

사이버 보안 캡스톤 디자인 최종 발표

Catch Number



1771250 조재연(팀장)



1771405 이연주



지도교수 최원석 교수님

| 목차



프로젝트 소개

주요 기능 및 시스템 구성

구현 결과 소개

프로젝트 수행 후기

| 프로젝트 개요

여성 혹은 그 외 사용자가 늦은 밤이나 혼자서 택시를 타고 가야 하는 경우, 동료나 본인이 번호판을 찍어 놓는 경우가 많다. 이 때, 택시가 너무 멀리 있어 번호판을 제대로 찍을 수 없거나, 본인의 시야에서 번호판이 흐릿하게 보이는 경우를 대비하여 택시의 번호판을 멀리서도 인식 가능하게 하여 번호판 정보를 정확히 알 수 있도록 하고자 한다.



주제 선정 배경



시사저널이 입수한 '택시기사 특정범죄 경력자 통보 현황'에 따르면, 2017년 한 해 동안 범죄 경력자 통보 건수는 매달 평균 62건에 이른다. 월별 통보 건수를 살펴보면 1월 64건, 2월 101건, 3월 99건, 4월 84건, 5월 54건, 6월 80건이었다. 하반기 통보 건수는 7월 75건, 8월 60건, 9월 55건, 10월 44건, 11월 32건으로 각각 나타났다. 11월 한 달간 통보된 택시기사 범죄 유형을 조사한 결과, 살인미수·상습절도·성범죄 등 범죄 전과자가 32명 적발됐다. 살인미수 1명, 상습절도 1명, 강간 치상·준강간 등이 2명이었고, 강제추행과 준강제추행 등 범죄를 저지른 택시기사는 12명이었다. 성폭력은 2명, 아동·청소년 성범죄는 1명이었고, 마약범 9명, 특정범죄가중처벌법 위반(도주·압력)으로 4명이 적발됐다.

조유빈, '전과자 택시' 중 53% 성범죄 저질렀다, <시사저널>, 2017.12.18



서울택시 최근 3년간 주요 유실물 습득 및 반환현황

단위:건



| 구분 | | 합계/수령율 | 2013년 | 2014년 | 2015년 |
|------|------|------------|-------|-------|-------|
| 총 합계 | 습득등록 | 4936 | 1641 | 1702 | 1593 |
| | 반환 | 3065/62.1% | 1038 | 1087 | 940 |
| 핸드폰 | 습득등록 | 2815 | 987 | 1005 | 823 |
| | 반환 | 2058/73.1% | 702 | 758 | 598 |
| 지갑 | 습득등록 | 673 | 247 | 226 | 200 |
| | 반환 | 348/51.7% | 141 | 109 | 98 |
| 가방 | 습득등록 | 394 | 110 | 136 | 148 |
| | 반환 | 241/61.2% | 76 | 85 | 80 |
| 옷 | 습득등록 | 152 | 46 | 53 | 53 |
| | 반환 | 59/38.8% | 18 | 22 | 19 |
| 쇼핑백 | 습득등록 | 64 | 17 | 19 | 28 |
| | 반환 | 28/43.8% | 9 | 9 | 10 |

