

A Study on the Implementation of Web Programming Using Optical Character Recognition for Traceability Management of Aviation Industry Products

Dong-Won Park*, Myung-Seob Yoon**, Seong-Hyun Park***

*M.S. Student, Dept. Information Technology Convergence,
Graduate School of Technology & Convergence, Kongju National University, Chungnam, Korea
**Professor, Dept. of Aviation maintenance, University of Gyeongnam Geochang, Korea
***Ph. Student, Dept. of Computer Engineering, Kongju National University, Chungnam, Korea

[Abstract]

This paper describes the design and implementation of web programming using optical character recognition (OCR) technology in inspecting and tracking Certificate of Conformance (COC) issued by various raw material manufacturers and distributors in the aviation industry market. According to the international standard AS9100 for aerospace and defense quality control systems, ensuring traceability of raw materials is a very important part of product quality maintenance and management. However, practical traceability tests depend on human experience, and they are implemented to be tested through web programming using Java language to increase efficiency and reliability. There is room for improvement in the cost and operability of OCR technology through further research in the future. In addition, web-based programming has room to supplement and improve security aspects because data transmission and reception are cloud-based. Through this programming, users expect to be effective in reducing the direct cost of testing by reducing errors that can occur through human factors and reducing the inspection time.

▶ **Key words:** OCR(Optical Character Recognition), COC(Certificate of Conformance),
Google Cloud Vision, JavaScript, AS9100

-
- First Author: Dong-Won Park, Corresponding Author: Seong-Hyun Park
 - *Dong-Won Park (pdwastro1113@gmail.com), Dept. Information Technology Convergence, Graduate School of Technology & Convergence, Kongju National University
 - **Myung-Seob Yoon (yms1279@naver.com), Dept. of Aviation maintenance, University of Gyeongnam Geochang
 - ***Seong-Hyun Park (a94270816@gmail.com), Dept. of Computer Engineering, Kongju National University
 - Received: 2024. 11. 04, Revised: 2024. 12. 06, Accepted: 2024. 12. 11.

[요 약]

본 논문은 항공 산업 시장에서 다양한 원소재 제조업체 및 유통업체에서 발행한 Certificate of Conformance(COC)를 검사하고 추적하는데 있어서 광학 문자 인식(OCR)기술을 활용한 웹 프로그래밍 설계 및 구현에 대해 기술한다. 항공우주 및 방산 산업의 품질 관리 시스템에 대한 국제 표준 AS9100에 따르면 원소재의 추적성을 확보하는 것은 제품의 품질유지 및 관리에 있어 매우 중요한 부분이다. 하지만 실질적인 추적성 검사는 사람의 경험도에 의존하고 있으며 보다 효율적이고 신뢰성을 높이기 위하여 Java언어를 이용한 웹 프로그래밍을 통해 검사를 할 수 있도록 구현하였다. 추후 추가적인 연구를 통하여 OCR 기술의 비용적인 측면 및 운용성에 대해서 개선할 여지가 있다. 또한 웹 기반의 프로그래밍은 데이터의 송수신이 클라우드 기반으로 이루어지기 때문에 보안적인 측면을 보완하여 개선할 여지가 있다. 이 프로그래밍을 통해 사용자는 인적요소를 통해 발생 할 수 있는 오류를 줄이고 검사시간을 단축시킴으로 검사하는데 드는 직접비용을 줄이는 것에 효과적일 것을 기대한다.

▶주제어: 광학 문자 인식, 원소재 성적서, 구글 클라우드 비전, 자바스크립트, 에이에스9100

I. Introduction

1. Background of Study

현재 대한민국의 항공기 제조업체 및 항공 부품품 제조업체에서 사용되는 원소재는 대부분 국산품이 아닌 외국 수입품의 의존도가 현저히 높다.

다음 표 1은 2022년 항공산업조사분석사업에 따른 국내 항공제조업체 수급 규모 및 동향을 나타낸 것이다.

