1. 프로젝트 명 : 키득키득

< 많은 서류의 내용을 정리하는 작장인들을 위한 객체 인식과 OCR을 통해 제공받은 키워드로 서류의 중요 자료를 정리하는 서비스>

2. 팀장 : 고경훈 (JS-B)

팀원 : 김가윤 (JS-B)

팀원 : 김청래 (JS-B)

팀원 : 장 건 (JS-B)

팀원 : 조호현 (JS-B)

팀원 : 하영진 (JS-B)

3. 프로젝트 소개 (상세히 많이 서술)

- 서류 내용에서 키워드 입력시, 원하는 내용을 알 수 있는 프로그램

- 종이 사용율 감소로 환경문제 일부 해결

4. 프로젝트 특장점 (상세히 많이 서술)

- 서류 작업 시, 불필요한 시간 단축 및 비용 절감

- 일렉트론 : 일렉트론은 첫 번째, 낮은 진입장벽을 가지고 있다. 이에 대해 설명하자면, 닷넷과 같은 다른 윈도우 어플리케이션 프레임워크와는 달리, 웹기술에서 요구하는 언어로도 어느정도의 완성도 있는 데스크탑 어플이 구축이 가능하다는 점이 있다. 또, 최근에 각광받고 있는 flutter라는 프레임워크도 electron과 매우 유사하다. 성능은 FLUTTER가 한 참 우위에 있지만, FLUTTER에서 제공되는 기본 UI는 모바일을 베이스로 삼아 제공되기 때문에 Window app이라는 약간 맞지 않는다는 느낌이 있다.

두 번째, 다양한 플랫폼으로의 빌드가 가능하다는 점이 있다. Mac이라거나, 리눅스와 같은 다른 OS로의 빌드를 하기 위해서는, 다른 프레임워크들은 따로 그에 맞는 제약조건과 기능에 맞춰 구성되어야 하는데, 일렉트론은 기본 웹기술로 만든 Front단을 그대로 사용할 수 있다. 이에 맞춰 빌드과정만 조금 손봐준다면 일렉트론이 지원하는 Mac, Linux, deskTop App 3가지의 운영체제에 맞는 플랫폼을 빌드할 수 있다.

세 번째, 일렉트론은 빌드툴을 이용하여 손쉽게 패키징하고 배포할 수 있으며 인스톨러 또 한 손쉽게 만들 수 있다. autoUpdater라는 구성요소를 사용하면 어플리케이션의 버전관리도 수월하게 진행할 수 있다.

네 번째, Node.js에서 제공되는 Api들을 사용할 수 있다는 점이 있습니다. 기본적으로 electron은 Node.js 기반의 백엔드서버를 지니고 있기 때문에, 별도의 설정만 살짝 해준다면 npm과 yarn을 통해 여러가지 노드 기반 API들을 가져와 쓸 수 있습니다.

일렉트론 베이스의 어플로는 슬랙, 디스코드, VSCODE등등이 있습니다. 현재, 해외에서는 일렉트론을 통한 개발이 뜨거운 감자로 떠오르면서 커뮤니티도 활성화되어 있습니다.

- vue.js : vue.js는 사용자 인터페이스(UI)및 단일 어플리케이션을 구축하기 위한 JavaScript 프레임 워크로, 핵심 라이브러리 HTML, CSS, JavaScript 기반으로 작성한다.

첫 번째, vue.js는 뷰파일 하나에 html, css, js를 작성하는데, 쓰여지는 영역( template - HTML영역, script - JS영역, style - CSS영역 ) 이 구분되어 있기 때문에, 한눈에 보기 편하고, 협업하기가 쉽다.

두 번째, 컴포넌트 별로 화면 영역을 구분하여 개발할 수 있기 때문에, 코드의 재사용성이 올라가고 빠른 화면 제작 가능과 유지보수가 용이하게 된다. 화면을 여러개의 작은 단위로 쪼개어 개발 하기 때문에 구현 속도가 빠르다.

세 번째, vue는 리액트와 마찬가지로 Virtual DOM을 사용하여, 화면의 DOM을 추가하거나 삭제하는 등의 변경이 일어날 때마다 화면을 다시 그리는 것이 아니라. 자바스크립트 객체로 DOM의 모양을잡아 놓고 화면의 렌더링 횟수를 최소화하여 브라우저의 부하를 줄이게 된다.

네 번째, vue.js는 vuetify라는 Material Design Framework, 즉 Vue UI 라이브러리를 제공하고 있어, HTML, CSS에 대한 지식이 부족하더라도 MDN을 참고하는 것만으로도 손쉽게 UI를 구축할 수 있다.

