**정보기술연구소**

**운전면허 자동인식 서비스 분석**

**Version 1.0**

*Issue Date: 2018/10/29*

*KCC정보통신*

*인턴 김 지 윤*

목차

* 1. 문서 [개요3](#_개요)
  2. [Ocr로 운전면허증 읽기3](#_OCR로_운전면허증_읽기)
  3. [ocr이란? 3](#_OCR이란?)
  4. [aws](#_AWS) 4
  5. [ms azure](#_MS_Azure) 5
  6. [진위여부 체크](#_진위여부_체크)7
  7. [개요](#_개요_1) 7
  8. [**운전면허증 자동검증시스템 API 분석**](#_운전면허증_자동검증시스템_API) **7**
  9. [Hybrid app 개념 10](#_Hybrid_app_개념)

# 문서 개요

본 문서는 OCR(Optical Character Recognition) 기능으로 운전면허증을 읽어들여 진위여부를 파악하는 서비스 개발을 위한 사전조사 단계의 산출물이다. OCR은 광학 문자 인식 기능으로, 다양한 기업에서 API로 기술을 제공한다.

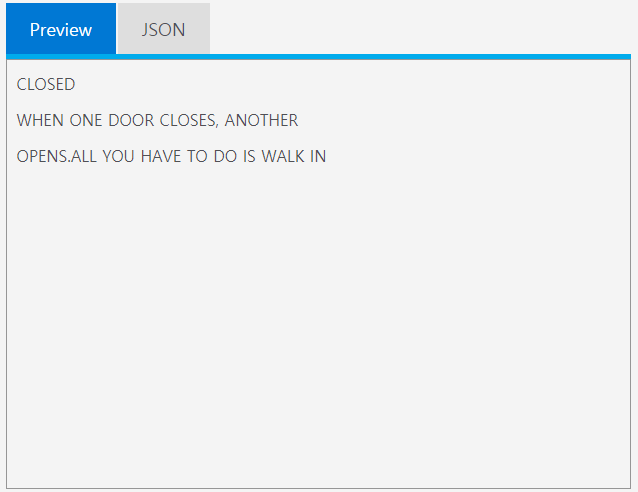
본 문서에서는 AWS(Amazon Web Service)에서 제공하는 Amazon Rekognition과, Microsoft Azure에서 제공하는 Computer Vision API를 주로 비교 및 분석한다.

또한, 본 문서에서는 OCR을 사용하여 읽어 들인 데이터를 도로교통공사로 넘겨 해당 운전면허증의 진위여부를 응답받는 과정까지 분석하여 개발 일정을 잡고 진행해 나가는데 차질이 없도록 작성하고자 한다.

# OCR로 운전면허증 읽기

## OCR이란?

OCR이란 Optical Character Recognition의 약어로, 광학 문자 인식을 뜻한다. 이미지에 나타난 글씨를 인지해 텍스트 데이터(기계로 판독 가능한 문자 스트림)로 치환하는 기능으로, 다양한 기업에서 인공지능(A.I) Vision API로 기술을 제공한다. (아래 사진 참고 Microsoft Azure : <https://azure.microsoft.com/ko-kr/services/cognitive-services/computer-vision/#text>)

위와 같은 기능이 제공되며, 텍스트를 읽어내는 정확도는 기업마다 각기 다르다. Google에서는 OCR 기술을 open source로 제공하는데, 복합적인 이미지 내 문자의 경우 정확도가 60% 정도이고, MS에서 제공되는 open API의 경우 85% 정도로 타 기업에 비하여 높으며 한글의 경우에도 성능이 꽤 좋은편이다. AWS와 ABBYY 또한 API로 제공하는데, ABBYY는 기기(ex. 모바일)의 카메라 Preview(미리보기) 화면에서 이미지를 스캔하여 별도의 저장 없이 텍스트를 실시간으로 인식할 수 있는 기술을 제공한다.

OCR API 참조 : <http://hoondongkim.blogspot.com/2017/02/ms-cognitive-service-ocr.html>

## AWS

AWS에서는 Amazon Rekognition으로 OCR open API를 제공한다. Amazon Rekognition은 이미지와 비디오 분석을 애플리케이션에 쉽게 추가할 수 있도록 도와주는데, Rekognition API에 이미지나 비디오를 제공하면 서비스에서 객체, 텍스트, 사람, 장면, 활동들을 파악할 수 있다.