Table 1. Size of supply and demand¹⁾

(Unit: 1Million \$)											
Catagory	2018		2019		2020		2021		2022		
	Result	Persent (%)	Result	Persent (%)	Result	Persent (%)	Result	Persent (%)	Result	Persent (%)	
Supply	Production	4,714	18.8	6,028	27.9	4,903	-18.7	5,035	2.7	6,351	26.1
	Import	4,635	30.5	5,281	13.9	3,555	-32.7	3,522	-0.9	4,499	27.7
계		9,349	24.3	11,309	21.0	8,458	-25.2	8,557	1.2	10,850	26.8
Demand	Domestic Demand	6,732	23.8	8,487	26.1	6,826	-19.6	6,845	0.3	8,435	23.2
	Export	2,617	25.5	2,822	7.8	1,632	-42.2	1,712	4.9	2,415	41.1

2022년 기준 항공제조업체의 수입 전망 증감률은 보다 커질 것으로 예상되며 약 40%정도의 공급이 수입품에 의존하고 있는 것을 알 수 있다. 항공기에 사용되는 부품 및 부품품에 대한 규제와 인증은 국제적으로 감항 당국의 엄격한 기준을 충족해야 한다. 이러한 인증을 받기 위한 기술적, 절차적 요구사항이 복잡하고 시간이 오래 걸리기 때문에, 이미 인증된 외국 부품을 수입하여 사용하는 것이 현실적일 수 있기 때문이다. 따라서 많은 현대 항공기

제조업체 및 항공 부품품 제조업체들은 글로벌 공급망에 의존하고 있다. 수없이 많은 부품과 원소재가 국제적으로 유통되며, 전 세계적으로 협력되는 공급망에서 부품 및 원소재를 조달하는 실정이다. 이러한 과정 중에서 항공 산업체는 AS9100에 따라 부품 및 원소재를 운용하고 관리해야 한다. AS9100이란 항공우주 및 방산 산업의 품질 관리 시스템에 대한 국제 표준으로, 이 표준은 품질 관리와 관련된 다양한 요구사항을 설정하고 있다. 그 중에서도 원소재 추적성은 AS9100의 중요한 부분 중 하나로, 항공우주 및 방산 제품의 품질과 신뢰성을 보장하기 위해 매우 중요한 역할을 한다. 항공우주 산업에서 품질과 안전성을 보장하기 위해 만들어진 국제 항공 품질경영시스템 표준 AS9100의 8.5.2항 에 따르면 추적성을 위한 요구사항이 존재한다. 특정 원소재 및 부품에 결함이 있고 항공기 재해를 유발 또는 잠재적으로 유발할 수 있는 사항이 발견되면 다른 항공기의 어떤 부품들이 영향을 받을 수 있는지 알아낼 수 있어야 하기 때문이다. 유통된 결함이 있는 원소재가 어떠한 비행기 추락 사고의 원인이 된 경우 동일한 원소재의 배치(Batch)로 다른 어떤 제품을 만들었는지 정확하게 알지 못하면 큰 문제가 발생할 수 있다. 동일한 원소재 및 부품을 사용한 다른 비행기들이 결함을 가지고 있는 지 아닌지 확실히 알 수 없기 때문에 의심스러운 모든 제품을 교체하게 된다. 이는 곧 실패비

1) 한재현, 2022 항공산업조사분석사업, 한국교통연구원, 2022, p.120

용의 증가로 이어지며 제조업체의 막대한 손실을 가져올 수 있다. 따라서 이러한 추적성은 각 개별 부품에 사용되는 원소재까지 확장되어야 한다. 이토록 항공기 제조업체 및 항공 부품품 제조업체에서 원소재의 추적성을 유지하고 관리하는 것은 매우 중요한 부분이라고 할 수 있다.

2. Research Objective

실제로 원소재를 수입하여 검사하고 그 추적성을 검사하기 위한 과정은 검사원의 눈과 손으로 직접 이루어진다. 단순히 서류만 검사하는 작업을 한다면 어느 정도 잠재적인 실수를 줄일 수 있겠지만 많은 작업이나 긴 근무시간으로 인해 작업자가 스트레스를 받거나 피로해짐으로 일부 서류를 누락 하거나 필요한 추적성 정보를 검사하는 데 시간이 오래 걸리는 문제가 발생한다. 또한 검사원의 경험도와 업무의 효율이 비례하는 특징이 있다. 충분한 교육이나 훈련받지 않은 작업자가 방대한 양의 서류와 복잡한 정보들을 검사할 때 문제가 발생할 수 있다. 따라서 검사 시 이러한 인적요인으로 발생하는 문제들을 방지하고자 웹 프로그래밍과 OCR기술을 활용하여 자동 검사 시스템을 설계 및 구현하고자 한다. OCR과 클라우드 비전에 대하여 2장 관련연구에서 기술하고, 설계, 구현, 실행 검증에 대하여 3장 본문에서 기술하며, 분석 및 향후 연구에 대해서는 4장 결론에서 기술한다.

