코웨이

coding
wherever
easily

소프트웨어공학부

20132889 신종혁 20123043 박세용

20123087 이성준 20142773 최인정

목차

1. 프로젝트 소개

프로젝트 개요

손으로 작성된 수도 코드를 Text 형태의 완성된 Java 코드로 변환하는 어플리케이션

Pseudo Code

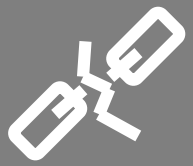
Text File
(Java)

프로젝트 개요

- **목표**

손으로 작성된 간단한 수도 코드를 사진으로 찍어 서버로 전송 후 Java 문법이 적용된 text로 변환해주는 어플리케이션 개발

아이디어 저장



코딩의 시 공간적 제약 탈피

로직에 집중

진입장벽 해소

1차 발표 피드백

Q.
1. 아이디어는 좋으나, 결과 예측에 대한 목표가 불명확합니다. 최종 평가 시 기준이 될 프로젝트의 정확도를 정량적으로 제시하세요. 2. 텍스트 추출 부분의 정확도는 어느 정도입니까?
A.
1. 80%이상의 정확도와 5초 이내의 처리 시간 목표 - 20% 오류 : 영상처리 오류 + 변환 오류로 가정 2. 현재 진행 상황으로는 정확도를 명확히 하기 어려움.

1차 발표 피드백

Q.
3. PSEUDO CODE의 형식이 배우기 쉬운지? 4. 순서도 그림 형태의 pseudo code도 처리 가능한지 궁금합니다.
A.
3. 일반적인 상위 언어에 비해 제약이 적어 간단. -변환 가능한 형태의 수도 코드를 사용자에게 안내하기 위한 튜토리얼 제공 예정 4. 순서도 형태의 pseudo code는 허용 범위에서 제외.

1차 발표 피드백

Q.

5. 전체 프로젝트의 목표는 pseudo code > Java code인데 영상 처리를 강조하는 이유가 무엇인가요?

6. Code Lib 가 Pseudocode mapping 으로 정확하게 mapping 이 될 수 있을지 기능성이 모호합니다.

A.

5. 손 글씨로 작성된 수도 코드를 변환하는 것을 프로젝트 목표로 설정

- 따라서 폰트가 적용되지 않은 텍스트에 대한 정확한 영상처리가 기본적으로 전제 되어야 높은 변환 정확도가 보장됨.

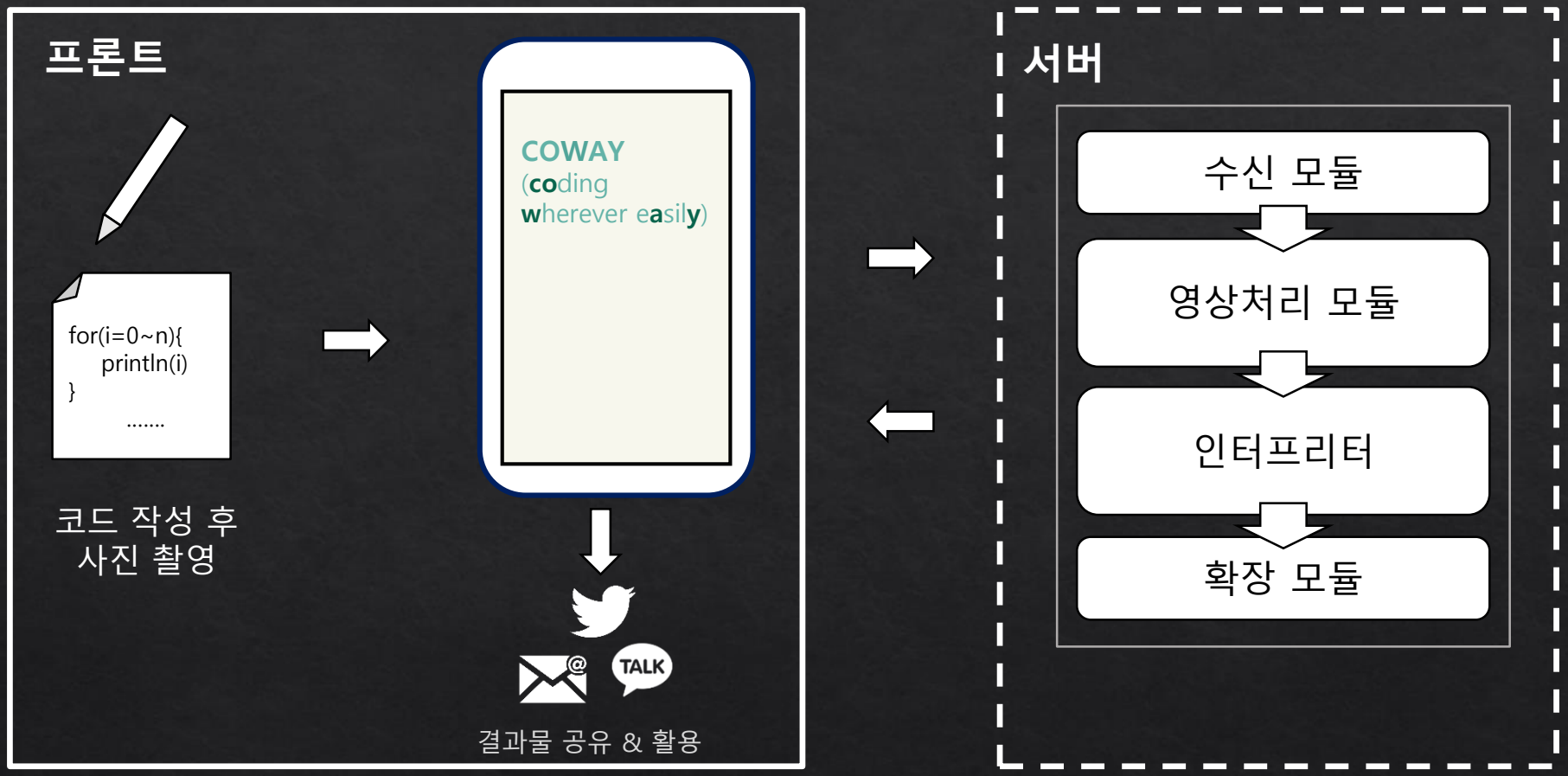
6. 외부 Lib import를 제외하였기 때문에 상당 부분 가능하다고 판단.

