

# 문서 정리, 텍스트 화

---

16011033 위진



# Contents

- 서론
- 설계 구성요소 분석
- 개발환경
- 동작 메커니즘
- 시연 및 평가
- 논의



# 01

## 서론

주제 선정이유  
시스템 동작 목표

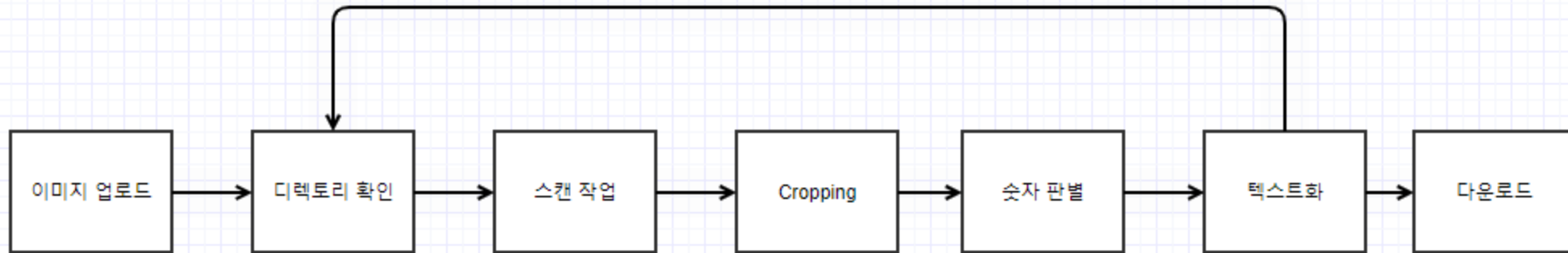
# 서론

## 주제 선정이유

- 여전히 문서를 카메라로 촬영하여 저장하는 경우가 많다.
- 촬영으로 문서를 보관할 때는 가독성이 좋지 않다.
- 촬영을 한 뒤 문서를 보관할 때 다시 인덱싱을 해 줘야 한다.
- 즉, **사용자의 편의성을 향상시키기 위해서 필요하다.**

# 서론

## 시스템 동작 목표



# 02

## 설계 구성요소

목표 동작 구현방법 구상

제한조건

개발 계획표

# 설계 구성요소

목표 동작 구현방법 구상

- 이미지가 웹을 통해서 들어온다.
- 스캔 작업을 한다.
- 스캔 된 이미지의 좌측 상단 부분을 잘라 낸다.
- MNIST로 학습 된 모델을 통해 해당 숫자가 무엇인 지 판단한다.
- 스캔 된 이미지를 텍스트 화 해서 저장한다.
- 인덱싱에 맞게 제목을 바꾼다.
- 사용자에게 제공한다.

# 설계 구성요소

## 제한사항 정의

- 라즈베리 파이 4를 사용하므로 GPU를 이용하지 않는다.
- 연산량을 줄이기 위해서 위치를 좌측 상단으로 고정하고, 영상을 처리하지 않는다.
- 모델의 학습은 외부 컴퓨터에서 진행합니다.



# 설계 구성요소

## 프로젝트 진행 계획

일정	개발내용	개발 기간		
		6/4~6/11	6/12~6/17	6/18~6/22
1.필수 기능 구현	스캔 기능, 숫자 인식 기능, 텍스트화 기능			
2.서버 필수 기능 구현	촬영 버튼 기능, 다운로드 기능			
3.서버 추가 기능 구현	알림 기능 등(여유가 된다면)			

# 03

## 개발환경

# 개발환경

사용한 S/W

- tensorflow 1.14
- keras 2.3.1
- pytesseract-OCR
- numpy
- cv2
- imutils
- matplotlib
- skimage
- PIL
- argparse



# 04

## 동작 매커니즘

탑승자 파악 메커니즘

요금계산 메커니즘

--예)동작/시연 시나리오별 설명--

# 동작 매커니즘

## 스캔 매커니즘

- 이미지의 선으로 추정되는 부분만을 추출해 낸다.
- 추출해 낸 이미지에서 사각형을 이루는 4개의 점을 찾는다.
- 해당 사각형을 4개의 점을 이용하여 직사각형 형태로 재배열 한다.

# 동작 매커니즘

## MNIST 매커니즘

- keras 라이브러리에서 mnist dataset을 가져온다.
- mnist 데이터 셋을 CNN을 이용하여 학습시킨다.
- 학습 시킨 모델을 라즈베리 파이에 저장한다.
- cropping 된 사용자의 이미지의 크기와 색을 적절히 변환한 후 모델에 집어넣고 결과를 리턴한다.

# 동작 매커니즘

이미지 텍스트화

- 정확도를 높이기 위해서 스캔이 된 이미지를 사용한다.
- 커맨드 명령을 이용해 tesseract를 사용하여 이미지를 처리한다.

# 동작 매커니즘

웹 (실패)

- 서버와 클라이언트 간의 파일 전송을 위해서 ssh를 이용하려고 하였다.
- 프로그램의 동작 시점과 완료 시점을 웹이 확인하기 위하여 자바 스크립트를 이용하려 하였으나 브라우저의 보안 상의 문제로 실패하였다.

