

Team 11

SignLanguage.txt

- 프로젝트 계획서

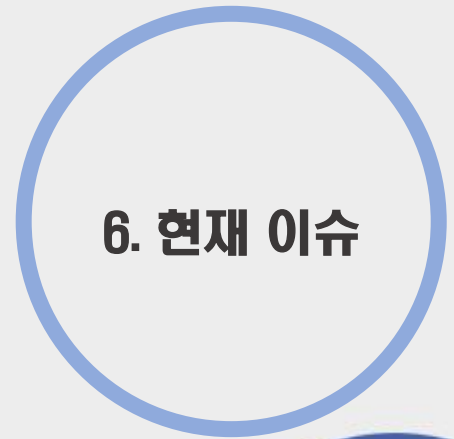
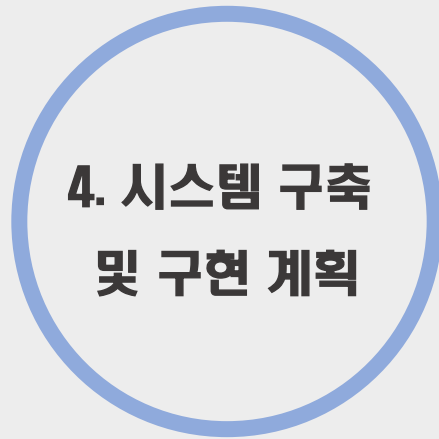
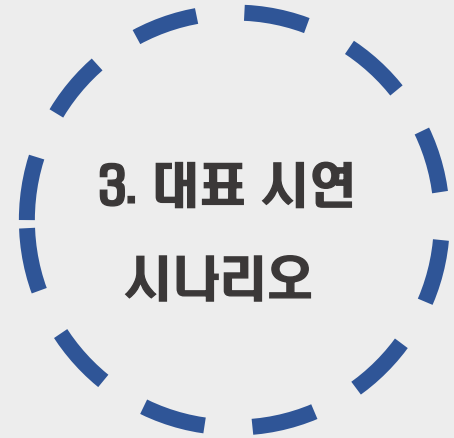
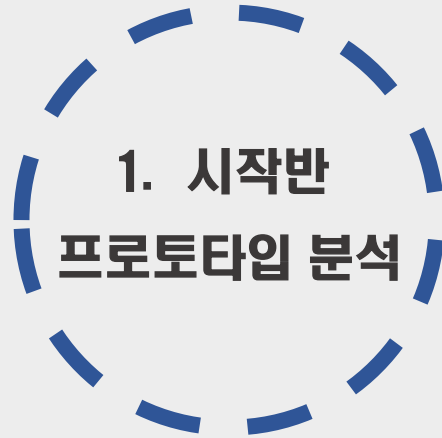
졸업프로젝트2
팀 11

201815004 이찬민


201611205 박승민

201814119 문지영

목차



1. 시작반 프로토타입 분석



프로토타입
리스크 및
해결 방안

- 성능 리스크 : “얼마나 수어 인식을 잘 할 수 있는가?”
➡ 학습된 모델로부터 모든 팀원 테스트
- 범용성 리스크: “임의의 사용자가 써도 차이가 없는가?”
➡ 타인에게 부탁하여 시연

