

III. 공개SW프로젝트 결과보고서

프로젝트명 : 영수증 이미지 인식을 통한 자동 정산 서비스

교과목명	공개SW프로젝트(02)
담당교수	송수환 교수님 이강만 교수님
팀 장	모수진
팀 원	박현수
	최희수
	하승연

시연 영상	
배포 URL	https://github.com/OSSW7guys/2025-1-OSSP-chillguys
테스트계정	

□ 결과보고서 요약

<p>프로젝트 추진배경 및 필요성</p>	<p>친구, 가족 모임과 같은 소규모 인원에서부터 동아리나 단체 모임과 같은 대규모 인원까지, 여러 인원이 함께 식사나 여행 등의 활동을 할 때 각자가 소비한 금액을 공평하게 나누는 방식인 더치페이는 사회적으로 큰 관심을 끌고 있다. 이에 따라 결제 서비스를 제공하는 금융 기업들은 기존 어플에 더치페이 서비스를 추가하는 등, 확산되는 수요에 맞춰 변화하였다. 그러나 이러한 서비스에도 사용자에게 여전히 불편함이 존재하고, 간단한 정산에 한정된 기존 서비스는 복잡한 결제 내역이나 회계 내역 등을 처리하기에는 한계가 있다. 따라서 본 프로젝트는 영수증 이미지 인식을 통한 자동 정산 웹 서비스를 개발하여 이러한 문제를 해결하고자 한다.</p>
<p>프로젝트 최종목적 및 세부 목표</p>	<p>본 프로젝트의 주 목표는 영수증 이미지 인식을 활용한 웹 서비스로 편리하고 정확한 정산 과정을 제공하는 것이다. 세부 목표는 다음과 같다.</p> <ul style="list-style-type: none"> • 영수증 이미지 인식 기술 구축 • 추출 데이터 활용을 위한 데이터베이스 구축 • 간편한 정산 기능 • 정산 내역 공유 및 관리 시스템 구축
<p>기대효과</p>	<p>기술적 측면</p> <ul style="list-style-type: none"> • 한글 영수증 인식에 최적화된 엔진을 선정하고 실용성을 확보 • 이미지 해상도 개선, 노이즈 제거 등 입력된 이미지에 대한 전처리 과정을 추가하여 OCR 기술의 성능 개선 • 숫자 인식과 한글 인식을 위한 후처리 로직을 구현하여 정확도 향상 <p>경제, 산업적 측면</p> <ul style="list-style-type: none"> • 일상적 회계 업무의 부담 경감 • 영수증/거래 명세서의 디지털화를 통한 회계 업무 효율성 증진 • 회계나 경리 자동화 시장에 진입할 수 있는 기반 마련
<p>키워드 (Keyword) (국문)</p>	<p>광학 문자 인식(OCR), 유사도 비교, 한글 인식, 이미지 전처리, 정산 자동화</p>
<p>키워드 (Keyword) (영문)</p>	<p>Optical Character Recognition(OCR), Text Similarity, Hangeul Recognition, Image Preprocessing, Auto-Settlement</p>

목 차

I. 프로젝트 과제의 필요성	2
1. 과제 개요 및 추진배경	2
2. 과제 목적 및 내용	3
3. 과제 범위	4
II. 프로젝트 목적 및 내용	4
1. 프로젝트의 최종목적	4
2. 프로젝트의 세부 목표	4
3. 프로젝트의 개발 및 연구내용	5
4. 프로젝트의 주요 결과물	14
III. 추진일정	17
1. 추진전략 및 방법	17
2. 추진체계	18
IV. 프로젝트의 활용 방안 및 기대효과	19
1. 기대효과 및 수익성	19
2. 활용 방안	19
V. 참여 인력	20
1. 업무 분담	20
2. 팀원 별 수행 성과	21
VI. 참고 문헌	22

제 1 장 프로젝트 과제의 필요성

제 1 절 과제 개요 및 추진 배경

1. 현황 및 문제의식

친구, 가족 모임과 같은 소규모 인원에서부터 동아리나 단체 모임과 같은 대규모 인원까지, 여러 인원이 함께 식사나 여행 등의 활동을 할 때 각자가 소비한 금액을 공평하게 나누는 방식인 더치페이는 사회적으로 큰 관심을 끌고 있다. 더치페이 문화의 확산에 맞춰 카카오톡, 토스, 농협 등 다양한 모바일 결제 서비스를 제공하는 금융 기업들은 각자만의 더치페이 기능을 제공하고 있다. 예를 들어, 농협은행은 더치페이 기능을 제공하는 어플을 제공하며 2달 만에 20만명의 가입자를 확보한 바가 있으며, 신세계는 자사의 SSG페이 서비스에 더치페이 기능을 추가하는 등 현재의 문화를 따라 변화하고 있다. 이와 같은 더치페이 서비스의 확대는 더치페이의 사회적 수요가 점차 커지고 있음을 보여주는 방증이라고 할 수 있다.

그러나, 현재 제공되고 있는 다양한 서비스에도 여전히 사용자들에게 불편함이 존재한다. 일반적으로 모임에서 결제를 진행할 때 한 사람이 모든 비용을 지불하고 참여자들에게 이후에 전체 금액의 1/N이나 메뉴별 비용을 청구하는데, 이 과정에서 결제한 영수증을 하나씩 확인하고 개별적으로 요청하는 데 많은 시간이 소요된다. 카카오톡 ‘정산하기’ 기능처럼 일부 디지털 서비스가 등장하면서 정산 과정이 다소 간소화되긴 하였으나, 여전히 간단한 1/N 정산에 한정되어 복잡한 결제 내역이나 회계 내역 처리, 여행과 같은 다양한 항목이 포함된 다수의 결제 처리 등에 있어 한계가 존재한다.

2. 과제 필요성

제시된 문제를 해결하기 위하여 본 프로젝트는 영수증 이미지 인식을 통한 자동 정산 웹 서비스를 개발하고자 한다. 사용자가 촬영한 영수증을 업로드하면, 이미지를 자동으로 분석하여 메뉴, 금액, 총액을 정확하게 인식하고 각자 부담해야 할 금액을 손쉽게 산출하여 공유할 수 있도록 한다. 이를 통해 정산 과정에 들어가는 시간과 노력을 절약하고, 정확한 정산을 편리하게 처리할 수 있는 서비스를 사용자에게 제공한다.

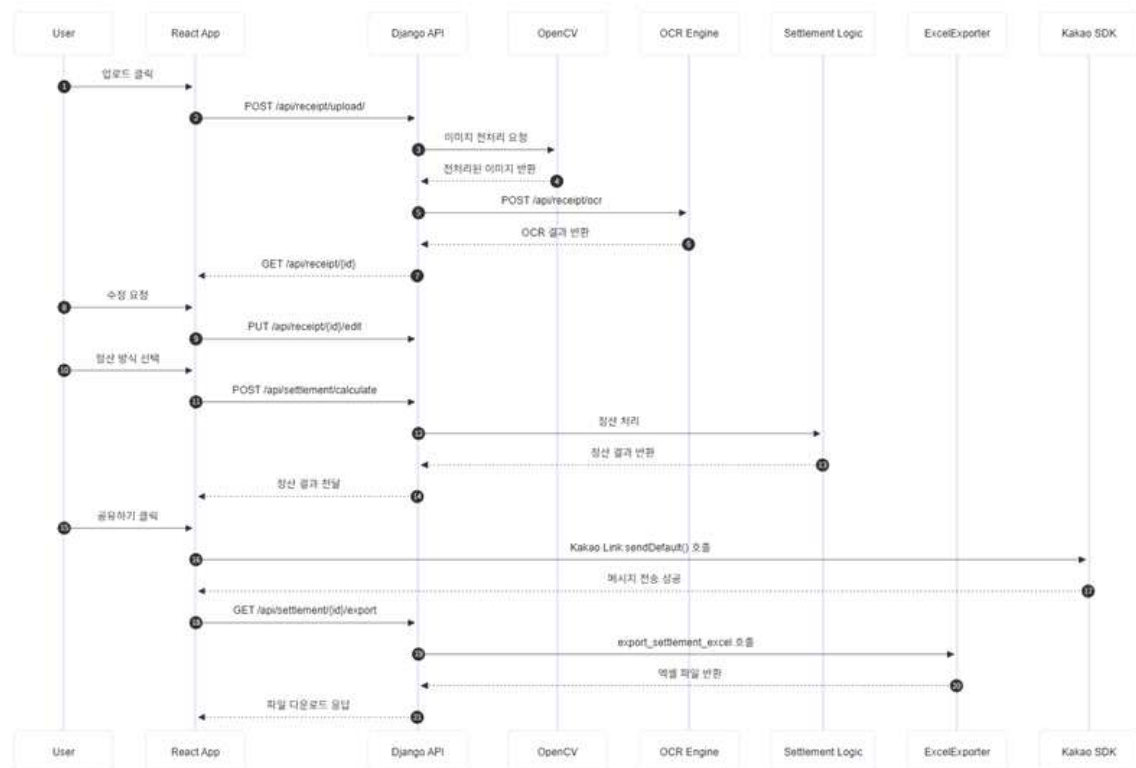
제 2 절 과제 목적 및 내용

1. 과제 목적

본 프로젝트는 편의성과 정확성을 중점으로 하는 정산 서비스를 사용자에게 제공하는 것을 목적으로 한다. 따라서 서비스 이용에 있어 직관적이고 간편한 UI 및 기능을 구현하여 사용자에게 편의성을 보장하고, 오픈소스 OCR 엔진 간의 비교로 한글 인식률이 높은 엔진을 선정하여 전, 후처리 과정을 포함한 문자 추출을 통해 정확성을 보장하고자 한다.