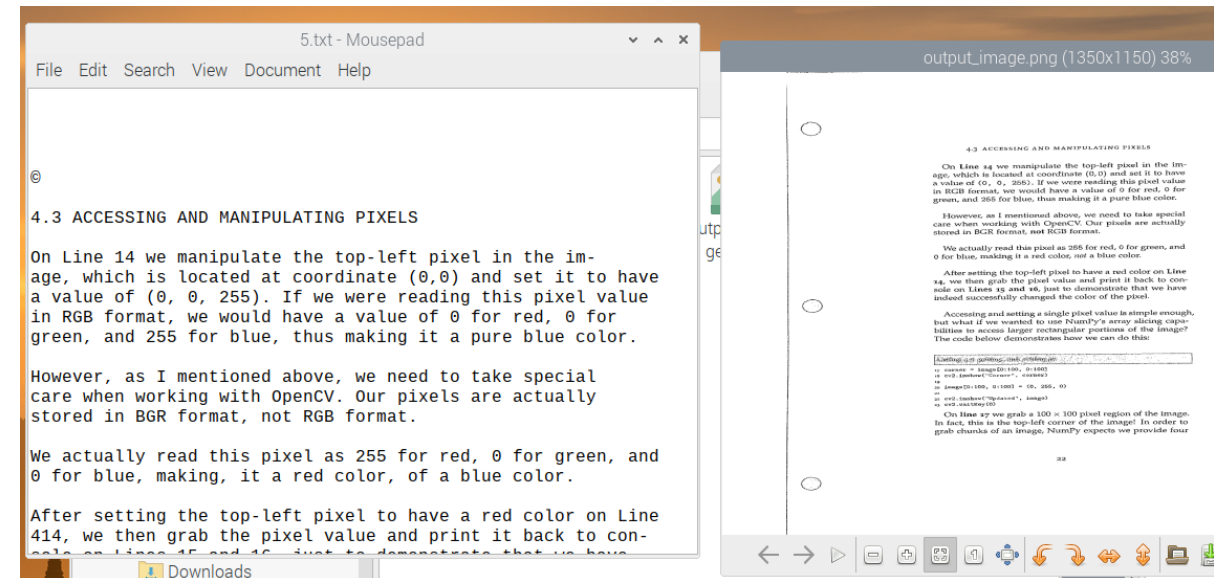
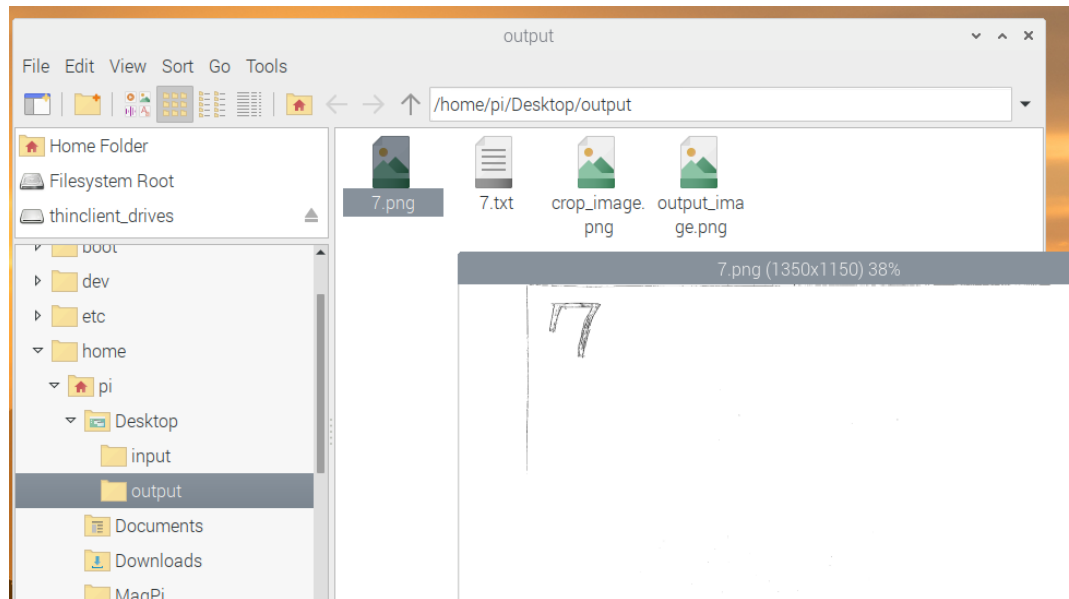


# 05

## 시연 및 평가

# 시연

## 구현 내용 시연



# 평가

## 구현 내용 평가

- 스캔, 이미지 텍스트화, 문서 정리는 구현이 되었다.
- 문서 정리 기능의 모델 자체는 정확도가 95% 이상이다.
- 그러나, 사진을 변형하는 과정으로 인해서 정확도가 낮아진다.
- 순차적으로 많은 일을 하기 때문에 처리 속도가 30초를 넘는다.
- 스캔 과정 실패, input폴더 비어 있는 경우 등, 예외 처리 부분이 부족하다.
- 웹 구현에 실패하였다.

# 06

## 논의

개인 소감  
요약

# 논의

개인 소감

## 위진

이미 구현된 여러 모델을 사용했기에 구현 난이도가 높지 않다고 예상하였으나, 라즈베리 파이에 그 프로그램을 올리는 과정에서 버전이 달라서 안된다거나, 없는 소프트웨어라거나, 운영체제에 가까운 함수라서 사용이 불가능한 등의 문제를 체험했습니다. 또한, 웹을 구현하려 했던 과정에서 라즈베리 파이와의 개인 PC의 통신, 서버 구축 관련해서도 많은 고민을 했습니다. 실제 임베디드 시스템의 환경 설정이 얼마나 어려울 지 생각해 보게 되는 계기가 되었습니다.

# 논의

## 요약

- 웹 구현 목표는 실패하였고,  
스캔 기능, 문서 정리 기능, 텍스트화 기능 등 주요 기능은 구현 완료 하였습니다.
- 그러나 예외 처리 부분에서 부족하며,  
실행 속도가 30초에 근접하는, 매우 느린 속도를 보여줍니다.  
그러나 개인적으로 주제의 특성 상 어느 정도 감안할 수 있는 부분이라고 판단합니다.

Thank You !