Amazon Rekognition의 장점으로는, 매일 수십억 개의 이미지와 비디오를 매일 분석 할 목적으로 개발하여 성능이 검증되었고, 확장성이 뛰어난 딥 러닝 기술을 기반으로 한다는 점이다.

본 서비스에서 필요한 기능은 ‘Detecting Text in an Image’ 기능이며, 대략적인 동작 과정은 아래와 같다(참고 : <https://docs.aws.amazon.com/ko_kr/rekognition/latest/dg/text-detecting-text-procedure.html> ).

- AmazonRekognitionFullAccess 권한과 AmazonS3ReadOnlyAccess 권한을 가진 IAM 사용자를 만들어 업데이트

- AWS CLI 와 AWS SDK를 설치 및 구성

- input image를 byte 배열(base 64 인코딩 이미지 바이트) 또는 Amazon S3 객체로 제공(이미지 S3버킷에 업로드)

\* Amazon S3(Amazon Simple Stroage Service) : 인터넷용 스토리지 서비스로, Amazon S3에서 제공하는 단순한 웹 서비스 인터페이스를 사용하여 웹에서 언제 어디서나 원하는 양의 데이터를 저장하고 검색 가능

참고 : <https://docs.aws.amazon.com/ko_kr/AmazonS3/latest/user-guide/create-configure-bucket.html>

- input image로 DetectText 작업 요청(OCR 부분)

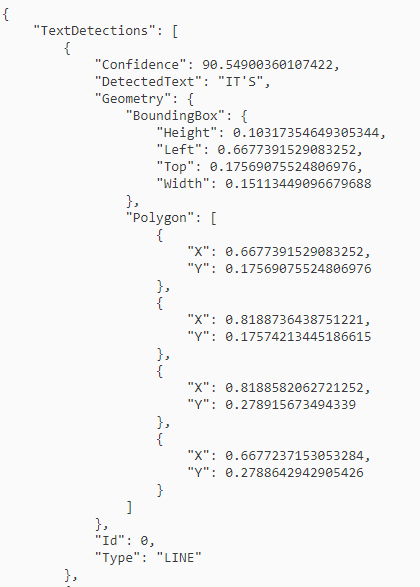
- DetectText 작업은 이미지 분석 후 TextDetections 배열 반환, 이때 각 요소(TextDetection)는 이미지에서 감지 된 LINE이나 WORD를 나타냄

- TextDetection마다 DetectText는 다음 정보를 반환



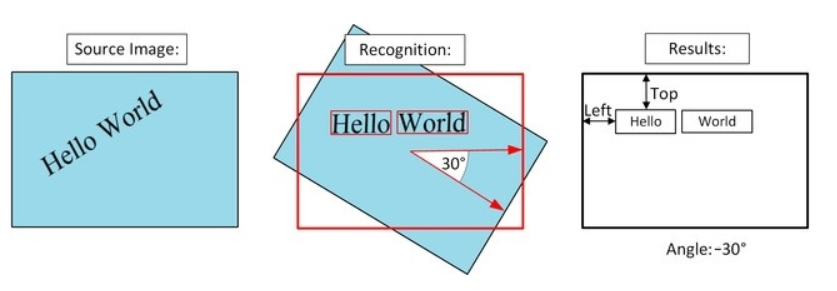
* 위 그림에 대한 예시(객체 형태 아래 이미지 참고)

|  |  |  |
| --- | --- | --- |
| 정보 | 내용 | 예시 |
| DetectedText | 감지 된 텍스트 | ‘but keep’ |
| Id/ParentId | Word와 Line의 관계  - WORD는 자신이 포함된 LINE의 id 값을 parentId로 가짐 | ‘but keep’의 id=3  ‘but’의 id=8, parentId=3  ‘keep’의 id=9, parentId=3 |
| Geometry | 이미지에서 텍스트 위치 확인 | - BoundingBox(텍스트를 둘러싼 직사각형 객체)  - Polygon(각 꼭지점의 x축, y축) |
| Confidence | 감지된 텍스트와 경계 상자의 정확성 | 0~100 사이의 실수 |
| Type | 감지된 텍스트의 유형 | WORD/LINE |



## MS Azure

Microsoft Azure는 Computer Vision API로 OCR API를 제공한다. Computer Vision의 OCR 기술은 이미지에서 텍스트 콘텐츠를 감지하고 식별된 텍스트를 기계로 판독 가능한 문자 스트림으로 추출한다. 또한, 25개 언어를 지원하며, 필요한 경우 가로 이미지 축을 중심으로 인식된 텍스트의 회전도를 수정한다(아래 사진 참조 : <https://docs.microsoft.com/ko-kr/azure/cognitive-services/Computer-vision/concept-extracting-text-ocr> ).