II. Preliminaries

1. Optical Character Recognition(OCR)

Optical Character Recognition(OCR)이란 사전적 의미로는 텍스트 이미지를 기계가 인식하여 읽을 수 있는 디지털 텍스트 포맷으로 변환하는 기술이다. 크게는 문자 탐지(Text Detection)와 문자 인식(Text Recognition)의 과정이 주를 이룬다. OCR 기술은 카메라를 통해 자동차 번호를 인식한다거나, 신분증에서 개인 정보 텍스트를 인식하는 등 다양한 분야에 사용되며 최근에는 속도를 개선하거나 프로세스를 조정하는 등 다양한 형태로 발전하고 있다. 또한 OCR 기술은 다양한 업체 서류의 이미지 및 문서 처리 업무 등을 자동화하여 프로세스 효율화 및 비용 절감에 큰 효과가 있다고 평가받고 있다. 관련 사례 연구에 따르면 OCR 기술을 활용하여 MSDS의 주요 정보

인 CAS 번호, 제품이름, 위험 번호, 예방 번호, 각 물질의 함량, 그리고 물질명을 효과적으로추출할 수 있었다. 이를 통해, MSDS에서 중요한 정보를 효과적으로 추출하고 이를 활용하여 화학물질의 안전 관리를 강화할 수 있을 것으로 기대된다.²⁾ 라고 이야기한 바 있다. 본 논문에서는 광학 문자 인식(OCR) 기술을 이용하여 실제 문서의 이미지를 기계가 인식하여 읽을 수 있는 디지털 포맷으로 변환하여 필요한 추적성 정보를 연결 짓고 검사하는 데 활용할 것이다.

2. Google Cloud Vision

본 논문에서는 구글 클라우드 비전의 OCR기술을 활용하여 프로그램을 설계 및 구현하였다. 구글 클라우드 비전의 아키텍처는 머신러닝과 딥러닝 모델을 기반으로 하고 있다. 구글은 텐서플로(TensorFlow)와 같은 자사 머신러닝 프레임워크를 활용하여 다양한 모델을 훈련 시켰으며, 이러한 모델은 구글의 클라우드 인프라를 통해 제공된다. 사용자는 REST API 또는 클라이언트 라이브러리를 통해 구글 클라우드 비전 서비스에 접근할 수 있으며, 이를 통해 이미지를 전송하고 분석 결과를 받을 수 있다. 무엇보다 구글 클라우드 인프라를 기반으로 하여 높은 확장성과 접근성이 우수하며 속도 측면에서 보았을 때 타사의 OCR 처리 기술보다 월등히 빠르다는 평이 있다. 그러나 무료로 사용할 수 있는 오픈소스가 아닌 점과 클라우드 시스템을 기반으로 하여 데이터를 전송하기 때문에 민감한 데이터의 경우 보안 이슈가 발생 있을 수 있다. 사용자는 이러한 구글 클라우드 비전의 장단점을 충분히 고려 한 후, 적절한 사용 전략을 수립해야 할 것이다.

3. OCR Work Flow

구글 클라우드 비전의 OCR 기술은 이미지나 문서 내의 텍스트를 자동으로 감지하고 추출하는 과정으로, 이미지에서 읽을 수 있는 텍스트 데이터를 구조화된 형태로 변환한다. 이 기술의 워크플로우는 여러 단계로 구성되며, 각 단계는 이미지의 전처리부터 텍스트 추출 및 후처리에 이르는 전체 프로세스를 포함한다. 워크플로우는 아래 그림 1과 같다.

2) Kim Kyu Hyeong, Kang Seong Hwan, & Cho We Duke (2023-02-08). Development of OCR-based algorithm for information extraction from Material Safety Data Sheets (MSDS). Proceedings of Symposium of the Korean Institute of communications and Information Sciences, Gangwon State.