자료:서울시

그래픽:김지영 디자이너

| 개발 일정

1~3주차

-초기 아이디어 제안 및 최종 아이디어 제안

4~6주차

-openCV 연구 및 예제 실습

-openCV를 활용하여 번호판 인식 알고리즘 연구 및 수행

7~8주차

-tesseract-OCR을 활용한 번호판 속 텍스트 인식 실습 수행

-tesseract-OCR을 활용한 번호판 속 텍스트 인식률을 높이기
고자 JTessBoxEditor를 이용하여 한글 학습

9주차

-중간 발표 및 교수님과의 미팅을 통한 개발 기술 변경

10~13주차

-객체 인식을 위한 새로운 기술로 Tensorflow와 YOLO v3 비교

-객체 인식을 위해 YOLO v3 연구 및 실습

-YOLO mark를 이용하여 수집된 번호판 데이터셋 학습

-YOLO v3를 통해 실시간 및 동영상 속 번호판 인식 수행

-YOLO v3를 기반으로 text detection과 함께 연동하는 것에 어려움이 생겨 openCV 활용으로 변경

14주차

-Android Studio 최종 ui 및 동작 코드 코딩

-최종 발표 준비

| 팀원별 담당 업무



조재연

- openCV를 활용하여 객체 인식 알고리즘 연구 및 개선
- YOLO v3를 활용한 객체 인식 및 학습
- Android Studio 동작 코딩
- Android Studio UI



이연주

- openCV를 활용하여 객체 인식 알고리즘 연구 및 개선
- Android Studio 동작 코딩
- JTessBoxEditor를 활용하여 번호판 속 한글 학습 및 한글 인식을 개선

| 주요 기능 및 시스템 구성

멀리 있는 자동차의 번호판을 인식하여 사용자가 번호판 속 텍스트를 정확히 표시

본인이 위급한 상황이 발생했을 때를 대비하여 긴급 연락망에 바로 발신

분실물 되찾기 위한 URL 이동 버튼



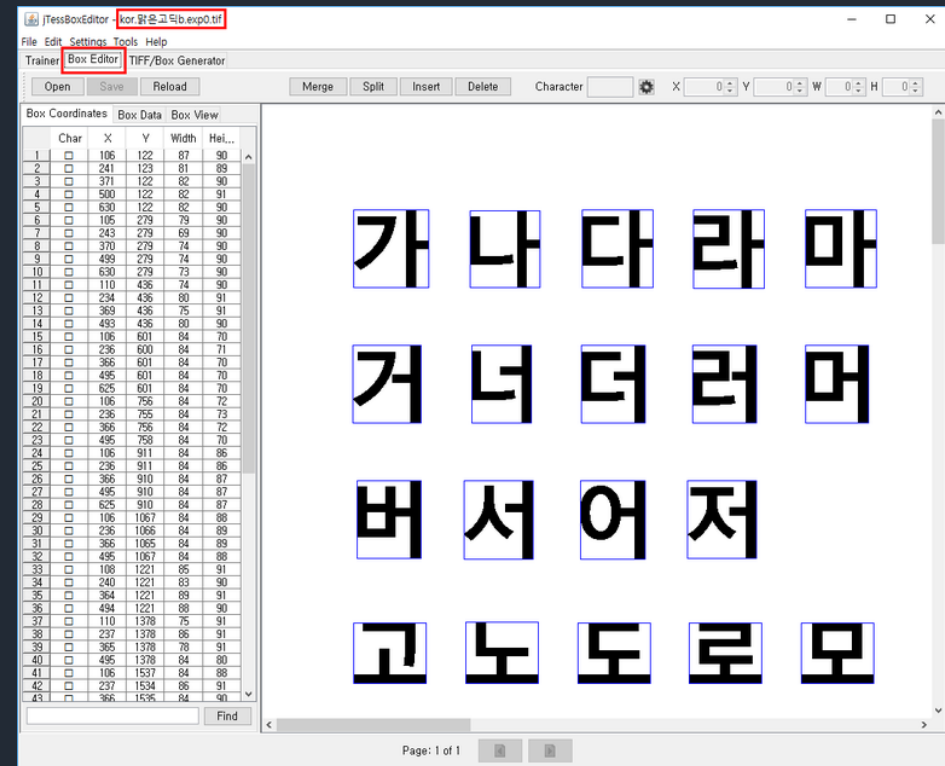
개발 시 요구된 기술 및 소프트웨어/하드웨어 소개

Opecv+tesseract-OCR in Android Studio



Tesseract-OCR 언어 학습

| 분류 | | 기호 |
|--------------|-----|--|
| 비영업용 (관용 포함) | | 가, 나, 다, 라, 마 거, 너, 더, 러, 머, 버, 서, 어, 저 고, 노, 도, 로, 모, 보, 소, 오, 조 구, 누, 두, 루, 무, 부, 수, 우, 주 |
| 영업용 | 일반 | 아, 바, 사, 자 |
| | 택배 | 배 |
| | 렌터카 | 하, 허, 호 |
| 군용 | | 국, 합, 육, 해, 공 |