1. 시작반 프로토타입 분석

수어 촬영시의
가이드라인 제시

- 상반신이 다 나오게 일어서도록 함
- 손이 앵글밖에 나가지 않게 관리



데이터에 따른
딥러닝 횟수 및
단어 수 제한

- 상정한 시나리오 내에서의 단어 선택으로 최소화
- 선택을 최소화 한만큼 한단어의 데이터 개수를 늘리는 방안
- 2000번 이내의 반복학습

2. 변경(개선)사항

1) 청각장애인의 수어 속도 제약

- 평소 수어 속도로 하게 된다면 너무 빨라 keypoints들이 제대로 그려지지 않을 수 있음

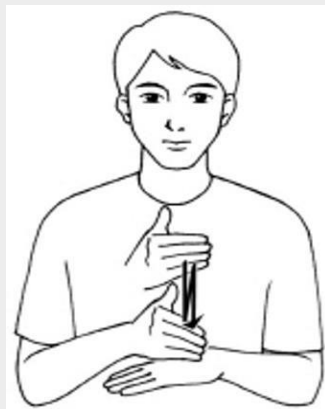
2) 특정 경우에 한하여 재시도 횟수 증가

- 비슷한 수어는 인식에 어려움이 있을 수 있음
- 따라서 재시도 횟수 최대 3회 -> 최대 5회

3) 자음과 모음 일단 배제

- 인식 속도와 사용성을 고려했을 때, 자모로 단어 표현은 너무 오래 걸림
- 따라서 자주 사용하는 단어를 최대한 선별하는 방향으로 진행할 예정

Ex)