Fig. 1. Work Flow of OCR Process

3.1 Uploading Image

사용자는 OCR을 수행할 이미지를 준비하고, 구글 클라우드 비전 API가 접근할 수 있도록 클라우드 스토리지(Google Cloud Storage)나 직접적인 파일 업로드를 통해 이미지를 전송한다. 이미지 포맷은 JPEG, PNG, GIF 등 다양한 형식을 지원한다.

3.2 Pre-processing Image

이미지가 업로드되면, 구글 클라우드 비전 서비스는 이미지의 품질과 가독성을 향상시키기 위해 전처리를 수행한다. 이 단계에서 이미지의 노이즈 제거, 대비 조정, 스케일링 등이 포함될 수 있다. 이러한 전처리 작업은 OCR의 정확도를 높이기 위해 필수적이다.

3.3 Text Detection

구글 클라우드 비전의 OCR 엔진은 이미지 내 텍스트가 포함된 영역을 감지한다. 이 단계에서는 텍스트가 포함된 박스를 정의하고, 페이지 레이아웃 분석을 통해 텍스트 블록의 위치, 크기, 방향을 결정한다.

여기서 사용되는 기술로는 머신러닝과 딥러닝 기반의 CNN(Convolutional Neural Networks)이 있으며, 이 기술은 텍스트의 특징을 학습하고 검출한다.

3.4 Text Recognition

감지된 텍스트 영역에서 실제 텍스트를 추출한다. 구글 클라우드 비전의 OCR 기술은 이미지의 각 텍스트 블록, 단어, 문자 수준까지 세분화하여 텍스트를 읽어낸다. 언어 모델이 사용되어 다국어 지원을 제공하며, 사용자가 요청한 언어에 따라 텍스트를 추출하고 인코딩한다. 또한 다국어 텍스트 인식을 위해 자연어 처리(NLP) 기술이 통합되어, 언어별 특성에 따라 텍스트를 정확하게 인식한다.

3.5 Restructuring

추출된 텍스트는 구문 분석 및 오류 교정을 거쳐 최종 결과로 제공된다. OCR의 텍스트 추출 결과는 JSON 형식으로 반환되며, 여기에는 텍스트의 내용, 위치 정보(좌표), 언어 정보 등이 포함된다. 사용자는 API 응답에서 텍스트 데이터를 연결하여 필요한 형태로 가공하거나, 데이터베이스에 저장할 수 있다.

III. The Proposed Scheme

1. Program Process

본 논문에서는 OCR 기술을 기반으로 하여 항공 산업체에서 사용되는 원소재 및 부품 성적서 등과 같은 서류의 추적성 검사를 자동화하여 효율적으로 수행할 수 있는 웹 프로그램을 Java 언어로 설계하였다. 모델의 프로세스는 다음 그림 2와 같다.

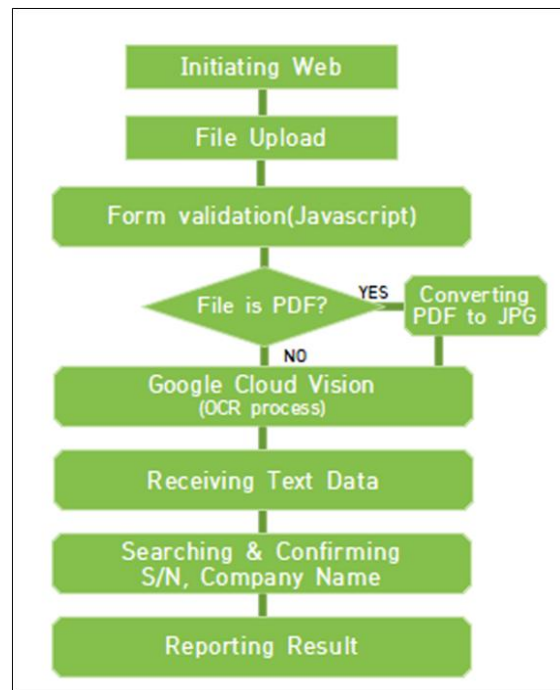


Fig. 2. Program Process

1.1 Initiating Web Browser

웹 브라우저에 접속하는 단계로 사용자가 웹 브라우저에 접근하면, html 파일을 제공한다. 브라우저 창과 코드는 다음 그림 3, 그림 4와 같다.



