2019/01/??

FDX Networks

RPA 와 OCR 인식 연동   
시나리오 테스트

**목 차**

1. 개요
2. 전제 조건
3. 개발 환경
4. 개발 단계
5. 시나리오
6. 각각 시나리오 개요

참고: 테스트 결과

1. **개요**
   * 고객의 OCR 관련 요구를 수용하기 위한 기술 및 방향 적정성 테스트
   * 실제 업무에 적용할 수 있는 세부 기술 분석
   * 이를 기초로 하여 고객 업무에 적용할 수 있는 scenario 개발
   * Scenario별 기본 sample/template 작성
   * 근본이 되는 OCR 기능 sample을 구현한 후 다양한 비즈니스 상황에 사용할 수 있도록 deskew, threshold 등을 적용할 수 있는 토대를 마련
   * 더 나아가, RPA와 연동하여 혹은 독립적으로 서비스가 될 수 있는 가능성을 검토한다

1. **전제 조건**
   * RPA 솔루션과 가급적 연동하는 것을 고려한다
   * 보편적인 고객의 환경에서 사용할 수 있는 기술, 기술의 성능, 그리고 미래 활용 가치를 종합적으로 고려한다.
   * 고객의 환경으로는 인터넷 접속이 안되는 경우도 있는 상황을 고려한다
2. **개발 환경** 
   * 본 기능을 구현하는데 필요한 technology는 다음과 같다
   * OCR 및 Image 처리
     + Google Cloud Vision (p1)
     + Tesseract (p1)
     + OpenCV (p1)
     + Java ImageIO (p2)
     + ABBYY (p3)
     + Datacap (p3)
   * Programming lang
     + Java (p1) (ImageIO 용)
     + Python (p1) (OpenCV용)
   * 기타
     + AA RPA
     + MongoDB (기능적으론 좋으나, 기업에서 활용 여부는 검토 요망)
3. **개발 단계**

* Phase-1:
  + 기본적인 OCR 기능 테스트로, Google Cloud Vision 및 Tesseract OCR 기능의 비교
  + 테스트 결과를 바탕으로, 향후 착수 우선순위 결정
  + 빠른 시일 내에 business에 이용할 수 있는 기초 구축
  + Scenario-1, scenario-2 등의 기능 테스트
* Phase-2:
  + Ph-1의 기능을 확장하여 사용 영역을 확장
  + Scenario-3 등의 기능 테스트
* Phase-3:
  + OCR 영역을 넘어 Image 처리와 RPA를 연계하는 방안 마련
  + Scenario-4 등의 기능 테스트

1. **기능적인 연관**
2. **시나리오**

* Scenario-1:
  + 가장 단순한 scenario로서, 단일 이미지 파일 내 1개 이상의 고정된 영역(들)을 OCR 인식
  + 결과는 clipboard에 복사
  + Scan 위치가 비교적 정확한 경우
  + Scan 위치, 상태가 불량일 경우 보완 대책을 마련
* Scenario-2:
  + Scenario-1을 확장한, 동일 format의 문서를 반복하는 업무
  + 지정된 폴더에 있는 동일 문서 이미지 파일들을 하나씩 영역(들)을 OCR 인식하여 excel에 저장
  + Scenario-1에서 마련된 기술을 기초로 업무에 확장 활용하는 방법

* Scenario-3:
  + 이름: 경우와 같이, “이름”을 지정 후 이의 값을 지정하는 경우 연결 (layout 설계 개념)
  + Layout GUI 필요 검토 (생성, 수정, 삭제 등의 일반적인 기능 등 고려) 혹은 scenario-1을 반복하는 방식으로 해결 가능 검토
  + Layout 정보 (위치, 타입 등) 를 DB에 저장
  + Scan 위치에 많은 보정을 필요로 할 때 보정에 대한 보완 방안
  + Text box가 아닌 grid 형태의 data 기능도 고려
* Scenario-4
  + OCR 인식이 아닌, 이미지 인식 (예: 서명, check 여부 등)
  + 이미지 관련한 다양한 고객의 요구를 수용하기 위한 이미지 처리 기술 습득

1. **각각 시나리오 개요**

* Scenario-1
  + 내용
    - 단일 이미지 파일 내 영역(들)을 OCR 인식
    - 인식 결과는 AA에서 활용할 수 있도록 clipboard에 복사
  + 페이지: 단일
  + 단계
    - 1단계: AA에서 pdf 이미지 파일 로드하여 영역 정의, 영역 ID capturing 수행
    - 2단계: 동일 pdf에서 여러 영역을 지정한 경우 정해진 폴더에 영역별 이미지 파일을 여러 개 생성 🡪 ID를 png 파일 이름에 추가
    - 3단계: 지정된 파일에 대해 Tesseract 혹은 Google Cloud를 이용하여 OCR 인식 🡪 clipboard에 복사. Excel에 Column명이 ID인 column, row에 저장
    - 4단계: 폴더 내 영역별 이미지 파일 삭제
  + 테스트
    - 12, 11, 10, 9, 8 크기 각각의 한글 폰트, 한영, 한영숫자 인식률
    - Paragraph 인식 시 문장 조합 및 인식률
    - Rectangle을 포함한 경우 인식률
    - Scan 위치 및 skew된 경우
    - 이미지 scan 상태
* Scenario-2
  + 내용
    - 지정된 이미지 파일 폴더에 저장되어 있는 각 이미지 파일에서 정해진 영역(들)을 OCR 인식
    - 결과는 excel로 저장 또는 이메일로 담당자에게 전송
  + 페이지: 단일
  + 단계
    - 1단계: 폴더 내 pdf 이미지 파일 로드하여 영역 정의, 영역 ID 부여
    - 2단계: 동일 pdf에서 여러 영역을 지정한 경우 정해진 폴더에 영역별 파일을 여러 개 생성 🡪 ID를 png 파일 이름에 추가
    - 3단계: 폴더를 open
    - 3단계: Tesseract 혹은 Google Cloud를 이용하여 OCR 인식 🡪 Excel에 Column명이 ID인 column, row에 저장
    - 4단계: 폴더 내 영역 파일 삭제
    - 지정된 이미지 파일 폴더에 파일이 있으면 3단계~5단계 반복 수행
  + 테스트
    - 12, 11, 10, 9, 8 크기 각각의 한글 폰트, 한영, 한영숫자 인식률
    - Paragraph 인식 시 문장 조합 및 인식률
    - Rectangle을 포함한 경우 인식률
    - Scan 위치 및 skew된 경우
    - 이미지 scan 상태

**참고: 테스트 결과**

1. Scenario-1 OCR 기능 테스트

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 대상 및 조건 | | 문서 | Font | size | 사각형 | skewed | Scan  상태 | 결과/ 가능성 | 시간  (초) | 비고 |
| tesseract | Built-in | (아래 1) | 한글 |  |  |  |  |  |  |  |
| tesseract | kor |  |  |  | - |  |  | X | > 35 |  |
|  | eng |  |  |  | - |  |  | X |  |  |
|  | kor+eng |  |  |  | - |  |  | X |  |  |
| GCP | - |  |  |  | - |  |  | O | < 4 |  |