감사합니다



미안합니다

3 대표 시연 시나리오

- 기존의 병원 시나리오를 **채팅 형식**으로 시연
- 오른쪽(초록색)이 의사이고 왼쪽(회색)이 청각장애인인 환자



"안녕하세요" 수어 동작 시



아픈 부위 가리키며 "아프다" 동작 시

4. 시스템 구축 및 구현 계획

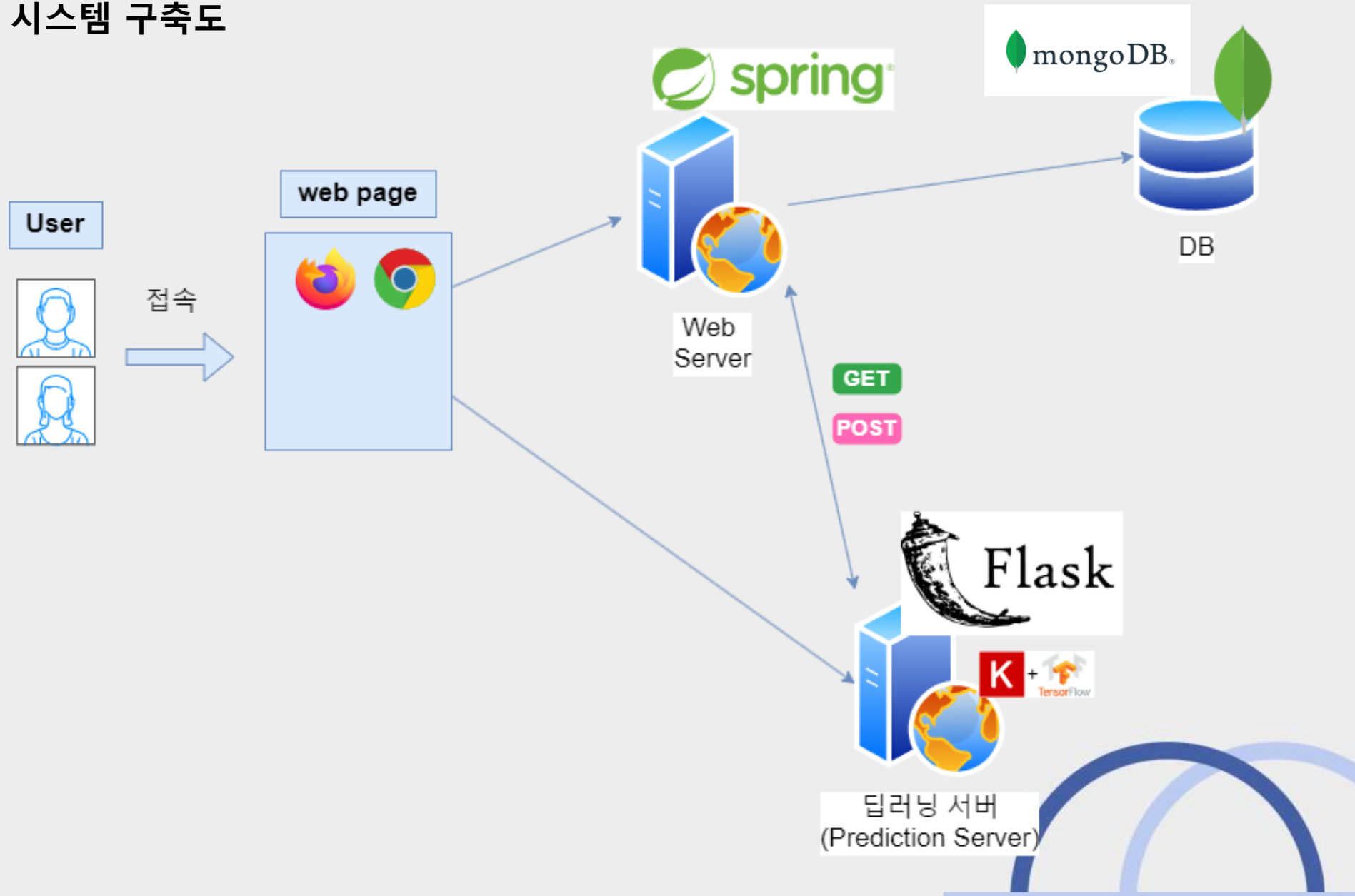
개발 환경 구축 계획

- 운영 체제: Windows 10
- 형상관리 소프트웨어: Git
- 웹서버: Spring Boot(웹과 채팅서버)
- **Web Application Server**: Tomcat(초기엔 내장 톰캣을 이용할 계획)
- **딥러닝 서버**: 화면에 웹캠 띄우기 및 모델을 통한 예측, 결과값 웹서버로 전송을
Flask 프레임워크를 이용하여 수행할 계획
- 데이터베이스: MongoDB
- 테스트 도구: Spring의 Junit
- **IDE**: IntelliJ, VSCode



4. 시스템 구축 및 구현 계획

- 시스템 구축도



4. 시스템 구현 계획

화면

- 메인 화면 – Bootstrap을 이용하여 구현
- 병원 페이지 화면 - html과 css를 이용하여 구현

채팅

- 채팅방 생성 – RoomController에서 HTTP POST를 통해 생성 가능하도록 구현
- 채팅방 입장 - RoomController에서 HTTP GET을 통해 목록을 받아와 화면에 출력하도록 구현
- 채팅 – 채팅방 입장 시 채팅이 가능하도록 구현
- 클라이언트 간의 통신 – STOMP를 이용하여 다수의 채팅방 생성 및 채팅이 가능하도록 구현하고 StompChatController를 통해 메시지 전달이 이루어지도록 구현
- 예측 결과 출력 – 딥러닝 서버로부터 받아온 결과를 출력하도록 구현



4. 시스템 구현 계획

웹캠

- 영상 촬영 - Spring에서 웹캠의 내용을 blob로 만든 후 avi 확장자로 변환하도록 구현
- 영상 전송 - 딥 러닝 서버에 HTTP Post를 통해 전송하도록 구현

예측

- 영상 획득 - Spring으로부터 avi확장자의 비디오를 획득하도록 구현
- 영상 예측 - 미리 학습된 모델을 통해 받아온 비디오로부터 예측결과를 생성하도록 구현
- 웹서버로 결과 전송 - HTTP 메소드를 이용하여 Spring 웹서버에 전송하도록 구현