2. 과제 내용

본 프로젝트에서는 OCR 기술을 활용하여 영수증에서 텍스트 정보를 자동 추출하고, 이를 기반으로 정산을 수행하는 웹 기반 시스템을 구현하고자 한다. OCR 인식 오류를 낮추기 위해 입력 데이터로 주어지는 이미지의 전처리 과정을 거치고, OCR 인식 결과에 대해 사전 기반 후처리 알고리즘을 적용하여 정확도를 향상시킨다. 추출한 영수증 정보, 사용자 정보, 정산 내역을 저장 및 조회할 수 있도록 데이터베이스 구조를 설계하고 API를 연동할 수 있게 구현한다. 사용자 인터페이스(UI)를 통해 정산할 항목과 대상을 편리하게 입력 및 수정할 수 있도록 하며, 항목별 정산과 1/N 정산을 모두 지원하도록 설계할 것이다. 또한, 정산 결과를 카카오톡 메시지로 공유하거나 Excel 파일로 저장할 수 있는 기능을 구현한다.



[그림 1] 시퀀스 다이어그램

3. 과제 수행 방법

프론트엔드 프레임워크
 백엔드 전체 개발 언어
 백엔드 프레임워크
 이미지 전처리
 OCR 엔진
 파싱 후처리
 DB
 협업

React, TypeScript, Tailwind
 Python 3.x
 Django REST Framework
 OpenCV
 EasyOCR
 Levenshtein
 MySQL
 Swagger, Notion, Git(Github), Discord

제 3 절 과제 범위

1. 사업적 측면

본 프로젝트는 오픈소스 OCR 기술을 활용하여 정산 처리 자동화 시스템을 구축하고 사전에 기반하여 텍스트를 정제한 결과를 얻음으로써 사용자에게 보다 편리하고 정확한 정산 서비스를 제공하고자 한다. 영수증 이미지를 입력하여 자동으로 결과를 추출하기 때문에 사용자는 수기로 입력하는 노력을 줄이고 데이터가 누락되는 일을 방지할 수 있다. 이에 더해, 향후 기능 추가 및 확대를 통해 회계 보조 도구나 API 서비스로 확장할 수 있다.

2. 연구적 측면

본 프로젝트는 전처리 과정을 거친 영수증 이미지를 OCR 기술을 통해 추출하고 추출 결과에 대해 후처리 기술을 적용해 최종 인식 결과를 도출한다. 따라서 기존 오픈소스 OCR 엔진 간의 분석 결과 비교와 후처리 과정을 통한 정확도 향상에 중점을 두고자 한다.

주요 연구 내용은 다음과 같다.

- 다양한 OCR 엔진의 인식 결과 비교 및 분석
- 제한된 포맷의 영수증 인식 후 사전과 매칭하여 정답을 결정하는 방식의 유효성 비교 및 분석
- 인식 오류 유형에 따른 보정 알고리즘 설계
- 전, 후처리 유무에 따른 정확도 분석 및 실용성 검토

제 2 장 프로젝트 목적 및 내용

제 1 절 프로젝트의 최종 목적

본 프로젝트의 주 목적은 영수증 이미지 인식을 활용한 웹 서비스로 간편하고 정확한 정산 과정을 지원하는 것이다. ‘영수증 인식 → 개인화된 정산 금액 산출 → 자동 정산 및 공유 → 그룹 관리 및 회계 문서 활용’이라는 일련의 과정을 하나의 웹 서비스로 구축하여, 복잡한 정산 문제를 편리하고 투명하게 해결할 수 있는 서비스를 제공하는 것을 목표로 한다.

제 2 절 프로젝트의 세부 목표

가. 영수증 이미지 인식 기술

본 프로젝트의 대표 기술인 영수증 이미지 인식 기술을 구축하고, 정밀하게 추출한 데이터를 데이터베이스에 저장하여 이후 정산 및 관리에 활용할 수 있어야 한다. 이에 PC, 모바일 모두 편리하게 접근 가능한 직관적인 반응형 웹 서비스 환경을 구축하고, 영수증 이미지를 분석하여 메뉴 이름과 금액 정보를 추출할 수 있는 알고리즘 및 라이브러리를 연구하여 개발하도록 한다. 여러 오픈소스 OCR 엔진을 비교 분석하여 최적의 OCR 엔진을 선정하고,

저장한 데이터베이스를 바탕으로 정확한 계산 처리를 위한 정산 로직을 구현한다.

나. 간편한 정산 기능

누구나 사용하기에 간편한 정산 기능을 제공하여야 한다. 이때 총액 1/N 정산 기능과, 각 참여자가 소비한 메뉴를 선택하고 부담해야 하는 금액을 계산하는 메뉴별 정산 기능을 구현하고자 한다

다. 공유 및 관리 시스템

각 참여자의 정산 금액과 세부 내역, 계좌 번호, 영수증 이미지 등을 통합하여 카카오톡 메시지 API를 활용한 자동 메시지로 전송·공유할 수 있도록 한다. 더불어, 그룹 폴더를 통해 팀 단위 여러 결제 내역을 관리하고 함께 열람하며 추후 회계 감사 등에 활용할 수 있게 한다. 특히 회계 감사 템플릿에 따른 .xlsx 형태로 추출할 수 있도록 엑셀 파일로 내보내기 기능을 구현하여 회계 장부 정리나 감사를 위한 간편화 기능을 제공함으로써 사용자 편의를 극대화하고자 한다.

제 3 절 프로젝트의 개발 및 연구 내용

가. 영수증 이미지 인식 기술

■ 오픈소스 OCR 엔진 비교 분석

프로젝트 진행 초기 단계에서의 기존 목표는 오픈소스 OCR 엔진 중 하나인 Google Tesseract OCR에 영수증 특화 모델을 학습시켜 사용하는 것이었다. 오픈소스 OCR 엔진 중 비교적 학습이 쉽고 대중적이었기 때문에 영수증에 사용되는 폰트와 유사한 GmarketSans 폰트를 학습시켜 영수증 특화 모델로 사용하기로 계획하였다. 학습에 사용할 BOX 파일 및 학습 데이터 생성은 성공하였으나 학습이 제대로 이루어지지 않았다. 이에 당초 목표인 영수증 특화 모델을 학습시키는 것이 아니라, 기존의 오픈소스 OCR 엔진 중 영수증 이미지의 인식률, 특히 한글 인식률을 비교하여 가장 적절한 엔진을 선정하는 것으로 목표를 수정하였다. Google Tesseract, easyOCR, paddleOCR 세 개의 오픈소스 OCR 엔진을 대상으로 비교를 진행하였으며, 영수증 이미지에 대한 추출 결과는 아래의 표와 같다.

세 엔진 모두 숫자와 영문 인식 정확도는 높았으나, 한글 인식에 있어서는 비교적 오답이 많은 것을 확인할 수 있다. 특히 Google Tesseract의 경우, 영수증의 구분선과 같은 불필요한 문자도 인식하여 추출하기 때문에 결과로 사용하기에 부적절하다고 판단하였다. easyOCR과 paddleOCR을 비교하여, paddleOCR의 한글 인식 정확도가 상대적으로 높지만 단어 인식 단위가 지나치게 세분화되어 하나의 단어로 인식되어야 하는 표현도 개별의 단어로 분리된다. 본 프로젝트에서 사용할 데이터는 영수증 이미지로, 상호명이나 품목 등 두 개 이상의 단어로 이루어진 표현이 다수 포함되기 때문에 이를 개별의 단어로 분리하여 인식하는 paddleOCR은 프로젝트의 목적에 부합하지 않는다. 이러한 이유로, 본 프로젝트에서는 easyOCR 엔진을 채택하여 사용하기로 결정하였다.