Fig. 3. Initiating Web Browser Window

```
@GetMapping("/")
public String getHome() {
    return "/main";
}
```

Fig. 4. Initiating Web Browser Code

1.2 File Upload

회사별 성적서의 스캔본 파일을 업로드 하는 것을 시작으로 한다. 회사별 파일을 업로드 했을 때에는 다음 그림 5와 같이 회사명과 Serial Number의 값을 입력할 수 있는 Input tag가 형성된다. 브라우저 창과 코드는 다음 그림 5, 그림 6과 같다.

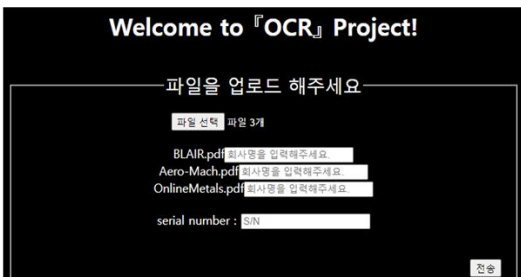


Fig. 5. Input tag for Company Name & S/N

```
let values = document.getElementById("values");
values.addEventListener("change", function() {
    let fileCount = values.files.length;
    // 기존 파일 리스트 초기화
    uploadedFile.innerHTML = "";
    for (let i = 0; i < fileCount; i++) {
        let fileNameDiv = document.createElement("div");
        let fileKeyInput = document.createElement("input");
        fileNameDiv.textContent = values.files[i].name;
        fileKeyInput.name = "companyName[" + i + "]";
        fileKeyInput.placeholder = "회사명을 입력해주세요.";
        fileNameDiv.appendChild(fileKeyInput);
        // 파일 이름을 uploadedFile 요소에 추가
        uploadedFile.appendChild(fileNameDiv);
    }

    let serialNum = document.createElement("input");
    serialNum.placeholder = "S/N";
    serialNum.id = "serialNumberInput";
    let serialNumBox = document.createElement("div");
    serialNumBox.innerHTML = "serial number : ";
    serialNumBox.appendChild(serialNum);
    serialNumber.appendChild(serialNumBox);
});
```

Fig. 6. File Selecting & Uploading Code

1.3 Form validation(Javascript)

본 코딩의 사용되는 Validation은 Client-side validation 방식으로 이루어진다. 이 방식은 폼이 서버로 제출되기 전에 즉각적인 피드백을 제공하여 사용자가 잘못된 데이터를 입력하더라도 서버로 전송되지 못하도록 도와준다. 앞서 이야기한 여러 Input tag를 For 문법을 통해 하나씩 접근하여 회사명 및 S/N를 전부 입력하지 않고 전송을 눌렀을 시 경고창을 띄워 입력값의 누락을 사용자에게 알리고, False를 반환하여 서버에 제출되지 않도록 한다. 모든 Input tag의 값을 입력했다면, 사용자가 업로드한 파일과 입력된 회사명, S/N를 서버에 전송할 준비를 한다. 브라우저 창과 코드는 다음 그림 7, 그림 8과 같다.



Fig. 7. Alert Message Box for null

```
for(let i = 0 ; i < companyNames.length; i++){
    if(companyNames[i].value == null || companyNames[i].value == ""){
        alert("회사명을 입력해주세요.");
        return false;
    }else{
        nameArr.push(companyNames[i].value.toUpperCase());
        formData.append("image", files[i]);
    }
}

if(serialNumber == null || serialNumber == "" || serialNumber.trim() == ""){
    alert("시리얼 넘버를 입력해주세요.");
    return false;
}
```