주요 코드

```
protected void takePicture() {
    if(null == cameraDevice) {
        Log.e(TAG, msg: "cameraDevice is null");
        return;
    }
    CameraManager manager = (CameraManager) getSystemService(Context.CAMERA_SERVICE); //카메라 프리뷰 화면 출력 및 크기 회전 설정
    try {
        CameraCharacteristics characteristics = manager.getCameraCharacteristics(cameraDevice.getId());
        Size[] jpegSizes = null;
        if (characteristics != null) {
            StreamConfigurationMap map = characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURATION_MAP);
            jpegSizes = map.getOutputSizes(ImageFormat.JPEG);
        }
        int width = 640;
        int height = 480;
        if (jpegSizes != null && 0 < jpegSizes.length) {
            width = jpegSizes[0].getWidth();
            height = jpegSizes[0].getHeight();
        }
        ImageReader imageReader = ImageReader.newInstance(width, height, ImageFormat.JPEG, maxImages: 1);
        List<Surface> outputSurfaces = new ArrayList<>(initialCapacity: 2);
        outputSurfaces.add(imageReader.getSurface());
        outputSurfaces.add(new Surface(textureView.getSurfaceTexture()));
        final CaptureRequest.Builder captureBuilder = cameraDevice.createCaptureRequest(CameraDevice.TEMPLATE_STILL_CAPTURE);
        captureBuilder.addTarget(imageReader.getSurface());
        captureBuilder.set(CaptureRequest.CONTROL_MODE, CameraMetadata.CONTROL_MODE_AUTO);
    } catch (CameraAccessException e) {
        Log.e(TAG, "Camera access failed: " + e.getMessage());
    }
}
```

주요 코드

```
int rotation = getWindowManager().getDefaultDisplay().getRotation();
captureBuilder.set(CaptureRequest.JPEG_ORIENTATION, ORIENTATIONS.get(rotation));
//final File file = new File(Environment.getExternalStorageDirectory()+"/pic.jpg");
ImageReader.OnImageAvailableListener readerListener = (reader) -> {
    Image image = null;
    try {
        image = reader.acquireLatestImage(); // 최신 이미지 가져옴
        ByteBuffer buffer = image.getPlanes()[0].getBuffer(); // 이미지의 픽셀 평면 배열
        byte[] bytes = new byte[buffer.capacity()];
        buffer.get(bytes);
        Log.d(TAG, msg: "takePicture");

        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inSampleSize = 8;

        Bitmap bitmap = BitmapFactory.decodeByteArray(bytes, offset: 0, bytes.length); // byte를 bitmap으로 변환
        bitmap = GetRotatedBitmap(bitmap, degrees: 90); // bit map 90도 회전

        Bitmap imgRoi;
        OpenCVLoader.initDebug(); // 초기화

        Mat matBase = new Mat();
        Utils.bitmapToMat(bitmap, matBase); // bitmap을 mat으로 변환
        Mat matGray = new Mat();
        Mat matTopHat = new Mat();
        Mat matBlackHat = new Mat();
        Mat matThresh = new Mat();
```