no	Google Tesseract	easyOCR	paddleOCR
1	eat Sut	동국학교 플바켓	동국대학교
2	Fas	유원품	플바셋
3	전화번호: 021234567	전화번호: 021234567	유원증
4	사업자번호: 6168171411	사업자번호: 6168171417	전화번호
5	POS01	PO:OI	021234567
6	판매 담당 : 관리자	판매 담당	사업자번호
7	테이블	관리자	6168171411
8	x	테이다	POS01
9	2oreree	모장	만매
10	2025/04/30 14:36:16	2025/04/30	담당
11	수 번호 : 58	14: 36;16	관리자
12	=	영수 번호	테이블
13	SSSR Sro Sse saasasscxmssoss	58	포장
14	Dasicesecs erro te eee eee	ICE 카페라떼	2025/04/30
15	Swen moo 우우우	2,700	14:36:16
16	ICE 카페라떼	2, 700	영수
17	2, 700	주문함계	번호
18	2, 700	2,700	:
19	=	반 금액	-
20	SSSSBR LSTA La sasasssirsnsnsstssstssssas	700	58
21	titer ri eee ee = 2 = = ==	2,	뿌고는
22	2H	금m가액	ICE
23	2, 700	2,455	카페라떼
24	el 을 SCH	부가세액	2700
25	2, 700	245	2 700
26	공급가액	현재금액	주문함계
27	2, 455	700	2 700
28	부가세액	2 .	금액
29	245	카	2
30	SE mene renner neintine eee micerinuttiewienadenine 서시 49641	드	700
31	2, 700	2, 700	공급가액
32	Wa A ' re	중인 카드	2
33	2, 700	KB국민카드	455
34	BRLnSlssaaeassrrrssaasassossassaassaa sss	승인임시	부가세액
35	Sete ert ttre een ee mew eee ee eee	2025-04-30 14:36:16	245
36	승인 카드 - 1#국민카드	카드번호	결제금액
37	승인임시	46020557***#***	2
38	2025-04-30 14:36:16	승인번호	700
39	카드번호	30026984	카
40	4602055 7#xxKHHX	임시불	-
41	30026984	가행번호	2
42	승인번호	82408173	700
43	일시불	KB국민카드	구군폭돌프는
44	가맹번호	승인금액	승인
45	82408173	2,700	카드
46	/ 때국민카드		K8국민카드
47	승인금액		-
48	2, 700		승인임시
49	or eee es cir inter ene ne cso eee ee i eet ie re nw ao a ea ec		2025-04-30
50			14:36:16
51			카드 번호
52			46020557w*뽕
53			승인번호
54			30026984
55			일시불
56			가맹번호
57			82408173
58			KB국민카드
59			승인금액
60			2700

[표 1] OCR 엔진별 인식 결과 비교

지표	EasyOCR	Tesseract
Accuracy (단어 정확률)	86.7% (52/60)	45.0% (27/60)
CER (문자 오류율)	0.12	0.42
WER (단어 오류율)	0.18	0.55
Precision (가짜 인식)	0.91	0.64
Recall (재현율)	0.88	0.50
F1-Score (조화 평균)	0.895	0.56

[표 2] 핵심 성능 지표 분석

지표명	정의	계산 공식
Accuracy (정확도)	전체 인식 항목 중 정답과 정확히 일치한 항목 비율	정확히 인식된 항목 수 ÷ 전체 비교 항목 수
CER (문자 오류율)	문자 단위로 비교하여 삽입/삭제/대체된 문자 비율	(삽입 + 삭제 + 대체된 문자 수) ÷ 전체 문자 수 (정답 기준)
WER (단어 오류율)	단어 단위로 비교하여 삽입/삭제/대체된 단어 비율	(삽입 + 삭제 + 대체된 단어 수) ÷ 전체 단어 수 (정답 기준)
Precision (정밀도)	OCR이 추출한 특정 정보 중 실제로 맞는 비율	정확히 추출된 항목 수 ÷ OCR이 추출한 항목 수
Recall (재현율)	실제 존재하는 정보 중 OCR이 정확히 인식한 비율	정확히 추출된 항목 수 ÷ 실제 존재하는 항목 수
F1-Score (조화 평균)	Precision과 Recall의 조화 평균 값으로, 전체 성능 균형 지표	$2 \times (\text{Precision} \times \text{Recall}) \div (\text{Precision} + \text{Recall})$

[표 3] OCR 성능 평가 지표

■ 이미지 전처리

OCR 기술의 인식 정확도를 향상시키기 위해서는 사용할 이미지가 선명하고 문자의 구분이 명확해야 한다. 그러나 실제 서비스를 사용하는 과정에서는 다양한 품질의 이미지가 입력으로 주어질 것이며, 이 중에는 화질이 낮거나 문자 구분이 어려운 이미지도 포함될 가능성이 있다. 따라서 OCR 적용에 앞서 입력된 이미지에 대해 전처리 과정을 선행하여 인식 정확도를 높이려고 하였다.

이미지 전처리 과정의 핵심 목표는 첫째, 주어진 이미지에서 영수증에 해당하는 영역만 추출하는 것이고, 둘째, 노이즈 제거, 밝기 조절, 흑백 처리 등을 통해 문자와 비문자 영역을 명확히 구분할 수 있도록 개선하는 것이다. 이를 수행하기 위하여 OpenCV를 기반으로 여러 방법을 시도하였으며, 각 시도에 대한 주요 내용과 한계를 정리하였다.

1. 이미지 품질 개선 후 영수증 영역 추출

첫 번째 시도에서는 주어진 이미지를 ‘회색조로 변환 → 이진화(adaptive threshold) → 블러링(Gaussian Blur) → 모폴로지 연산(Morphological Close) → 영수증 추출(findContours) → 왜곡 보정(Perspective Transform)’의 절차로 전처리를 시도하였다. 그러나 findContours 함수가 실행되는 과정에서 예상과 다르게 영수증 추출이 제대로 이루어지지 않았다. 이진화 과정에서 adaptive threshold를 사용하는 대신 canny edge detection을 사용하는 것으로 변경하여 다시 시도하였으나, 경계선을 일부 감지하는 것에 그치고 영수증 윤곽을 정확하게 검출하지는 못하였다.

이에 전처리 과정을 ‘회색조로 변환 → Otsu Binarization → Dilate로 글자 강조 →

Morphological Open 및 Close 연산 순차 적용 → 영수증 추출(findContours) → 왜곡 보정(Perspective Transform)' 으로 수정하고 Contour 추출 시 마스크 영역을 사용하도록 다시 진행하였다. 그 결과, 글자 인식은 개선되었으나 영수증 전체 영역의 윤곽을 검출하는 것에는 실패하였다.

여러 차례 전처리 과정을 조정하였음에도 영수증 영역 전체를 안정적으로 추출해내는 것에 한계가 있었다. 문자 인식을 위한 이미지 품질은 개선할 수 있었으나, 이미지 내에서 영수증 영역을 정확히 분리해내는 목표는 이루지 못하였다. OCR 전처리의 일반적인 흐름대로 수행하는 방식은 영역 검출을 고려하지 않은 절차이기 때문에 본 프로젝트의 목표를 달성하기에 적절하지 않다고 판단하였다.

2. 영수증 영역 추출 후 이미지 품질 개선

첫 번째 시도에서는 크게 영수증 영역 추출과 텍스트 전처리로 단계를 나누어 진행하였다. 영수증 영역을 추출하는 과정은 '이미지 크기 조정 → 회색조로 변환 → 블러링(Gaussian Blur) → 영수증 추출(findContours)'의 순서로 진행하였고, 텍스트 전처리는 'Blackhat → Sobel → OTSU 이진화 → Morphology 연산 → 텍스트 블록 그룹핑 및 ROI 저장'의 순서로 진행하였다. 영수증 영역을 추출할 때 블러링 과정에서 노이즈 등을 제거할 수 있도록 하였으며, findContours로 영수증을 추출할 때는 Canny Edge Detection을 사용해 Contour를 검출하고 가장 큰 사각형을 기준으로 영수증이 추출되도록 설계하였다. 그러나 사각형으로 추출되는 영역이 없어 실패하였고, 분석 결과 사용한 이미지에 촬영 각도 등으로 인한 왜곡이 존재하였기 때문에 영수증 영역이 사각형으로 인식되지 않은 것으로 결론지었다.

다음 시도에서는 minAreaRect 방식을 적용하는 것으로 수정하여 진행하였고, 시도 결과 영수증 영역만을 추출하는 것에는 실패했지만 이미지 전체를 아웃라인으로 인식하여 반환하였기 때문에 영수증 일부가 잘리는 등의 데이터 손실이 발생하는 경우는 없었다. 이후 minAreaRect 기반이 되 아웃라인 크기가 사진 면적의 50% 이하라면 사진 전체 반환하도록 조건을 수정하여 실행하였는데, 여전히 영수증의 아웃라인이 잘 잡히지 않는 문제가 발생하였다. 이미지 내의 영수증이 명확한 사각형의 형태가 아니기 때문에 아웃라인 기반으로 영수증 영역을 추출해내는 방식은 채택하기 어렵다고 판단하여, 사각형의 아웃라인 검출 대신 색상 대비로 영수증 영역을 추출해보는 대안을 생각하였다. 그러나 이 방식 역시 그림자나 영수증과 배경 사이의 색상 낮은 경우 영역 검출에 실패할 가능성이 높아 적절한 대안이 아니라고 판단하였다. 텍스트 블록 그룹핑만으로 대체하는 방식은 이론상 가능하겠지만, 기준이 민감하여 인식이 제대로 되지 않은 일부로 인해 영수증 전체가 누락되는 위험성이 존재한다.

여러 방식으로의 이미지 전처리 시도 후 영수증 영역 검출은 데이터의 왜곡에 민감한 방식이라는 결론을 내렸으며, 서비스에 사용될 데이터가 왜곡 가능성이 높은 이미지이기 때문에 영수증 영역 검출의 정확도를 기대하기 어렵다고 판단하였다. 따라서 전처리 단계에서는 노이즈 제거 등 텍스트 강조에 집중하고, OCR 이후 후처리 단계에서 필요한 텍스트를 정제해 내 정확도를 높이는 방향으로 계획을 재조정하였다.