Fig. 8. Alert Code

1.4 Preparing Submission

구글 클라우드 비전 OCR은 JPG, PNG, GIF 등 이미지 파일만을 읽을 수 있다. 따라서 불러온 파일의 확장자가 PDF일 경우 전송 전에 PDF 파일이 이미지 파일로 변환될 수 있게 준비한다. Form에 입력된 파일과 값들을 클라우드 비전 서버로 전송하기 위해 준비하는 단계 중 PDF를 이미지로 변환하는 코드를 작성하였다. 전송은 Ajax 통신을 사용하여, /imageSubmit이라는 경로에 post 방식으로 formData를 전송한다. 전송 준비 코드는 그림 9와 같다.

```

@ResponseBody
@PostMapping("/imageSubmit")
public Object imageUrl(
    @RequestParam("image") List<MultipartFile> files,
    @RequestParam("serialNumber") String serialNumber,
    @RequestParam(value = "nameArr", required = false) List<String> companyNames
) throws Exception {

    Map<String, MultipartFile> myMap= new HashMap<>();

    for(int i = 0 ; i < files.size() ; i++) { //회사이름, 파일이 저장되도록
        myMap.put(companyNames.get(i).toUpperCase(), files.get(i));
    };

    for(String key : myMap.keySet()){ //map 순회하면서 파일이 pdf면 이미지로 변환
        MultipartFile file = myMap.get(key);
        if(myMap.get(key).getContentType().equals("application/pdf")){
            MultipartFile img = visionService.processPdfFile(file);
            myMap.put(key, img);
        }
    }

    S.ajax({
        url: '/imageSubmit',
        method: 'post',
        data : formData,
        processData: false, // jQuery가 데이터를 문자열로 처리하지 않도록 설정
        contentType: false, // jQuery가 contentType 헤더를 설정하지 않도록 설정
        success:(data){
            send.remove();
            console.log(data);
            for (const [key, value] of Object.entries(data)) {
                // value가 null이거나 빈 문자열이 아닌 경우에만 수행
                if (value !== null && value !== "") {
                    // 새로운 div 요소 생성
                    const div = document.createElement("div");
                    div.innerHTML =
                        `<div style="font-size:25px">${key}</div><br>
                        <div>${value}</div><br>
                        <div>Lot : ${serialNumber}</div>
                        <div style="height:60px"></div> `; // key와 value 값을 div에 넣음

                    // 화면에 추가 (해; body 안에 추가)
                    result.appendChild(div);
                }
            }
            error(){
                console.log("error");
            }
        }
    });
}

```

Fig. 9. Preparing Submission

1.5 Submitting Image File to Cloud Vision(OCR)

앞서 준비된 회사명과 S/N, PDF 파일일 경우 변환된 이미지 파일을 모두 서비스단(Back End)으로 전송한다. 이후에 서비스단에서 Google Cloud Vision으로 이미지 파일을 전송하여 이미지 파일을 텍스트로 변환하는 작업을 수행하고 OCR을 통해 변환된 문자열, S/N, 회사명을 텍스트로 변환하는 작업을 마치면 다시 서비스 단으로 전송한다. 코드는 그림 10과 같다.

```

public String extractTextFromImageUrl(MultipartFile file) throws Exception {
    ByteString imgBytes;

    try (InputStream in = file.getInputStream()) {
        imgBytes = ByteString.readFrom(in);
    }

    Image img = Image.newBuilder().setContent(imgBytes).build();
    Feature feat = Feature.newBuilder().setType(Feature.Type.TEXT_DETECTION).build();
    AnnotateImageRequest request = AnnotateImageRequest.newBuilder().addFeatures(feat).setImage(img).build();
    List<AnnotateImageRequests> requests = new ArrayList<>();
    requests.add(request);

    try (ImageAnnotatorClient client = ImageAnnotatorClient.create()) {
        BatchAnnotateImagesResponse response = client.batchAnnotateImages(requests);
        StringBuilder stringBuilder = new StringBuilder();
        for (AnnotateImageResponse res : response.getResponsesList()) {
            if (res.hasError()) {
                System.out.printf("Error: %s\n", res.getError().getMessage());
                return "Error detected";
            }
            stringBuilder.append(res.getFullTextAnnotation().getText());
        }
        return stringBuilder.toString();
    } catch (Exception e) {
        System.out.println(e);
        return e.toString();
    }
}

Map<String, String> resultMap = new HashMap<String, String>();

for(String key : myMap.keySet()) {
    String result = visionService.extractTextFromImageUrl(myMap.get(key));
    resultMap.put(key, result.toUpperCase().trim());
}

Object process = visionService.findSerialNumber(resultMap, serialNumber, companyNames);

return process;