1차 발표 피드백

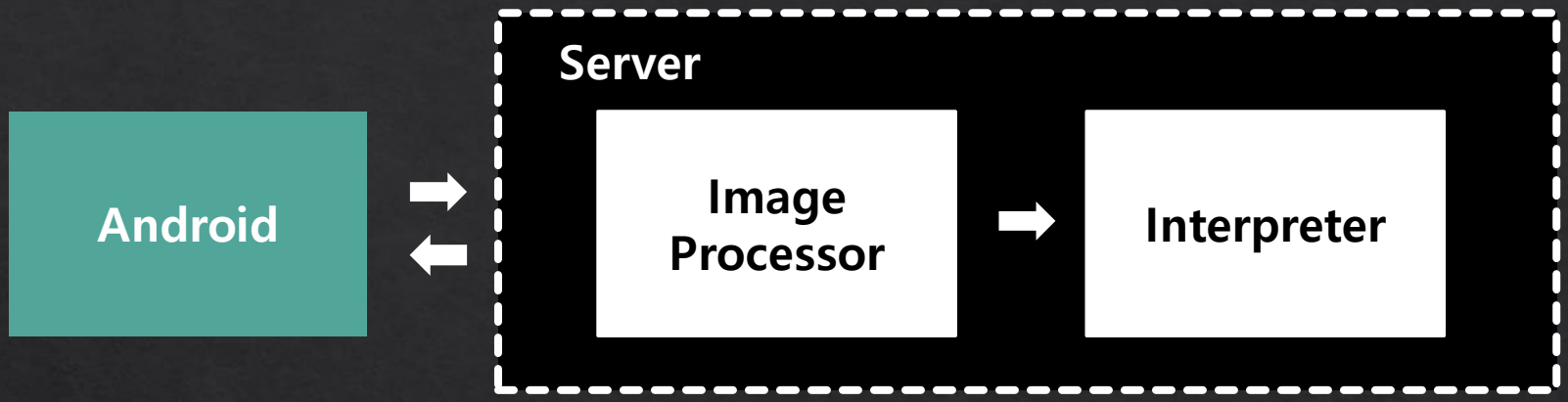
Q.
7. 어느 정도로 복잡한 pseudo code 를 다룰 수 있나요? 8. 어떤 알고리즘을 이용해서 pseudo code > Java code를 변환할 예정인가요?
A.
7. 처리 가능한 수도 코드에 대한 명시 - 한 줄에 하나의 문법 허용 - 영어 표현식이 아닌 기호 기반의 표현식 - 외부 라이브러리 기능 제외 8. 인터프리터 시연 동영상을 통해 알려드리겠습니다