본 서비스에서 필요한 기능은 ‘OCR을 사용하여 인쇄된 텍스트 추출’ 기능이며, 대략적인 흐름은 아래와 같다(node.js 기준으로 분석, 참고 : <https://docs.microsoft.com/ko-kr/azure/cognitive-services/Computer-vision/quickstarts/node-print-text> ).

- Computer Vision에 대한 구독 키 필요

- input image는 JPEG, PNG 또는 BMP 형식으로 제공되어야 함

- npm request 패키지 설치

- 제공하는 코드를 .js 확장명의 파일로 저장 후 실행

- 성공적인 응답은 JSON을 통해 반환, 반환 된 객체는 아래와 같은 속성들을 가짐(객체 형태 아래 이미지 참고)

: language

: orientation

: textAngle

: regions <- boundingBox, lines 객체(boundingBox, 여러 개의 words객체)



# 진위여부 체크

## 개요

OCR로 가져 온 운전면허증 데이터를 도로교통공단에서 제공하는 운전면허증 자동검증시스템 API를 이용해 진위여부를 가리는 과정을 분석한다.

OCR로 가져 온 운전면허증 데이터에는 ‘**운전면허번호(12), 운전면허자명, 생년월일, 암호일련번호, 면허종별(코드), 대여기간시작, 대여기간종료**’ 7가지 이다.

## 운전면허증 자동검증시스템 API 분석

* 통신 방법

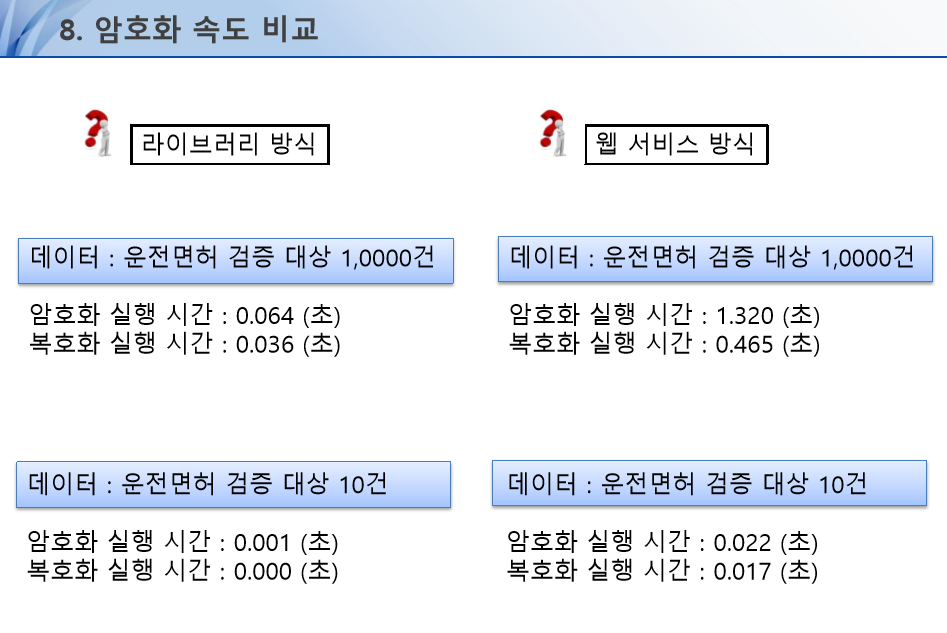
|  |  |
| --- | --- |
| HTTP Protocol | 개발 서버 : HTTP |
| HTTP Method | POST |
| 데이터 포맷 | JSON |

* 보안 방법

\* Aria Block 암호와 알고리즘 사용하여 암호화, 데이터는 Base64 인코딩

Base64 : 8비트 이진 데이터를 문자코드에 영향 받지 않는 공통 ASCII 영역의 문자들로만 이루어진 일련의 문자열로 바꾸는 인코딩 방식

\* 암호와 알고리즘은 JAVA의 경우 라이브러리, JAVA가 아닌 경우 웹 서비스로 제공(암호화 속도 비교 참고 사진 : 운전면허자동검증시스템 암호화 알고리즘 제공 방안)