■ OCR 인식 결과에 대한 후처리

전처리된 이미지에 대하여 별도의 후처리 없이 OCR만 적용할 경우, 다음과 같은 문제점이 발생할 수 있다.

1. 가격 인식의 정확도 : 쉼표(,) 표시가 없거나, 숫자 '0' 을 알파벳 'O' 로 인식하는

등의 문제가 발생하여 가격 정보의 정확도가 저하된다.

2. 한글 인식의 정확도 : 추출된 텍스트에서 오타자 및 부정확한 띄어쓰기로 인해 품목명 등의 정보가 정확하게 추출되지 않는 경우가 있다.
3. 수량 인식 누락 : 수량 정보에 대한 값이 보통 숫자 1처럼 단독으로 표기되어 있기 때문에, 맥락 없이 분리되어 인식되지 않거나 무시되는 문제가 발생한다.
4. 불필요한 제품번호로 인한 혼동 : 영수증에 포함된 제품번호와 같은 숫자가 가격으로 잘못 인식될 수 있어 가격 파싱의 정확도가 저하된다.

이러한 문제들은 OCR 인식 결과의 신뢰도에 직접적인 영향을 미치며, 본 프로젝트의 핵심 목표인 정확한 정산 과정 구현을 위해서는 반드시 선행되어 해결해야 할 문제이다. 따라서, 이와 같은 문제점을 해결하고자 각 문제점에 대응하는 해결 방안을 설계하고 후처리 과정에 적용하였다. 이하에서는 구체적인 후처리 방안을 항목별로 설명한다.

1. 가격 인식 정확도 향상

OCR 결과에서 가격 정보에 대한 오류는 정산의 정확도를 크게 저하하기 때문에 후처리 정교화가 필수적이다. 이를 해결하기 위하여, 우선적으로 기본적인 오류 처리를 적용하였다. 숫자 '0' 이 알파벳 'O' , 'U' , 'o' 등으로 잘못 인식되는 경우를 보정하기 위하여, 유사한 형태의 문자가 숫자 '0' 으로 치환되도록 처리하였다. 또, 천 단위 구분 기호인 쉼표(,)가 생략되거나 마침표(.)로 대체되는 경우를 보정하기 위하여, 잘못 인식된 마침표 혹은 숫자 사이의 불필요한 공백을 쉼표로 대체하여 원래의 형태로 복원하였다. 그러나 이러한 처리 방식은 한 줄에 두 개 이상의 숫자 형식이 존재할 경우(예: 900 900) 오히려 잘못된 금액으로(예: 900,900) 보정될 수 있다. 따라서 다음의 조건 기반 세부 로직을 추가로 적용하였다.

두 숫자 사이에 공백이 존재할 때 앞의 숫자가 1~2자리, 뒤의 숫자가 3자리인 경우, 이는 쉼표가 생략된 것으로 보고 공백을 쉼표로 대체하도록 하였다.(예: 9 500 → 9,500) 금액이 10원 또는 1원 단위인 제품은 현실적으로 거의 없기 때문에, 이와 같은 형태는 천 단위 금액일 가능성이 높다고 판단하였다.

다음으로 앞의 숫자가 3자리, 뒤의 숫자가 '000' 인 경우 쉼표가 생략된 것으로 보고 공백을 쉼표로 대체하도록 하였다.(예: 100 000 → 100,000) '000' 이라는 숫자가 단독으로 쓰이는 경우는 사실상 없기 때문에, 이와 같은 형태는 십만 단위의 금액일 것이라고 판단하였다.

그 외의 경우, 즉 앞의 숫자와 뒤의 숫자가 모두 3자리인 경우에는 쉼표로 대체하지 않고 공백을 유지하였다.(예: 900 900 → 900 900) 해당 경우, 물론 실제로 쉼표가 생략되었을 수도 있지만, 영수증 상 10만 단위 이상의 금액은 대부분 총액에 해당한다. 총액은 상품 개별의 가격 인식이 정확하다면 이를 기반으로 후처리에서 재계산이 가능하기 때문에, 큰 수보다는 개별 가격을 위한 작은 수의 정확도에 우선순위를 두었다.

2. 한글 인식 정확도 향상

OCR 인식 결과 숫자와 영문에 대한 결과는 정확도가 높지만, 한글 텍스트에 대해서는 인식 오류가 자주 발생한다. 특히 상품명, 상호명과 같은 고유명사는 띄어쓰기나 문자 자체의 오류로 인해 정확한 보정이 어렵다. 이에 따라 다음과 같은 후처리 방안을 고안하였다.

우선 사용자 참여 기반의 사전을 구축하는 것이다. 기존에 구축된 사전을 기반으로 인식 결과를 검증하고, 사전에 없는 단어에 대해서는 사용자의 확인을 거쳐 오류를 수정하고 그 결과를 새로운 매핑 사전에 저장하는 방식이다. 이 방법을 사용하면, 초반에는 사용자 의존도가 높을 수 있지만 시간이 지날수록 사전에 데이터가 축적되면서 후처리 성능이 점진

적으로 개선될 것을 기대할 수 있다. 그러나 최악의 경우 하나의 단어에 대해 발생할 수 있는 모든 오류를 사전에 수기로 등록해야 하기 때문에 효율성과 확장성에 한계가 있다.

다른 방법으로는 자연어 처리 기법을 적용하는 것이다. 해당 방식을 채택한다면 형태소나 품사를 분석해 특정 단어가 상품명, 상호명, 결제일시, 승인번호 등에서 어디에 속하는지 분류할 수 있을 것으로 기대된다. 그러나 영수증 텍스트는 대부분 문장 구조가 아니라 단일 명사 위주로 구성되어 있기 때문에 일반적인 형태소/품사 분석이 어렵고 효율이 떨어진다. 또, 자연어 처리 기법도 궁극적으로는 사전 기반 학습에 의존하기 때문에 고유명사와 같은 특수 단어들은 인식이 불가능한 경우가 많고, 본 프로젝트에서 사용하는 데이터는 상품명, 상호명과 같은 고유명사가 대부분이기 때문에 적절한 해결책이라고 보기 어렵다.

위 연구를 진행하고 해결책을 적용하면서 추가로 띄어쓰기 복원과 문자 오류 수정 간 상호 의존성에 대한 문제를 발견하였다. 영수증에서는 상품명이나 상호명이 종종 하나의 단어로 붙어서 표기되는 경우가 있는데, 잘못 인식된 문자를 수정하기 위해서는 단어 분리가 선행되어야 하고 반대로 단어를 분리하기 위해서는 인식 오류가 먼저 정정되어야 한다. 따라서 띄어쓰기 복원과 문자 오류 수정을 각각 따로 처리하기보다는, 사전 중심의 통합적 접근이 필요하다고 의견을 모았다. 이를 바탕으로 제시한 세 번째 해결책은 상품명과 상호명 등을 하나의 단어로 처리하는 특수한 사전을 생성하고 영수증 포맷, 상호명, 상품명에 대한 사전을 분리하여 관리하는 것이다.

3. 영수증 포맷 관련 사전을 이용한 오류 수정

기본적인 매커니즘은 OCR 결과의 각 단어와 사전 단어 간의 유사도를 바탕으로 가장 유사도가 높은 항목으로 치환하는 방식이다. 본 프로젝트에서는 유사도에 집중하여, 단어별 유사도를 계산하는 방식에 대해 고찰하였다.

첫 번째로 시도한 것은 Levenshtein Distance 알고리즘을 활용하여 문자 단위로 유사도를 계산하는 것이다. 이 방식은 두 문자열 간의 최소 편집 횟수를 측정하며, 이를 활용해 유사도를 구하는 계산식은 다음과 같다.

$$\text{유사도}_{Levenshtein} = 1 - \frac{distance(s_1, s_2)}{\max(len(s_1), len(s_2))}$$

Levenshtein Distance 알고리즘은 간단하고 계산 속도가 빠르다는 장점이 있으며, 특히 영수증에서 자주 사용되는 4~5자 길이의 단어(예: 승인번호, 거래일시 등)에 대해 한 글자 오류가 발생한 경우 안정적인 결과를 도출할 수 있었다. 그러나 두 글자 이상 오류가 발생한 경우 유사도가 급격히 하락하여 정확한 치환이 어려워졌으며, 이를 위해 유사도 임계값을 낮추면 오히려 다른 단어로 잘못 치환되는 문제가 발생하였다.