주요 코드

```
Mat matDilate = new Mat();

Imgproc.cvtColor(matBase, matGray, Imgproc.COLOR_BGR2GRAY); // GrayScale로 변환하고 matgray로 저장
Mat matKernel = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new org.opencv.core.Size( width: 3, height: 3), new Point( x: 0, y: 0)); //사각형 모양으로 변환
Imgproc.morphologyEx(matGray, matTopHat, Imgproc.MORPH_TOPHAT, matKernel, new Point( x: 0, y: 0)); //원본에서 열기연산 제외
Imgproc.morphologyEx(matGray, matBlackHat, Imgproc.MORPH_BLACKHAT, matKernel, new Point( x: 0, y: 0)); //닫기연산에서 원본 제외

Mat matAdd = new Mat();
Core.add(matGray, matTopHat, matAdd); //matgray+mattophat=matadd
Mat matSub = new Mat();
Core.subtract(matAdd, matBlackHat, matSub); //matAdd-matBlackHat=matSub
Mat matBlur = new Mat();
Imgproc.GaussianBlur(matSub, matBlur, new org.opencv.core.Size( width: 5, height: 5), sigmaX: 0); //노이즈 제거
Imgproc.adaptiveThreshold(matBlur, matThresh, maxValue: 255, Imgproc.ADAPTIVE_THRESH_GAUSSIAN_C, Imgproc.THRESH_BINARY_INV, blockSize: 19, C: 9); //영상이진화
Imgproc.dilate(matThresh, matDilate, matKernel); //엣지 테두리 더 굵게 처리

List<MatOfPoint> contours = new ArrayList<>(); //컨투어 배열
Mat hierarchy = new Mat();
Mat matContour = new Mat();
Imgproc.cvtColor(matDilate, matContour, Imgproc.COLOR_GRAY2BGR); //matgrayscale로
Imgproc.findContours(matDilate, contours, hierarchy, Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE); //관심영역 추출
//Imgproc.drawContours(matContour, contours, -1, new Scalar(255, 0, 0), 5);
```

주요 코드

```
new Thread((Runnable) () -> {
    runOnUiThread(() -> {
        imageResult.setImageResource(0); //결과 값 보여주기

        textView.setText("");
    });
}).start();

Imgproc.adaptiveThreshold(matGray, matGray, maxValue: 255, Imgproc.ADAPTIVE_THRESH_MEAN_C, Imgproc.THRESH_BINARY, blockSize: 15, C: 20); //글자 선명화 처리
imgRoi= Bitmap.createBitmap(matGray.cols(), matGray.rows(), Bitmap.Config.ARGB_8888); // 비트맵 생성
Utils.matToBitmap(matGray, imgRoi);
int nContourCount;
//float ratio;

for(int idx = 0; idx >= 0; idx = (int) hierarchy.get( row: 0, idx)[0]) {
    MatOfPoint matOfPoint = contours.get(idx);

    if(Imgproc.contourArea(matOfPoint) < 1000.0)//컨투어 면적이 1000이하면 실행
        continue;

    Rect rect = Imgproc.boundingRect(matOfPoint);//사각형
```

주요 코드

```
if (rect.width < 30 || rect.height < 30 || rect.width <= rect.height || rect.width <= rect.height * 3 || rect.width >= rect.height * 6)
    continue; // 사각형 크기에 따라 출력 여부 결정

Mat matRoi = matThresh.submat(rect);
//모든 번호판 유형 표시
Imgproc.rectangle(matContour, rect.tl(), rect.br(), new Scalar(0, 255, 0), thickness: 2);
nContourCount = getContourCount(matContour, matRoi, rect);
if(nContourCount < 6 || nContourCount > 9) //인식된 글자갯수 체크
    continue;
//최종선택 번호판 유형 표시
Imgproc.rectangle(matContour, rect.tl(), rect.br(), new Scalar(255, 0, 0), thickness: 2);
roi = Bitmap.createBitmap( imgRoi, (int)rect.tl().x, (int)rect.tl().y, rect.width, rect.height);
new Thread((Runnable) () -> {
    runOnUiThread(() -> {
        imageResult.setImageBitmap(roi);
        new AsyncTess().execute(roi);
        btnTakePicture.setEnabled(false);
        btnTakePicture.setText("번호판 인식중...");
    });
}).start();
break;
```