* 사용자 인증

\* OAuth2 인증방식에 따라 제공되는 Client\_ID, Client\_Secret으로 인증 토큰 발급받아 Token 총해 API 호출

\* 인증 URL

|  |  |  |
| --- | --- | --- |
| 테스트 서버 | POST http://개발서버/oauth2/token | 토큰 발급 API(return URL을 통해 발급, Base64 인코딩 된 값) |
| 운영 서버 | POST http://운영서버/oauth2/token |

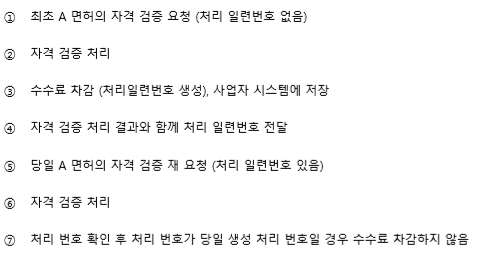
\* 인증 키(Token)은 생성 후 3시간 유효

* 필요한 API

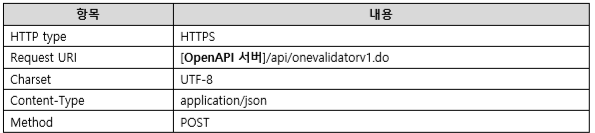
|  |  |
| --- | --- |
| API 명 | 면허정보 단건조회(v1) |
| METHOD | POST |
| 설명 | 한 명의 운전자에 대한 면허 정보를 검증.  단, 당일 동안 해당 면허자에 한해 1회만 수수료 과금 |

* 사용 예제

\* 처리 프로세스



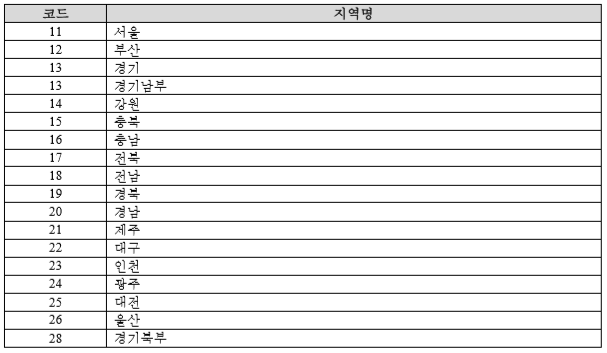
\* 프로토콜 상세



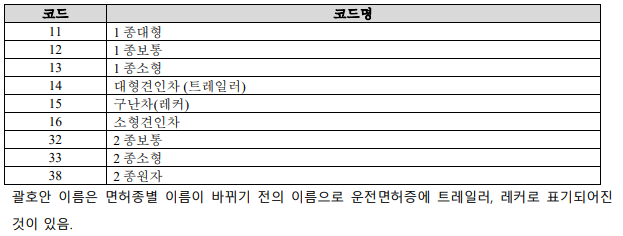
\* 요청 변수(Request Parameters)



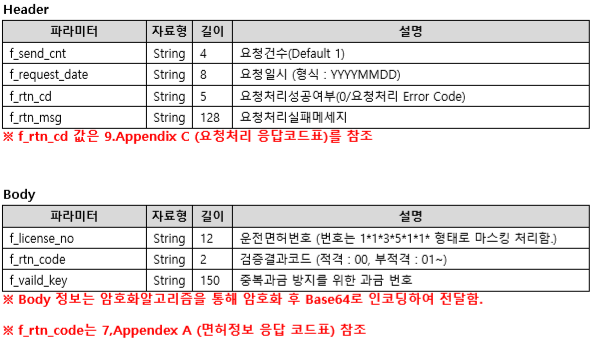
+ 운전면허번호(f\_license\_no)는 총 12자리, 번호 앞에 지역이름이 쓰인 경우는 지역 명 숫자로 치환하여 전달 -> 아래 Appendix D – 지역 코드표 – 참고



+ 면허종별(코드)는 아래 Appendix B – 면허 종별 코드표 – 참고



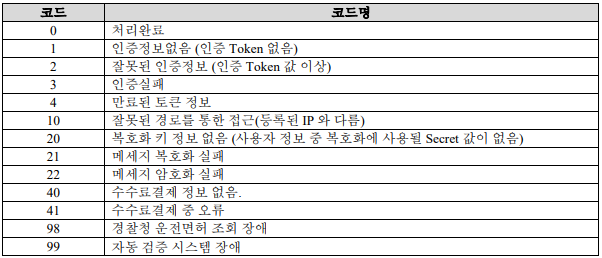
\* 응답 변수(Response Parameters)