따라서 두 번째 시도부터는 단어를 자모 단위로 분해한 뒤 유사도를 계산하는 방식으로 수정하였다. 유사도 계산 알고리즘으로는 Jaro-Winkler Distance 알고리즘과 Levenshtein Ratio 알고리즘을 사용하였는데, 유사도를 구하는 계산식은 다음과 같다.

$$\text{유사도}_{Jaro} = \frac{1}{3} \left(\frac{m}{len(s_1)} + \frac{m}{len(s_2)} + \frac{m-t}{m} \right)$$

m = 위치상 매칭된 문자 수, t = 매칭된 문자 중 순서가 바뀐 쌍의 개수/2

$$\text{유사도}_{Ratio} = \frac{len(s_1) + len(s_2) - distance(s_1, s_2)}{len(s_1) + len(s_2)}$$

자모 단위로 쪼개어 계산하였기 때문에 더 정밀하게 단어의 유사도를 비교할 수 있었다. 특히 Jaro 방식의 경우 짧은 단어에서 더 효과적이었음을 확인할 수 있었다. 그러나 사람에게서는 다른 단어로 인식되어도 두 단어가 Jaro 방식으로 계산하면 유사도가 높아 잘못 치

환되는 경우가 발생하였다. (예: '아임이'와 '승인일시'의 유사도가 0.7)

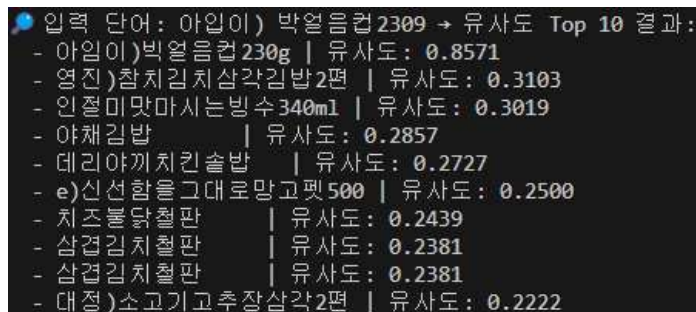
반대로 긴 단어에서 안정적인 방식인 Levenshtein ratio로 유사도를 계산하여 진행한 결과, '금액'과 '금맷'처럼 Jaro 방식을 사용할 때는 유사도가 0.7 이상이었지만 Levenshtein ratio 방식으로는 그 이하의 유사도가 나오는 등 짧은 단어에서의 정밀도가 떨어졌다. 이를 위해 임계값을 낮춘다면 전혀 다른 단어로 치환되는 문제점이 생긴다.

이러한 한계를 극복하기 위해, 짧은 단어에서는 Jaro 방식이, 긴 단어에서는 Levenshtein 기반의 Ratio 방식이 더 효과적이라는 점에 착안하여 단어 길이에 따라 알고리즘을 다르게 적용하는 방식으로 유사도 계산을 정교화하였다. 또한 OCR 과정에서 종종 발생하는 글자 누락 오류를 고려하여, 원 단어보다 치환 단어가 더 짧은 경우에는 일정 비율의 유사도 패널티를 부여하여 잘못된 치환 가능성을 줄이고자 하였다.

이와 같은 방식으로, 사전 기반의 유사도 계산을 통해 영수증 포맷에 해당하는 주요 단어(예: 총액, 승인번호, 결제일시 등)의 오류를 효과적으로 정정할 수 있도록 후처리 로직을 구성하였다.

4. 상호명, 상품명 관련 사전을 이용한 오류 수정

상호명, 상품명은 영수증 내에서 여러 단어가 조합되어 긴 단어로 구성되어 있고 고유명사가 많기 때문에, 영수증 포맷과 달리 오류가 더 많이 발생한다. 따라서 사전 기반의 유사도 보정을 적용하여 유사도 임계값을 완화하여 40으로 설정하여 진행하였다. 해당 기준은 아래 그림과 같이 실험에 의거하여 정한 값으로, 실제로 정답이 아닌 대부분의 단어들과의 유사도는 40% 이하로 도출되었다. 그림의 "아임이) 박얼음컵230g"과 같은 인식 결과에서도 일정 수준의 오류가 존재함에도 불구하고, "아임이)빅얼음컵230g"과 같은 정답으로 무리 없이 매핑되는 것을 확인할 수 있었다.



입력 단어: 아임이) 박얼음컵230g → 유사도 Top 10 결과:

- 아임이)빅얼음컵230g | 유사도: 0.8571
- 영진)참치김치삼각김밥2편 | 유사도: 0.3103
- 인절미맛마시는빙수340ml | 유사도: 0.3019
- 야채김밥 | 유사도: 0.2857
- 데리야끼치킨술밥 | 유사도: 0.2727
- e)신선함을그대로망고펫500 | 유사도: 0.2500
- 치즈불닭철판 | 유사도: 0.2439
- 삼겹김치철판 | 유사도: 0.2381
- 삼겹김치철판 | 유사도: 0.2381
- 대정)소고기고추장삼각2편 | 유사도: 0.2222

[그림 21 인식 결과에 대한 유사도 측정]

5. 상품 정보 추출

상품 정보 추출은 OCR 인식 결과 중에서 상품명과 관련된 정보를 식별하고, 이를 단가, 수량, 총액과 함께 정리하는 과정이다. 기본적으로는 한 줄 내에 "문자 숫자 숫자"의 구조를 갖는 텍스트를 상품 정보로 간주하고, 첫 번째 문자는 상품명, 첫 번째 숫자는 단가, 두 번째 숫자는 총액으로 해석하였다. 이와 같은 해석은 간단하고 대부분의 경우에서 높은 인식률을 보였지만, 몇 가지 한계도 존재하였다. 예를 들어, 영수증의 포맷에 따라 일부 영수증에서는 단가가 명시되지 않고 "상품명 금액"의 형태로만 표기되기도 하며, 상품명에 공백이 포함된 경우에는 구조가 "문자 문자 숫자 숫자"가 되어 인식이 어렵다.

이를 보완하기 위해, 텍스트에서 상품명 사전에 존재하는 단어와의 유사도를 기준으로 상품 정보 줄임을 식별하는 방식으로 개선하였다. 한 줄 내에 여러 개의 문자 조합이 있는 경우에는 처음 문자부터 연속적으로 이어지는 문자 조합에 대해 유사도를 계산하고, 가장

유사도가 높은 조합을 상품명으로 판정하였다. 예를 들어, ‘ICE 카페라떼 6,500 6,500’이라는 줄에서, ‘ICE’, ‘ICE 카페라떼’ 등으로 단어를 점차 조합하며 유사도를 비교하다 ‘ICE 카페라떼 6,500’에서 숫자 형태가 나오므로 조합을 종료하고, 가장 유사도가 높은 ‘ICE 카페라떼’를 상품명으로 확정하는 방식이다.

숫자 형태의 처리 또한 세 가지 경우로 나누어 적용하였다. 숫자가 한 개만 있는 경우는 총액만 명시된 것으로 판단하여 수량은 1, 단가는 총액과 동일하게 설정하였다. 숫자가 두 개인 경우는 첫 번째 숫자를 단가, 두 번째 숫자를 총액으로 판단하고, 수량은 총액을 단가로 나누어 산출하였다. 숫자가 세 개 있는 경우는 순서대로 단가, 수량, 총액으로 간주하여 각 항목에 대응시켰다. 이를 통해 다양한 영수증 포맷에 유연하게 대응하면서 상품 정보를 안정적으로 추출할 수 있도록 설계하였다. 추가로, 제품번호와 같은 숫자 배열은 잘못 인식할 경우 가격 정보에 혼동을 줄 수 있기 때문에, 10자리 이상의 숫자는 추출 대상에서 제외하였다.

나. 정산 기능

■ 웹페이지 UI 구현

정산 기능을 제공하는 웹페이지는 사용자가 서비스를 이용할 때 직관적이고 수기로 입력해야 할 내용을 최소화하는 방향으로 디자인하고 구현하였다. 전체적인 사용자 경험(UX)은 ‘누구나 쉽게 쓸 수 있는 UI’를 목표로 하였으며, 사용자와 영수증 정보를 입력하고 관리하는 과정을 가능한 간단하게 구성하였다. 예를 들어, 사용자는 참여자 목록을 추가하거나 삭제할 때 별도의 복잡한 조작 없이 한두 번의 클릭으로 조작할 수 있고, OCR로 인식할 영수증 이미지 또한 편리하게 불러오거나 수정할 수 있도록 설계하였다. 각 버튼과 입력란에는 라벨이나 플레이스홀더를 통해 어떤 기능을 하거나 어떤 입력을 받는지 등을 사용자가 한 눈에 알 수 있게 하였다.

특히, 항목별 정산과 1/n 정산 방식이 혼재될 수 있는 상황을 고려하여, 구현 과정에서 두 정산 방식을 시각적으로 명확히 구분될 수 있도록 하였다. 각 정산 방식은 별도의 버튼으로 분리되어 있어 사용자가 선택하거나 확인할 때 혼동이 없도록 하였으며, 시각적 강조를 통해 어떤 방식이 적용되고 있는지를 직관적으로 파악할 수 있도록 하였다. 이러한 UI 설계를 통해 정산 과정을 시각화하고 사용자 조작 부담을 줄여, 실질적인 편의성을 향상시켰다.

■ 백엔드 DB 및 서버 구현

OCR 기술을 통해 사용자가 입력한 영수증 이미지에서 텍스트 정보를 추출한 후, 해당 데이터를 효율적으로 저장하고 활용하기 위한 데이터베이스를 설계하고 서버 API를 구현하였다. 전체 DB는 Participant, Receipt, ReceiptInfo, Settlement로 구성되며, 각 테이블에 대한 설명은 다음과 같다.