주요 코드

```
protected int getContourCount(Mat matContour, Mat matSubContour, Rect rcComp) {
    List<MatOfPoint> contours = new ArrayList<>(); // contour 배열
    Mat hierarchy = new Mat();
    Imgproc.findContours(matSubContour, contours, hierarchy, Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE); // 상하구조 관계 구성하지 않고 컨투어 라인을
    int nCount = 0;
    float fHeight, fWidth;
    float fCompHeight = rcComp.height;
    for(int idx = 0; idx < contours.size(); idx++) { // 컨투어들을 찾고 사각형으로 표시 후 비율과 가로 세로 비교를 통해 번호판 적합성 구분
        MatOfPoint matOfPoint = contours.get(idx);
        Rect rect = Imgproc.boundingRect(matOfPoint); // 사각형

        fHeight = rect.height;
        fWidth = rect.width;
        if (rect.width > rect.height || fHeight / fWidth < 1.2 || fHeight / fWidth > 3.0 || fCompHeight / fHeight > 2.1 || fCompHeight / fHeight < 1.2)
            continue; // 번호판 내부 글자 유형 비율 체크
        nCount++;

        Imgproc.rectangle(matContour, new Point(x: rcComp.x + rect.x, y: rcComp.y + rect.y),
            new Point(x: rcComp.x + rect.x + rect.width, y: rcComp.y + rect.y + rect.height), new Scalar(0, 0, 255), thickness: 2); // 번호판 내부 글자
    }
    return nCount;
}
```

주요 코드

```
private class AsyncTess extends AsyncTask<Bitmap, Integer, String> {
    @Override
    protected String doInBackground(Bitmap... mRelativeParams) {
        tessBaseAPI.setImage(mRelativeParams[0]);
        return tessBaseAPI.getUTF8Text();
    } //tess로 인식

    protected void onPostExecute(String result) {
        //특수문자 제거
        String match = "[^\\uAC00-\\uD7A3\\xfe0-9a-zA-Z\\s]";
        result = result.replaceAll(match, replacement: " ");
        result = result.replaceAll(regex: " ", replacement: "");
        if(result.length() >= 5 && result.length() < 8) {
            textView.setText(result);
            Toast.makeText(context: MainActivity.this, text: "" + result, Toast.LENGTH_SHORT).show(); //인식 성공시 결과값 표시
        }
        else {
            textView.setText("");
            Toast.makeText(context: MainActivity.this, text: "SUCCESS CAPTURE", Toast.LENGTH_LONG).show(); //인식 실패시 캡처 성공만 표시
        }

        btnTakePicture.setEnabled(true);
        btnTakePicture.setText("번호판 인식");
    }
}
```

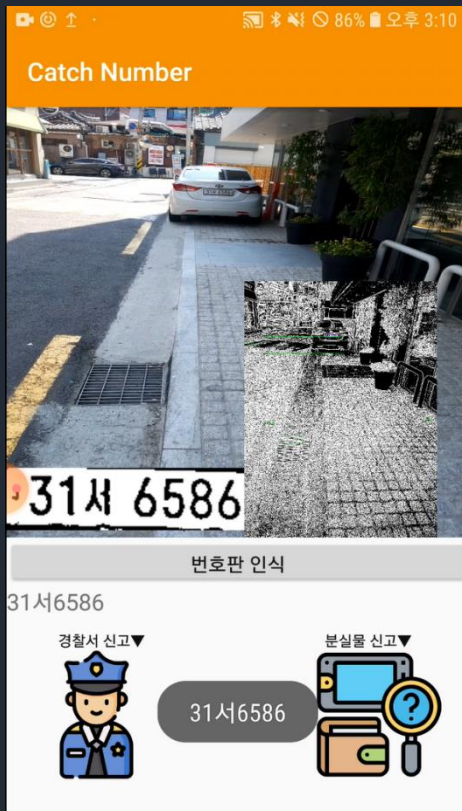

구현 결과 상세 소개



초기 화면

Android Studio를 기반으로 한 어플리케이션이다. 어플리케이션을 실행 할 시 다음과 같이 카메라가 동작 된다. 또한 아래에 번호판 인식버튼, 경찰모양의 이미지 버튼과 분실물 신고 이미지 버튼이 보여짐