+ Appendix A – 면허정보 응답 코드표 –



+ Appendix C – 요청처리 응답코드표 –



출처 : <file:///C:/Users/Jiyoon/Downloads/운전먼허정보자동검증_HTTP_연동가이드_V1.2.pdf>

# Hybrid app 개념

* 개념

기본적으로 ‘네이티브앱 + 웹앱’ 개념이며, 일반적으론 HTML 기반의 웹 앱을 개발한 후, 오픈 소스 크로스 프레임워크를 이용해 네이티브 앱으로 변환한 후 배포하는 형식을 말한다(최종 앱 배포에 필요한 패키징 처리만 아이폰, 안드로이드 플랫폼 안에서 처리한 어플리케이션)

.

(사진 출처 : <https://m.blog.naver.com/lool2389/220908482525> )

- 장점

네이티브 API와 브라우저 API를 이용한 다양한 개발 가능

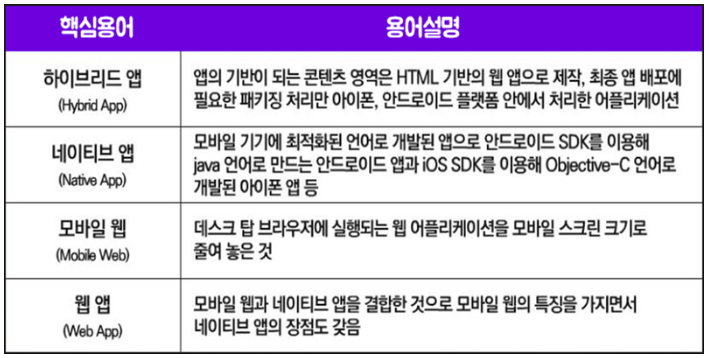
웹 개발 기술을 사용해 앱 개발 가능

한 번의 개발로 다수의 플랫폼에 대응 가능

- 단점

웹 뷰에서 앱을 실행하는 경우이기 때문에, 앱의 성능이 곧 브라우저의 성능

UI 프레임워크 도구를 사용하지 않는다면 개발자가 UI를 제작해야 함



출처 : <https://m.blog.naver.com/PostView.nhn?blogId=acornedu&logNo=221012420292&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>

* Framework

네이티브 기능들을 호출할 때 하이브리드 프레임워크(Cordova 폰갭 등) 사용하면 js함수 한 두줄로 네이티브의 카메라 기능 호출 등이 가능해진다.

**- Apache Cordova -**

\* HTML, CSS 및 JS가 포함 된 모바일 앱 그리고 단일 코드 기반의 멀티플 플랫폼을 타켓으로 한 무료이자 오픈소스 프레임워크

\* 15가지 툴 존재 -> 어도비 폰갭(Adobe PhoneGap)이 가장 유명

\* 간단한 데스크탑 및 개발자 앱을 사용해 HTML, CSS 및 자바스크립트(JavaScript)를 몇 분 안에 기기의 앱으로 전환 가능

\* 크로스 플랫폼 워크플로(Android, iOS, Window Phone 등 8개 플랫폼 지원) -> 통합 자바 스크립트 API로 Wrapping(앱 포장) 기능 제공

\* 플러그인 -> 웹 뷰(Web View)기반이라 네이티브 앱 보다 다양한 기능을 구현하기 어려운 하이브리드 앱의 단점을 극복하기 위해 제공(직접 개발도 가능) : One source Multi device!

\* Adobe PhoneGap : 대표적인 모바일 개발 프레임워크, HTML5, CSS, JavaScript를 활용해 모바일 기기를 위한 어플리케이션 제작 가능

\* Visual studio(개발 툴) : Android, iOS 및 Windows 용 크로스 플랫폼 앱 제작에 인기가 많고, 고급 빌드 빛 디버깅 지원으로 완성됨

\* lonic : lonic 프레임워크는 앵귤러(Angular JS)와 조화를 이루며 작동, 다양한 유틸리티 지원, 각 OS에 따른 개별적 수정 가능, UI 컴포넌트를 제공

참고 : <https://m.blog.naver.com/PostView.nhn?blogId=acornedu&logNo=220981674315&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>