DB

- 채팅방 이름과 목록 저장



5. 시험 계획

Testcase	#	조건	기대 결과	Usecase
수어 동작 하기	1	수어 동작을 한다.	수어 정보를 학습된 모델로 전송한다.	수어 입력
텍스트 입력하기	2	채팅에서 텍스트를 입력한다.	입력된 텍스트를 서로의 화면에 표시한다.	텍스트로 입력
버튼 선택하기	3	자주 사용하는 표현은 버튼을 누른다.	선택된 버튼의 텍스트를 서로의 화면에 표시한다.	텍스트로 입력
수어 학습 모델 정확도 확인하기	4	충분한 양의 데이터를 학습시킨다.	정확도가 90%이상이고 categorical_crossentropy의 손실함수가 정상적으로 그려졌는지 확인한다.	
학습된 모델 성능 확인하기	5	학습시킨 단어의 수어 동작을 입력한다.	정확한 수어 예측값이 나온다.	
	6	동작이 유사한 단어의 수어를 입력한다.	정확한 수어를 예측 시간이 2초 이내이다.	
	7	제3자의 수어 동작을 입력한다.	정확한 수어 예측값이 나온다.	
수어를 텍스트로 번역하기	8	사용자로부터 학습된 수어 데이터를 받는다.	학습된 모델로 예측을 하고 예측된 결과를 웹서버로 전송한다.	수어를 텍스트로 번역
	9	사용자로부터 학습되지 않은 수어 데이터를 받는다.	사용자에게 수어 동작을 다시 요구한다.	
번역한 텍스트 출력하기	10	수어 예측값을 웹서버에 전송한다.	서로의 화면에 예측값을 텍스트로 보여준다.	번역한 텍스트 출력

6. 현재 이슈

1. 사용자의 웹캠 영상을 예측 서버로 전송하는 방법
 - 영상을 서버로 전송하는 것은 큰 문제가 없어 보이나 실시간의 경우에는 웹캠 공유 문제로 인해 힘들어 보임
 - WebRTC를 이용하는 방안을 쓰면 웹캠 공유 문제는 해결 되지만 RTT가 높게 나와 영상이 너무 느리게 재생되는 단점이 보임
2. 예측 서버로부터 받아온 결과를 웹 화면에서 보여주지 못함
 - 아직 예측서버에서 결과를 웹서버로 POST해주는 기능을 구현을 못하고 있어 결과를 웹 화면에 보여주지 못함
3. DB 로컬 연결
 - 현재 로컬 DB와 연동이 가능하지만 외부에서 DB를 접속하지 못함



팀원 역할 분담



이찬민(리더)

- 모델학습을 위한 데이터 생성
- 모델 생성 및 학습
- 모델 테스트
- 웹페이지 생성
- 모델 비교
- 웹서버, 딥러닝 서버 구축



박승민

- 모델학습을 위한 데이터 생성
- 모델 테스트
- 모델 비교
- 웹페이지 구성



문지영

- 모델학습을 위한 데이터 생성
- 시나리오 생성
- 모델 테스트
- 채팅방 DB 관리
- 수어 예측 결과 DB 관리



일정표

	프로젝트 진행 계획	보고서 제출 일정
3월	- 수어를 글로 번역하는 주제 선정	- 프로젝트 신청서 제출
4월	- 특정 목적(예, 병원)을 위한 수어 데이터를 수집 - 사용할 오픈소스를 탐색	- 요구사항 분석 - 요구사항 분석서 작성 - 시스템 설계
5월	- 딥러닝을 이용한 영상 처리	- 시스템 설계서 작성 - 프로토타입 설계 및 구현 - 프로토타입 테스트
6월	- 딥러닝을 이용한 영상 처리	- 프로토타입 구현결과 발표 - 기말보고서 작성
7월	- 웹페이지 및 서버 개발 - 인식속도 및 인식률 개선	
8월	- 웹페이지 및 서버 개발	
	-	
9월	- 웹 서버에 OpenCV를 올리기 - 웹페이지에 채팅 및 버튼 구현	- 프로젝트 계획서 - 시스템 구조 설계서 수정
10월	- 추가 시나리오를 위한 단어로 모델 학습	- 풀 스케일 시스템 구축
11월	- 디버깅 - 최종 완성 및 시연	- 시험 계획 수립 및 시험 시행 - 시험 결과 분석을 통한 디버깅 및 성능/기능 최종 개선
12월	- 코드 리팩토링	- 최종발표

감사합니다.