- 1) Participant : 정산에 참여하는 참가자 정보
- 2) Receipt : 업로드된 영수증 파일 정보
- 3) ReceiptInfo : 영수증에서 추출된 품목, 가격 등 상세 정보
- 4) Settlement : 참가자별 정산 결과 및 내역

이러한 DB 구조를 기반으로 사용자의 입력과 웹 서비스의 처리 과정을 유기적으로 연결하기 위해 각 기능에 맞는 서버 API를 구현하였다.

1. /api/receipt/upload/ : POST

사용자가 메인 페이지에서 영수증 이미지를 첨부하고 PayCheck 버튼을 누르면 /api/receipt/upload/ API를 통해 서버로 POST 요청이 전송되어 영수증 이미지가 서버에 저장된다. 사용자가 두 장 이상의 영수증 내역을 정산할 수 있기 때문에 API 역시 여러 장의 영수증 이미지를 동시에 처리할 수 있도록 설계되었다.

2. /api/participant/join/ : POST

메인 페이지에서 정산에 참여할 참가자 정보를 입력하면 /api/participant/join/ API를 통해 서버에 참가자 명단이 배열 형태로 전송되며, 서버는 이를 개별 Participant 항목으로 분리하여 데이터베이스에 저장한다.

3. /api/receiptinfo/analyze/ : GET

Upload API로 데이터베이스에 저장된 영수증 이미지는 /api/receiptinfo/analyze/ API를 통해 분석된다. 웹 페이지에서 PayCheck 버튼을 클릭하면 서버는 영수증 이미지에 대해 이미지 전처리 - OCR 모델 - 파싱 후처리 단계를 거쳐 상품명, 수량, 단가 등의 정보를 추출한 후, GET 요청을 통해 프론트엔드로 전달한다. 해당 정보는 CheckPage에서 UI 상으로 출력되도록 설계하였으며, 단가는 제외되고 품목명, 수량, 금액, 프론트엔드에서 이 정보를 기반으로 계산한 총 결제 내역이 화면에 보여진다.

4. /api/participant/members/ : GET

/api/participant/members/ API는 데이터베이스에 저장된 참여자 목록을 조회하는 역할을 하며, CheckPage에서 사용자가 항목별 정산 참여자를 입력할 때 유효한 참여자를 검증하는데 사용된다. 항목별 정산에서는 각 영수증 항목 옆에 참여자를 직접 선택할 수 있는데, 이때 MainPage에서 등록한 참여자 목록에 포함된 이름만 추가할 수 있도록 제한을 두었다. 유효하지 않은 이름을 입력할 경우, 붉은 글씨로 오류 메시지를 출력하여 사용자에게 피드백을 제공한다.

■ 정산 로직 구현

/api/settlement/calculate/ API는 사용자가 입력한 정보를 바탕으로 각 참여자 별 정산 금액을 계산하여 반환하는 기능을 수행한다. 1/N 정산의 경우, /api/participant/members/ API를 통해 조회된 전체 참여자를 기준으로 해당 정산에 포함된 모든 영수증의 결제 내역을 균등하게 분배한다. 서버에서는 각 참여자의 이름을 키로 하고, 부담해야 할 금액을 값으로 매핑한 결과를 응답으로 전달한다.

항목별 정산의 경우에는 사용자가 각 항목 옆에 참여자를 추가하거나 삭제할 때마다 해당 API로 요청이 전송되어, 실시간으로 항목별 정산 금액을 계산한다. 하나의 항목에 여러 명의 참여자가 있을 경우, 해당 항목의 금액이 참여자 수만큼 나뉘어 각자의 정산 금액에 더해진다. 응답은 이름을 키로, 계산된 정산 금액을 값으로 하는 형태로 반환되며, 그 결과가 프론트엔드에서 UI 상에 정산 결과로 사용자에게 보여지도록 하였다.

다. 정산 내역 공유 기능

■ 카카오톡 공유 기능

소규모 모임 등과 같은 비공식적인 상황에서 정산 결과를 간편하게 공유할 수 있도록 하는 카카오톡 메시지 공유 기능을 구현하였다. Kakao API를 연동하여 정산 정보를 간단하게 카카오톡 메시지로 공유할 수 있게 한다. 이때 사용자로부터 입력 받은 예금주명과 계좌번호를 포함하여 개인 정산 금액, 송금 요청 내용을 자동 메시지 포맷으로 구성하여 전송하도록

한다. 이 기능을 통해 사용자는 별도의 수기 작성 없이 정산 내용을 간단히 공유할 수 있으며, 메시지를 전달받은 상대는 첨부된 하이퍼링크를 통해 정산 결과 웹페이지에 접속해 상세 정산 내역을 확인할 수 있다. 이러한 공유 기능을 통해 정산 커뮤니케이션의 편의성과 정확성 증대를 기대하였다.

■ 엑셀 파일로 내보내기

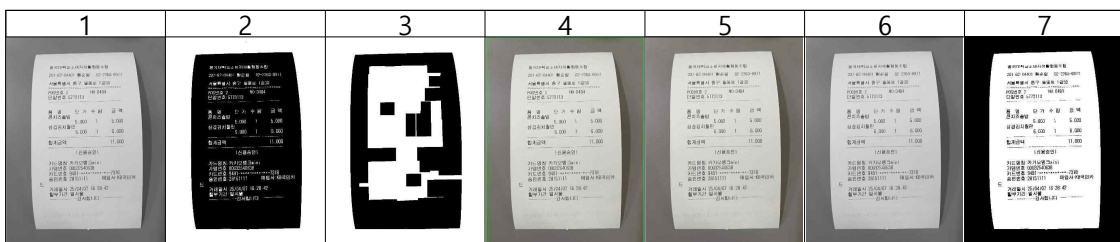
정산 결과를 공식적인 문서로 활용하는 것을 고려하여 엑셀 형식으로 정산 데이터를 내보내는 기능을 구현하였다. 데이터베이스에 저장된 정보를 사전에 정의한 형식에 맞춰 가공 및 정렬하여 엑셀 시트로 변환한다. 정산 결과는 항목별로 적절한 컬럼에 분류되어 출력되며, 사용자가 .xlsx 형식의 파일을 직접 다운로드할 수 있도록 제공한다. 이 기능을 통해 서비스 이용자가 정산 내역을 간단히 회계 자료로 출력하여 제출 혹은 보관할 수 있도록 하였다.

제 4 절 프로젝트의 주요 결과물

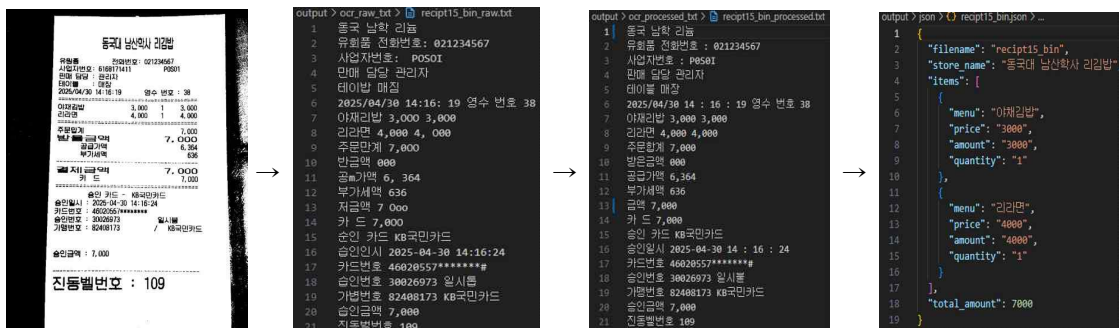
가. 영수증 이미지 인식 기술

■ 이미지 전처리

1. 전체 이미지 → 그레이스케일 변환 : 명암 정보만 추출
2. Otsu Threshold → 마스크(이진화) 생성 : 최적 임계값으로 마스크, 영수증 영역 확보
3. Morphology Closing → 닫힘 연산으로 마스크 구멍 메우기 : 그림자, 작은 노이즈 제거
4. Contour 검출 및 윤곽선 그리기 : 가장 면적이 큰 외곽선을 찾아 영수증 네 개 모서리로 추출
5. Perspective Transform → 컬러 크롭 : 왜곡 보정, 영수증 추출
6. 크롭된 컬러 → 그레이스케일 변환 : 추출된 영수증 기반 보정
7. Otsu Binary + Dilate + Morphology → 최종 이진화 : 텍스트 강조, 노이즈 제거



■ OCR 인식 결과에 대한 후처리



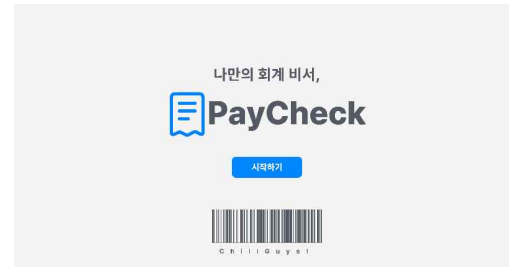
나. 정산 기능

■ 웹페이지 UI

Landing Page
(/)

사용자가 서비스를 이용하고자 할 때 가장 처음 도달하는 페이지.

‘시작하기’ 버튼을 눌러 정산 기능을 이용할 수 있다.