```

Fig. 10. Submitting Image File to OCR code

1.6 Searching Traceability Data

우선적으로 서류상에 시리얼 넘버가 있는지를 확인한다, 변환된 텍스트 Data 안에 입력한 시리얼 넘버가 존재하면 브라우저 창으로 시리얼 넘버를 반환시켜준다. 코드는 다음 그림 11과 같다.

```

for (String key : map.keySet()) {
    String text = map.get(key);
    if (text.contains(serialNumber.toUpperCase())) {
        process.put(key, "");
    }
}

```

Fig. 11. Searching S/N code

한 회사에서 어느 회사로 부품 및 원소재를 보내었는지 추적성을 확인하기 위해서는 일반적으로 “SOLD TO~, SHIP TO~, BILL To~” 와 같은 문구 뒤에 부품을 받는 회사명을 확인한다. OCR을 통해 텍스트로 변환되어 온 Data 안에 지정된 문자열(SOLD TO, SHIP TO, BILL To)이 있는지 확인하기 위해 배열을 선언 및 값을 지정한다. 그 배열의 단어를 찾았다면 찾은 단어 이후 25글자까지의 텍스트를 한 번 더 추출한다. 이후 추출된 25글자 안에 입력했던 회사명이 있는지 찾았다면 해당 서류의 회사가 어느 회사로 제품을 보냈는지 브라우저 창에 반환한다. 최종 수행 브라우저 창과 코드는 다음 그림 12, 그림 13과 같다.

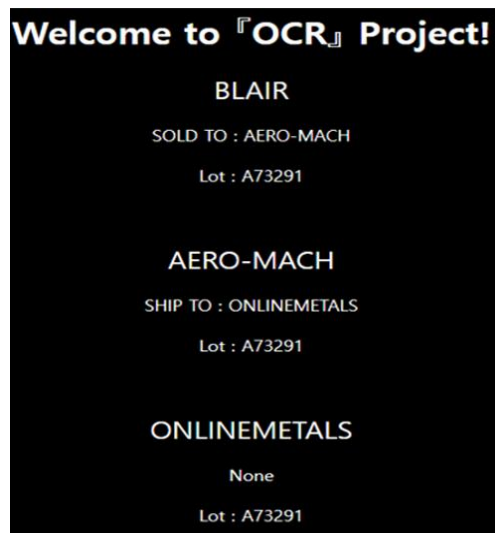


Fig. 12. Reporting Result Browser Window

```

public Object findSerialNumber(Map<String, String> map, String serialNumber, List<String> companyNames) {
    Map<String, String> process = new HashMap<>();
    String[] to = { "SOLD TO", "SOLDTO", "SHIP TO", "SHEPTO", "BILL TO", "BILLTO" };
    System.out.println("companyNames : " + companyNames);
    for (String key : map.keySet()) {
        String text = map.get(key);
        if (text.contains(serialNumber.toUpperCase())) { //문자 세로 읽기 시도를 남기 문제를 확인함.
            process.put(key, "");
        }
    }
    for (String key : process.keySet()) {
        String text = map.get(key);
        for (String keyword : to) {
            String conName = processText(text, keyword, companyNames);
            if (conName != null) {
                process.put(key, conName);
            }
        }
    }
    System.out.println(process);
    return process;
}

public String processText(String text, String keyword, List<String> companyNames) {
    String tempStr = text.toUpperCase();
    if (tempStr.contains(keyword)) {
        int startIndex = text.toUpperCase().indexOf(keyword);
        int endIndex = Math.min(startIndex + 25, text.length());
        String subStr = text.toUpperCase().substring(startIndex, endIndex);
        for (String conName : companyNames) {
            if (subStr.contains(conName.toUpperCase())) {
                return keyword + " : " + conName; // Exit early if a match is found
            }
        }
    }
    return null;
}

```

Fig. 13. Reporting Result code

IV. Conclusions

본 논문에서는 원소재의 추적성을 효율적으로 확인할 수 있는 OCR활용 웹 프로그래밍을 구현했다. 총 3개의 회사와 6장의 서류를 대상으로 하여 실행한 결과 본 논문에서 구현한 프로그램으로는 S/N의 유무를 확인하고 회사의 추적성이 연결되는지 확인하기까지 약 1분이 걸리지 않았다. 사람의 경험과 환경에 따라 다르겠지만 일반적으로 6개월 이상 숙련된 검사원이 6장의 서류를 보며 정보를 정확히 확인하는데 3분가량의 시간이 필요하다. 다만 정확한 효율을 계산하기 위해서는 정량적인 실험을 통하여 비교분석을 해 볼 여지가 필요하다.

