌 环境配置: 确保 Python 环境已安装必要的库, 包括 OpenCV (用于图像处理和 QR 码生成)。

📌 运行步骤: 使用 Roboflow 模型对输入图像进行推理, 识别目标物体, 并生成 QR 码。

📌 输出结果: 生成包含目标检测框和 QR 码的图像, 用于展示检测结果。

📌 项目结构:

- 项目目录结构:


1. AI 模型推理: Roboflow 模型推理 yolov11n.pt 文件, 生成目标检测框。
2. OpenCV 图像处理: 使用 OpenCV 库 (VideoCapture) 加载视频帧, 使用 QRCodeDetector 检测 QR 码, 使用 polylines 和 imshow 显示检测结果。
3. Pillow 图像处理: OpenCV 模型输出结果 (BGR 格式) 转换为 RGB 格式, 使用 Pillow 库生成 QR 码。

📌 运行命令: 使用 Python 脚本运行目标检测程序。

📌 项目总结: 本项目展示了如何使用 YOLOv11n 模型进行目标检测, 并结合 OpenCV 和 Pillow 库生成 QR 码。



□□ □□□ □□ □ □ :

1. 
2. 
3. 