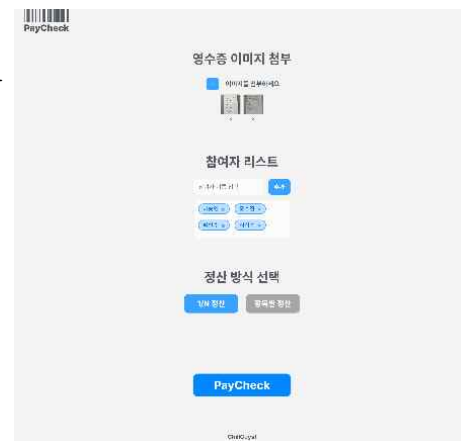
Main Page
(/main)

사용자가 실제로 정산을 위한 데이터를 입력하는 페이지.

상단의 ‘이미지를 첨부하세요’ 왼쪽 버튼을 눌러 정산하고자 하는 영수증 이미지를 업로드한다.

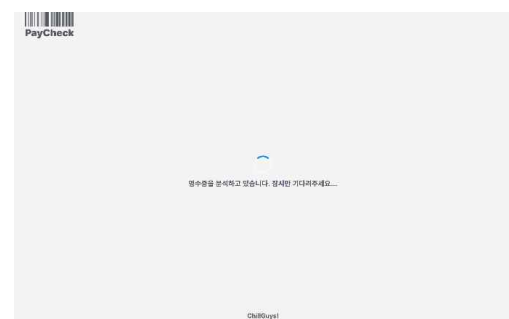
정산에 참여할 인원의 이름을 참여자 리스트 입력란에 작성하고 ‘추가’ 버튼을 눌러 정산 참여자 목록에 추가한다.

하단의 정산 방식 선택 버튼 중 원하는 방식의 버튼을 누른 후 ‘PayCheck’ 버튼을 눌러 정산 결과를 확인할 수 있다.



Result Page
(/check)

업로드한 영수증 이미지의 텍스트 추출이 진행되는 동안 안내 메시지가 화면에 출력된다.



Result Page
(/check)

영수증 분석이 완료되면 상단에는 정산 내역이 간소화되어 출력되고, 하단에는 참여자별 정산 결과와 공유하기 버튼이 출력된다.

정산 방식으로 '1/N' 방식을 선택하였다면, 총액이 참여자 목록에 추가된 모든 인원에게 균등하게 분배된다.

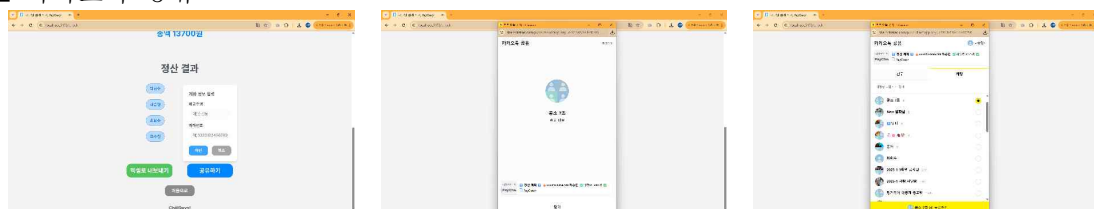
정산 방식으로 '항목별 정산' 방식을 선택하였다면, 영수증 세부 내역에 참여자를 선택할 수 있는 입력란이 추가로 나타난다. 사용자가 항목별로 정산 참여자의 이름을 기입하면, 해당 참여자 수만큼 금액이 나뉘어 참여자의 정산 결과에 합산된다.

만약 참여자 목록에 존재하지 않는 이름을 기입하려고 하면 오류 메시지와 함께 참여자가 추가되지 않는다.

처음부터 정산을 새로 하고싶다면 최하단의 '처음으로' 버튼을 눌러 다시 시작할 수 있다.

다. 공유 및 관리 시스템

■ 카카오톡 공유



■ 엑셀 내보내기

	A	B	C	D	E	F
1	영수증 1					
2	상호명	동국대학교 6차 자생학원 동주합				
3	업로드일	2025-06-12 20:26:04				
4						
5	매뉴얼	수량	단가	총액	참여자	
6	근치조출발	1	5000	5000	박현수	
7	상인입차합판	1	6000	6000	최승연, 최희수	
8						
9	총결제금액		11000			
10						
11	영수증 2					
12	상호명	동국대학교 출판부				
13	업로드일	2025-06-12 20:26:04				
14						
15	매뉴얼	수량	단가	총액	참여자	
16	ICE 카페라테	1	2700	2700	모수진	
17						
18	총결제금액		2700			
19						
20	전체 정산 총액		13700			
21	정산 방식	합계별 정산				
22						
23	정산 결과					
24	참여자	정산 금액				
25	모수진	2700				
26	박현수	5000				
27	최희수	3000				
28	최승연	3000				

제 3 장 추진 일정

제 1 절 추진전략 및 방법

1. 추진 목표

- OCR 기반 영수증 이미지 인식 정확도 향상을 위한 전처리 및 후처리 알고리즘 구현
- 사용자 친화적이고 직관적인 웹 UI 설계 및 구현
- 참여자 및 영수증 정보를 효과적으로 관리하기 위한 DB 설계 및 API 구현
- 다양한 정산 방식과 정산 내역 공유를 지원하는 자동 정산 시스템 구축

2. 추진 전략 및 방법

- 데이터 전처리 및 후처리 정확도 향상: OCR 인식 결과의 오류를 최소화하기 위한 다양한 후처리 기법 적용
- 사용자 편의성 중심의 웹 UI 설계: 직관적인 UI, 정산 과정에서 수기 입력의 최소화
- API 및 DB 설계: 서비스에 사용되는 다양한 데이터를 관리하는 관계형 DB 설계, 프론트 엔드와의 효율적인 데이터 송수신을 위해 각 기능 단위로 분리된 API 구현
- 유연한 정산 처리 방식 적용: 다양한 상황에 대응할 수 있는 정산 기능 설계, 목적에 따라 선택 가능한 공유 기능 구현

제 2 절 추진체계

1. 프로젝트 계획

세부 개발 내용	세부 추진일정(주)											비 고
	1	2	3	4	5	6	7	8	9	10	11	
프로젝트 개발 계획 수립												
오픈소스 OCR 엔진 비교 분석												
이미지 전처리												
OCR 인식 결과에 대한 후처리												
웹페이지 UI 구현												
백엔드 DB 및 서버 구현												
정산 로직 구현												
공유 기능 구현												
프로젝트 개발 마무리 및 고도화 작업												
분기별 진척률(%)	50%						100%					

2. 프로젝트 실행

- 개발 환경: 프로젝트 개발 환경을 통일시키기 위해 사용할 프레임워크와 툴, 라이브러리를 정한다. 원활한 협업을 위하여 깃과 노션, 디스코드 등 커뮤니케이션 툴도 포함한다.
- 역할 분담: 각 팀원별로 분담하여 프로젝트를 진행한다. OCR 기술 고도화 단계와 어플리케이션 개발 단계로 나눈 후, 단계별로 각각 전처리 및 후처리, 프론트엔드 및 백엔드로 역할을 나누어 분담한다.
- 구현: 동시에 개발할 수 있는 기능에 대해서는 분담한 팀별로 진행하며, 상호 간의 협업이 필요한 기능에 대해서는 대면 미팅과 협업 툴을 통해 소통하며 진행한다. 각각 구현한 기능을 단계별로 통합하여 전체 어플리케이션을 완성한다.

3. 품질 관리

- 정확도 및 신뢰성 확보: 기존의 계획에서 전처리 및 후처리 과정을 강화하는 방향으로 수정함으로써 OCR 인식 결과의 정확도를 높인다.
- 검증 및 테스트: OCR 인식 결과를 실제 영수증과 비교하는 과정을 반복하여 정확도를 평가하고 검증한다.
- 오류 관리: 개발 과정에서 발생하는 오류와 그에 대한 해결 방안 모색 과정을 노션에 기록하고 공유해 프로젝트 품질 향상에 기여한다.
- 주기적인 점검 및 피드백: 주기적인 대면 미팅을 통해 문제 상황을 공유하고, 지도 교수님께 피드백을 받아 구현에 반영한다.

4. 프로젝트 모니터링 및 제어

- 진척 상황 관리: 일정 계획과 비교하여 현재 진행 상황을 점검하고 개발 속도와 향후 일정을 조절한다.
- 구현 항목 모니터링: 구현이 끝난 기능에 대해 디버깅과 테스트를 진행하여 오류 발생

여부를 확인하고 코드를 통합한다.

- 커뮤니케이션: 정기 회의를 통해 진행 상황에 대해 공유하고 프로젝트를 동기화한다.

제 4 장 프로젝트의 활용 방안 및 기대효과

제 1 절 기대효과 및 수익성

1. 기술적 측면

영수증 이미지에서 텍스트 정보를 추출하기 위해 Google Tesseract, EasyOCR 등 다양한 오픈소스 OCR 엔진을 적용하고 결과를 비교함으로써 한글 영수증 인식에 최적화된 엔진을 선정하고 실용성을 확보하였다. 또, 이미지 해상도 개선, 노이즈 제거 등 입력된 이미지에 대한 전처리 과정을 추가하여 OCR 기술의 성능 개선 효과를 기대할 수 있다.