제시한 프로그래밍은 신뢰성이 100% 완벽하다고 할 수 없다. 회사마다 성적서의 포맷이나 글꼴 및 구조가 다양하고, 스캔 된 서류의 상태에 따라 OCR의 인식률이 달라지기 때문이다. 또한, Google Cloud Vision OCR의 사용에 비용적인 문제와 클라우드 기반으로 데이터를 주고받기에 제품 정보와 회사의 민감한 정보가 노출될 수 있는 문제도 존재한다. 향후 정량적인 효율성 분석과 보안 취약점을 강화 하기 위한 추가적인 연구가 필요할 것으로 보인다.

REFERENCES

- [1] Google Cloud Vision official document. <https://cloud.google.com/vision>
- [2] Sundararajan, S., & Chae, S. (2020). "Deep Learning Applications in Image Processing: A Comprehensive Review." *Journal of Computer Vision*.
- [3] Kumar, V., & Mishra, A. (2021). "Cloud-Based Image Recognition: Comparative Analysis of AWS Rekognition, Google Cloud Vision, and Azure Computer Vision." *Cloud Computing Review*.
- [4] Apache PDFBOX (pdf to image) <https://jin3260.tistory.com/m/40>
- [5] Jae-hyeon Han, 2022 Aviation Industry Research and Analysis, Korea Transport Institute, 2022, p.120
- [6] How Does Optical Character Recognition Work. <https://www.baeldung.com/cs/ocr>
- [7] IAQG SCM. 9100:2016 Series Clarification Table. Resources 9100:2016-Series-QMS: Aviation, Space and Defense Organizations Standards Clarifications. <https://scmh.iaqg.org/standard/9100-qms-requirements-for-aviation-space-and-defense-organizations/>
- [8] IAQG SCM. SCM. 3.6.2 Human Factors in New Manufacturing Rev D Dated 20FEB2024.
- [9] WIKIPEDIA. Type approval or certificate of conformity. https://en.wikipedia.org/wiki/Type_approval
- [10] Kim Kyu Hyeong, Kang Seong Hwan, & Cho We Duke (2023-02-08). Development of OCR-based algorithm for information extraction from Material Safety Data Sheets (MSDS). *Proceedings of Symposium of the Korean Institute of communications and Information Sciences, Gangwon State*.
- [11] JaidedAI. (2023). EasyOCR.GitHub, <https://github.com/JaidedAI/EasyOCR>
- [12] Sang-Ho Han (2003). Condition of Quality Management System in Aerospace industry and AS 9100. *The Journal of Aerospace Industry*, 64, 72-88.
- [13] Bong-gu Jeong (2016) The Overview of AS 9100 Rev. D.
- [14] SAE, AS 9100D "Quality Management Systems - Requirements for Aviation, Space, and Defense Organizations", Sep. 2016

Authors



Dong-Won Park received the B.S. degrees in Academic Credit Bank System in 2023. Currently, the Master's degree program in Information Technology Convergence, Graduate School of Technology &

Convergence, Kongju national university, Chungnam, Korea, in 2023. He is currently a teaching in the Department of Aviation Maintenance. University of Gyeongnam Geochang. He is interested in Aviation Maintenance, project management systems, automated testing, algorithms, machine learning, and automated systems.



Myung-Seob Yoon received the B.E. degree in Korea national open university, Seoul, Korea, in 2005, and the M.S. degree in Kongju national university, Chungnam, Korea, in 2016 and Ph.D. degrees from the

same university candidate in 2019, And he has been with Div. of Aviation maintenance of University of Gyeongnam Geochang, Korea, as a professor from 2017. He interested in Avionics, Aviation Maintenance and ATE.



Seong-Hyun Park received the B. S. degree in College of Arts and Music from Chungnam National University, Korea in 2017. The M. S, and Ph. D. degrees in Computer Engineering from Kongju National

University, Korea, in 2017, 2020. He is currently a teaching in the Department of Computer Science & Engineering. Kongju National University. He is interested in computer music, convergence Education. and Real Time System and Management and Cloud computing and Communication.