2. 수행 내용 및 중간결과

Workflow


프론트
for(i=0~n){
println(i)
}
.....
코드 작성 후
사진 촬영
COWAY
(coding
wherever easily)
TALK
결과물 공유 & 활용
서버
수신 모듈
영상처리 모듈
인터프리터
확장 모듈

중간 결과



Android

- 프로세스에 따른 화면 구성 완료
- 갤러리 사진 연동 / 카메라 / CROP 기능 추가
- 사진 전송 및 결과 수신을 위한 비동기 통신 구현
- 변환 결과에 따른 화면 창 분리 완료
- 서버로 전송하는 이미지 용량 축소 시도 중

중간 결과

Server

- Socket 통신 구현 완료
- FE(Android) – BE(Server) 간 통신 프로토콜 정립
- 변환 결과에 따른 통신 시퀀스 구현 완료
- 서버 내부 모듈 호출 구현 중

중간 결과

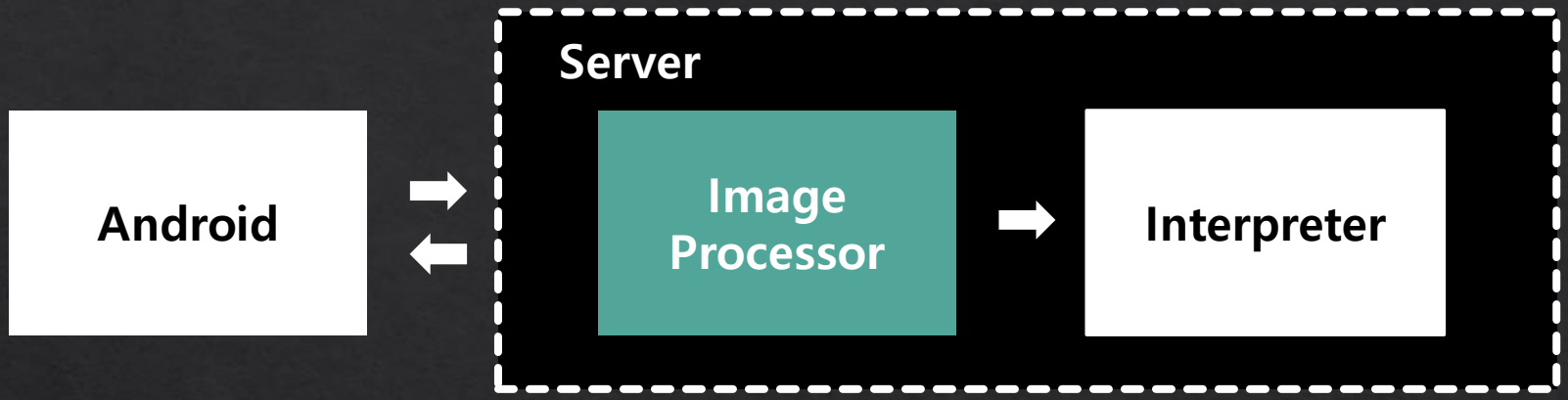
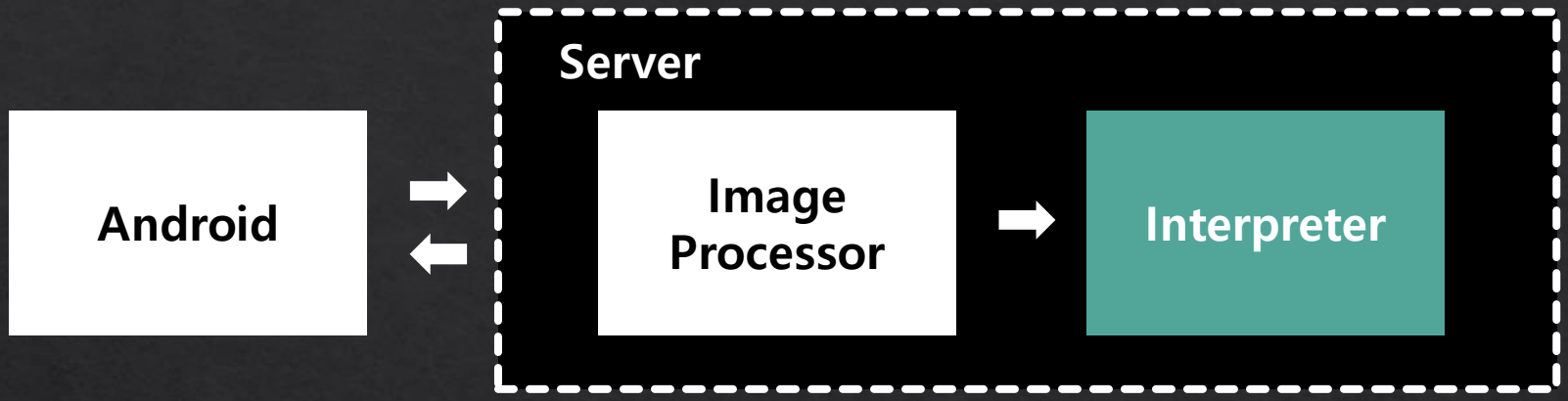