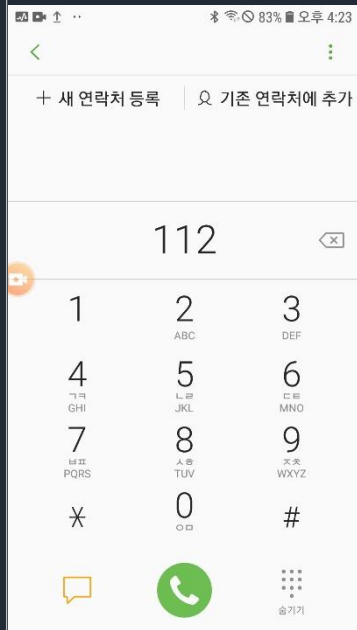
구현 결과 상세 소개



동작 화면

번호판 인식 버튼을 누르게 될 시 카메라가 작동하여 사진을 찍고
번호판 부분을 보여주며 텍스트가 인식되어 표시

구현 결과 상세 소개



이미지 버튼 클릭 화면

경찰 이미지 버튼을 클릭하게 112로 바로 발신이 가능하도록 다이얼로 번호를 표시해주는 화면을 출력
분실물 신고 이미지 버튼을 클릭하면 유실물 센터 - 전국택시운송사업조합연합회의 사이트로 이동

| 참고문헌

Android Studio Camera API 2 부분

- <https://inducesmile.com/android/android-camera2-api-example-tutorial/>

OpenCV를 활용한 번호판 인식 알고리즘 부분

- https://www.youtube.com/watch?v=PpTl7xxGXh4&feature=emb_logo

YOLO v3 활용 부분

- <https://pjreddie.com/darknet/yolo/>

프로젝트 수행 후기

조재연

프로젝트를 진행하면서 처음 접해보는 다양한 알고리즘 및 소프트웨어들을 다루기가 쉽지 않았습니다. 처음에는 접근조차 쉽지 않았던 다양한 소프트웨어들을 이번 프로젝트를 통해서 하나씩 연구하고 실습하며 진행함으로써 다양한 지식들을 얻은 것 같습니다. 한 가지 아쉬웠던 점이 있다면 YOLO v3를 이용하여 객체 인식을 하고 싶었으나 이번 프로젝트에서는 마무리 단계에서 Text detection과 연동에 어려움을 겪어 활용하지 못한 점이 아쉬웠던 것 같습니다.

프로젝트 수행 후기

이연주

프로젝트를 진행하면서 확실히 openCV보다는 yolo를 활용한 OCR 시도가 훨씬 많고, 옴로를 많이 지원하고 있음을 많이 느꼈다. 한글 텍스트 인식률만 높아진다면 개별 프로젝트로 주차장 관리 시스템 같은 서비스 어플을 한 번 개발해 보고 싶다. 데이터 베이스와 연동하여 번호판이 인식되면 출입하고 나갈 때의 상태에 대한 데이터를 ec2에 저장하여 주차장 요금이 얼마 예상 되는지에 대한 프로젝트를 구현해 내는 것도 우리가 이번 프로젝트를 진행하면서 더 응용할 수 있는 굉장히 유의미한 결과물이 될 것 같다. 한글 인식률을 높이기 위한 여러 시도를 해보았지만 큰 차이를 내지 못한 것 같은 부분이 굉장히 아쉬웠다.