마지막으로 vue에는 뷰 디렉티브가 있다. 뷰 디렉티브는 HTML 태그 안에 v-접두사[ex) v-if, v-for, v-on…] 를 가지는 모든 속성들을 의미하는데, 디렉티브로 인해 화면의 요소를 더 쉽게 조작할 수 있다. 뷰의 데이터 값이 변경되었을 때 화면의 요소들이 변경된 데이터 값에 따라 갱신하게 된다. 이런 식으로 화면의 요소를 직접 제어할 필요 없이 뷰의 디렉티브를 활용하게 되면 화면 요소들을 쉽게 조작 할 수 있다.

- 몽고db

-Yolov5를 통해 문서안에 있는 표를 추출.

- Paddle OCR 를 통해 이미지 Data Frame으로 변환

테이블 이미지를 텍스트로 가져와서 개발자로 하여금 자신의 입맛에 맞게끔 코드를 작성할 수 있게 만들 수 있음.

- 키워드를 추출 통해 원하는 데이터를 저장

5.프로젝트 개발내용 (상세히 많이 서술)

- PDF파일을 업로드 시, PDF파일 각각의 페이지를 이미지로 추출

- Yolo v5을 통해 학습 된 모델을 사용해서 각각의 Page 에서 표를 이미지로 추출

- 인공지능을 통해 데이터 프레임으로 도출

위의 과정에서 도출된 테이블 이미지에서 텍스트를 추출하기 위해서 5개의 OCR (- 구글, 네이버, 카카오, 파이테서렉트, paddle)의 후보군을 두고 인식률과 정확도를 기준으로 테스트를 돌린 결과 paddle OCR의 인식률이 90퍼센트 대로 가장 높았고, 산출된 데이터프레임 역시 가장 좋은 결과물을 도출하였기에 선정했음.

- 사용자가 원하는 키워드를 받고 결과 값 도출

OCR 결정 후 반응을 바로 볼 수 있는 코랩에서 진행. 우선 출력되는 데이터프레임을 살펴본 결과 1열과 2열에서 찾기 위한 키워드 값들이 도출됨을 확인. 그걸 기반으로 데이터프레임에서 1열과 2열만 출력해줌으로써 사용자에게 자신이 원하는 키워드를 확인 및 선정할 수 있게 만듬. 그 후 빈 칸에서 nan값이 출력되는데 해당 위치는 윗 열의 문자와 동일하기 때문에 중복문자이므로 하나의 셀로 합친 것이기 때문에 fillna(method=ffill’’)을 사용하여 윗 행의 문자를 가져와서 채워주는 형식으로 작성. 이러한 과정을 거치고도 발생하는 nan값은 텍스트가 아닌 이미지를 읽어와서 작성하는 것이므로 dropna()과정을 통해 결측치를 제거. 본 과정에서 모든 테이블들의 feature 값들이 중구난방하기 때문에 검색에 난해함을 느낌. 그래서 isin을 사용하여 작성. 이 과정에서는 정확한 검색값을 요구하기 때문에 사용자로 하여금 대문자, 소문자, 영문, 한글, 한자, 특수문자 등 입력값을 정확히 해야함을 고지. 이후 테스트에서 정확한 입력값을 넣고 입력했을시 해당 문자 값이 포함된 모든 행을 출력.

6.주요 사용 기술 및 기대효과 (상세히 많이 서술)

- vue.js와 Electron을 통해 프로그램 개발

- Yolo5 모델을 통해 이미지 추출,

- OCR을 통하여 데이터 프레임 생성

본 과정에서 지정한 pdf형식 뿐만 아니라 간단한 코드 개조를 통해 jpg파일 등 다양한 형식의 파일들을 앞의 과정을 생략하고 검색기능을 사용할 수 있음.

- Flask와 Node.js를 활용하여 Web Server 구축

- Node.js는 프로그램 Web Server로 활용

- Flask는 분석 Server로 활용

Flask는 가장 기본적으로 작성할 수 있는 라이브러리로 간단한 과정을 거쳐서 모델 구축 가능

7. 프로젝트 관련 부품 또는 Web, App 화면 이미지 (잘 나온 것으로 최소 10장 이상)

- 별도로 폴더안에 화질 좋은 이미지로 저장해주세요

8.팀사진 (잘 나온 것으로 3장 이상

- 별도로 폴더안에 화질 좋은 이미지로 저장해주세요