Image Processor

- 이미지 명암 제거 및 바이너리 이미지 생성
- Text 라인 별 인식
- 각 캐릭터 윤곽선 추출, 정렬 및 통합
- 폰트 텍스트 인식 및 추출
- 실행 스크립트 생성
- Vision API를 활용한 손글씨 인식 진행 중

중간 결과



Interpreter

- For문 변환 규칙 구현
- print/ println 변환 규칙 구현
- 실행 스크립트 작성
- 변수 추론 변환 규칙 구현 진행 중
- 세미콜론 붙이는 경우의 수 연구 진행 중
- 변환된 코드 인덴트 정렬 기능 진행 중

3. 수정된 연구내용 및 향후 추진 계획

수정 사항

Image Processing

- 텍스트 추출 방법의 변경

기존 계획 : 테저렉트를 활용한 영상처리

BUT 수기로 작성된 이미지에서는 작동하지 않아 텍스트 추출이 불가능.

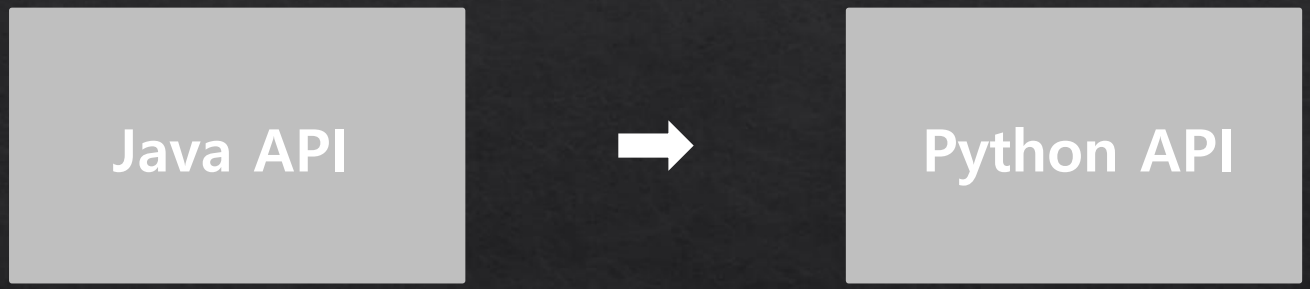
⬇

구글 비전 api를 통한 영상처리 시도 중

수정 사항

Server

- 영상처리 모듈 구동 API 변경



Python Script를 호출하는 메서드 구현 필요

수정 사항

Interpreter

- 변수 추론 기능이 새롭게 추가 됨

  ➡ 핵심 기능 위주로 빠르게 구현 후 보완

- 인식 가능한 예약어(수도코드)의 범위 산정이 복잡

  ➡ 외부 수도코드 분석을 통해 가장 대중적인 예약어를 선정

- 모듈 간 프로토콜 통일 필요

  ➡ commons 라이브러리를 생성 (I/O, DTO packet)

향후 추진 계획

Image Processing

- Google Vision api를 활용한 텍스트 인식
- 원본 형태를 유지한 텍스트 추출

Server

-FE와의 통신 고도화

-영상처리 모듈 호출 및 인터프리터 호출 부분 최종 작성

향후 추진 계획

Interpreter

- 변수 타입 추론 기능 완료
- 인식 가능한 수도 코드 확장
- 변환 실패 경우의 수 파악 및 보완

Android

- 변환 성공 시 자바 코드 저장 및 전송 기능 추가
- 사용자 튜토리얼 제작
- 통신 오류 제어 및 디자인 보완

향후 추진 계획

Analysis & Test

- 외부 수도 코드 수집 및 분석
- Logical error 유발하는 경우의 수 분석
- 수기 데이터 수집 및 제작
- 수기 데이터 별 오류 케이스 분류

Q&A

감사합니다