후처리 과정에서는 숫자 인식과 한글 인식의 문제점을 각각 분석하여 정확도 향상을 위한 로직을 따로 구현하였는데, 이를 통해 후처리 과정을 포함하지 않거나 하나의 알고리즘만을 적용했을 때와 비교하여 정확도를 향상시키고 나아가 서비스의 신뢰도를 높이하고자 하였다.

2. 경제, 산업적 측면

회계 전문 지식이 부족한 일반 사용자들도 간편하게 정산 및 기록을 할 수 있어, 경조사·스터디·소모임 등 개인 및 소규모 단체(학생회, 동아리, 동호회 등) 사용자들의 일상적 회계 업무의 부담을 크게 줄일 수 있다. 중소기업자의 관점에서는 매일 발생하는 영수증/거래명세서를 쉽게 디지털화하여 소상공인의 회계 업무를 효율적이게 하고, 기존 수기 장부 대신 디지털로 기록함으로써 감사/세무 대비가 가능하다.

또, 회계나 경리 자동화 시장에 진입할 수 있는 기반을 마련할 수 있다. 본 서비스는 API 형태로 모듈화가 가능하기 때문에 기존 ERP나 가계부 앱에 쉽게 통합될 수 있고, 금융 데이터 분석이나 소비 패턴 추적 등 다양한 확장 서비스와도 연결이 가능하다. 장기적으로는 디지털 영수증 인식과 AI 정산 시스템이 결합되어, 무인 점포나 스마트 영수증 기술과 연동하는 차세대 회계 서비스로 진화할 수 있다.

3. 수익성

사전 기반의 텍스트 정제 로직을 모듈화하여 다양한 OCR 기술 엔진 사용자에게 부가 기능으로 제공할 수 있다.

제한된 범위의 영수증에서 정형화된 포맷의 영수증을 대상으로 하는 서비스로 확장해 API로 제공할 수 있다.

제 2 절 활용 방안

가. 동아리 및 소규모 모임

동아리 회비나 식사, 단체 구입 물품 등의 지출을 효율적으로 관리하고, 인원이 여러 번 변동되는 복잡한 상황에도 메뉴별 정산을 자동화할 수 있다. 카카오톡 메시지 공유 기능으로 각종 내역과 계좌 정보를 즉시 전달해 분쟁이나 착오를 예방하며, 필요 시 엑셀 파일로 내보내어 회계 장부처럼 사용할 수 있다. 이 과정을 통해 모든 참여자가 편리하고 투명하게 비용을 부담하여 모임 운영의 만족도를 높일 수 있다.

나. 공공기관·교육기관 및 비영리단체

각 부처나 기관 단위로 발생하는 지출을 영수증 인식으로 빠르게 수집·정산함으로써 행정 업무의 효율을 높일 수 있다. 특히 대학이나 연구실, 비영리단체의 공익 예산 지출도 한눈에 관리하고 투명하게 공개함으로써 신뢰도를 강화할 수 있다. 반응형 웹 서비스와 엑셀 내보내기 기능을 통해 예산 사용 내역을 다수의 사용자가 쉽고 정확하게 확인·보고할 수 있기 때문에 내부 감사 및 외부 검증 시에도 편의성을 제공한다. 부가적으로, 부정 사용 및 중복 예산 집행 예방과 예산 관리의 체계화를 통해 깨끗한 조직 운영에 기여하고 공공 자원의 낭비를 최소화하는 효과도 기대할 수 있다.

다. 학회, 축제 등 대규모 행사 및 프로젝트

여러 부스나 공급업체, 참여자가 모이는 축제·학회 등에서 대량의 영수증을 자동으로 분류·정산할 수 있어 운영 효율이 크게 향상된다. 한꺼번에 많은 거래가 이뤄지는 만큼, 실시간 비용 투명성을 확보해 분쟁과 관리 리스크를 줄이고 의사결정 속도를 높인다. 행사 운영진은 적시에 카카오톡 등을 통해 비용 보고를 공유하여 즉각적인 의사 결정을 할 수 있으며, 마무리 단계에서는 모든 지출 내역을 엑셀로 정리하여 회계 감사 시간을 단축할 수 있다. 이처럼 복잡한 대규모 현장에서도 안정적으로 재정을 통합 관리함으로써 참가자·후원자들에게 신뢰를 주고 행사 성과를 극대화하며 사후 업무도 간소화할 수 있다.

제 5 장 참여인력

제 1 절 업무 분담

NO.	성명	소속			담당분야	참여도(%)
		학과	학번	학년		
1	모수진	컴퓨터공학전공	2022111980	3	프론트엔드, OCR비교분석, 디자인, 문서화	20%
2	박현수	컴퓨터공학전공	2021111933	3	백엔드, OCR비교분석, 파싱 후처리	25%
3	최희수	컴퓨터공학전공	2023112540	3	백엔드, OpenCV, 발표	25%
4	하승연	컴퓨터공학전공	2023111312	3	프론트엔드, OpenCV, 디자인, 발표	30%

[표 4] 참여인력 및 담당분야

제 2 절 팀원 별 수행 성과

모수진

OCR 엔진별 성능 비교를 위한 결과 확인 및 분석

프론트엔드 구현

- 페이지 UI 구상 및 컴포넌트 배치
- Result Page UI 구현
- Kakao API 관리 및 연동
- 카카오톡 공유하기 컴포넌트 구현

문서 작업

- 프로젝트 제안서, 프로젝트 최종 보고서, 기능명세서 작성
- 회의록 작성 및 정리
- 팀 회의 진행 및 관리
- 팀원 간 일정 취합 및 지도 교수님과의 미팅 일정 조율
- Github Organization 생성 및 관리
- 프론트엔드 레포지토리 생성 및 코드 관리

박현수

OCR 인식 정확도 향상을 위한 후처리 알고리즘 담당

- 한글 고유명사(상호명, 품목명 등)의 인식 오류를 교정하기 위한 사전 기반의 유사도 매칭 로직 구현
- Levenshtein Distance, Jaro-Winkler, Levenshtein Ratio 알고리즘을 자모 단위 비교에 적용하여 문자 유사도를 분석
- 단어 길이에 따른 최적의 알고리즘 탐색
- 오인식률 감소 및 정확도 향상을 위한 최적의 임계값 테스트
- 가격 정보의 오인식 문제에 대한 규칙 기반 보정 로직 설계
- 다양한 영수증 포맷에 대응할 수 있도록 조건 분기와 예외 처리 강화

백엔드 OCR 정보 추출 및 파싱 로직 구현

- Django REST Framework 기반의 서버에서 ‘영수증 이미지 업로드 → 전처리 → OCR → 후처리’의 전체 흐름을 구성
- 추출된 텍스트를 구조화하는 과정에서 복잡한 패턴 인식과 사전 기반 검증을 적용
- 프론트엔드에 제공할 수 있는 정제된 데이터 포맷을 반환하는 API를 구축
- 이미지 전처리 단계에서의 출력값을 후처리에 연결
- 인식 결과의 신뢰도 향상을 위한 로직 최적화 기여

최희수 OpenCV 라이브러리를 활용한 모든 이미지 전처리 과정 구현

백엔드 구현

- 서비스에 사용되는 전체 데이터베이스 설계 및 구현
- 정산 API 구현
- 엑셀 내보내기 기능을 위한 API 구현

문서 작업

- 서비스 전체 시퀀스 구상 및 다이어그램 제작
- 프로젝트 제안서 작성 및 발표
- 프로젝트 최종 결과물 발표 자료 제작

하승연 OpenCV 라이브러리를 활용한 모든 이미지 전처리 과정 구현

프론트엔드 및 백엔드 구현

- 전체 페이지 UI 및 구체적인 기능 흐름 디자인
- 서비스 로고 디자인 및 슬로건 제작
- 페이지 UI 구현 및 API 연동
- 엑셀 내보내기 기능, 카카오톡 공유하기 기능 API 개발
- 데이터베이스 초기화 기능 등 백엔드 로직 수정

문서 작업

- 프로젝트 주제 선정을 위한 아이디어 발의 및 기획안 제안
- 아이디어 구체화를 통한 구현 기능 설계
- 팀 일정 및 서류, 개발 과정 관리
- 회의록 작성
- Github 레포지토리 관리 및 최종 코드 통합
- 실행을 위한 가이드 제작
- 프로젝트 제안서, 최종 프로젝트 발표안 작성 및 발표 진행

제 6 장 참고문헌

- 1) 장혁수, 고상호, 이재현, 박승권. (2023). 웹 검색엔진 및 딥러닝 기반 한글 단어 인식 OCR 시스템. 한국통신학회논문지, 48(9), 1169-1174.
- 2) 김정섭, 김현정, 유상현. (2023). 오픈소스 기반 OCR의 한국어 인식성능 비교분석에 관한 연구. 한국정보처리학회 학술대회논문집, 30(2), 471-472.
- 3) 황준상, 전형호, 김상현, 권구락. (2017). 처방전 스캔을 위한 OCR 이미지 인식을 디지털 솔루션. Proceedings of KIIT Conference, 대전.