

얼굴인식을 통한 자동로그인이 가능한 채용 웹사이트 개발

최효현*, 조민호^o

*인하공업전문대학교 컴퓨터정보공학과,

^o인하공업전문대학교 컴퓨터정보공학과

e-mail: hchoi@inhac.ac.kr, chominho14@naver.com

Development of Facial Recognition-enabled Automatic Login for Recruitment Websites

Hyo Hyun Choi*, Min-Ho Cho^o

*Dept. of Computer Science, Inha Technical College,

^oDept. of Computer Science, Inha Technical College

● 요약 ●

본 논문에서는 해당 웹 사이트에 얼굴 인식을 통한 자동로그인 시스템 서비스를 구현한다. 얼굴 인식 라이브러리로 face_recognition을 사용한다. 웹 사이트에 접근 시 원하는 채용공고에 지원할 수 있으며, 원하는 기업을 검색하고 저장하여 모아 볼 수 있는 서비스를 제공하는 웹 애플리케이션을 설계하고 구현한다. React를 사용하여 프론트엔드를 구성하고 SpringBoot와 Flask를 사용하여 백엔드를 구현하였다. 자동로그인을 위한 얼굴 인식 라이브러리로 face_recognition을 사용한다.

키워드: 얼굴인식(Facial recognition), 자동로그인(Automatic login), 채용 시스템(Recruitment system)

I. Introduction

컴퓨터와 인터넷의 진보로 인해 쇼핑, 예약, 편지 등 일상의 다양한 활동을 인터넷을 통해 간편하게 처리할 수 있게 되었다. 이러한 서비스를 이용하기 위해서는 웹 사이트에 회원가입하고 로그인해야 한다. 그러나 각 웹 사이트마다 별도의 아이디와 패스워드를 기억해야 하는 번거로움이 있다. 이에 따라 다수의 웹 사이트가 등장하면서 사용자는 여러 계정의 아이디와 패스워드를 기억해야 하는 어려움에 직면하게 되었다. 이러한 문제를 해결하기 위해 소셜커머스 로그인과 같은 방법도 등장했지만, 로그인 정보를 기억해야 한다. 본 논문에서는 사용자의 편의성을 향상하기 위해, 얼굴 인식을 활용하여 로그인을 간소화하는 시스템을 기획하고 개발한다.

face_recognition은 얼굴 인식 라이브러리로, 딥러닝 기반으로 제작된 dlib의 얼굴 인식 기능을 사용하여 구축되었다. python 또는 CLI에서 얼굴을 인식하고 조작할 수 있다.[2]

III. The Proposed Scheme

1. Recruitment Services

본 논문에서는 React를 사용하여 fig 1과 같이 일반 회원에게 회원가입, 로그인, 채용공고 검색 및 지원 기능을 서비스한다. 기업 회원은 채용공고 생성, 수정, 삭제 기능을 서비스한다.

II. Preliminaries

1. Related works

springboot는 JVM에서 실행되는 독립형 애플리케이션을 만들기 위한 오픈소스 프레임워크인 Spring을 사용하여 빠르고 쉽게 애플리케이션과 마이크로서비스를 개발하도록 돕는 도구이다. springboot는 Tomcat과 같은 서버를 직접 포함하고 spring 및 타사 라이브러리를 자동으로 구성한다는 특징이 있다.[1]

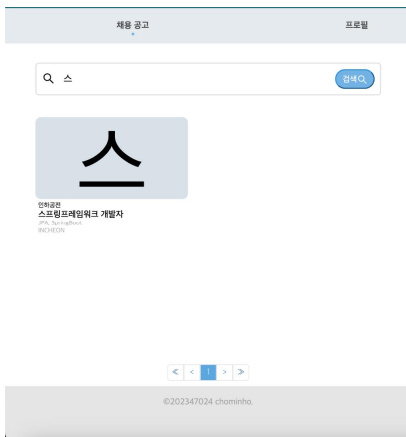


Fig. 1. Main page

2. Face recognition Login Services

face_recognition 라이브러리와 python Flask를 사용하여 얼굴을 인식하고 인식된 대상의 이미지 이름을 fig 2와 같이 JSON 형태로 반환한다.[3]

```
@app.route('/detectMember')
def detectMember():
    member = camera.detectMemer()
    print("member :", member)

    return jsonify({"imgName":member})
```

Fig. 2. main.py JSON Code

이렇게 반환된 데이터를 SpringBoot에서 받아와 데이터베이스 member 테이블에 있는 img_name 값 중 같은 값이 있는지 확인한 뒤 일치하는 값이 있다면 자동으로 로그인 할 수 있도록 서비스한다.

3. Face recognition Service Architecture

fig 3과 같이 사용자가 특정 사이트에 접근하면 JAVA서버로 JSON 요청을 보내고 서버에선 Machine Learning 서버로 요청을 보낸다. 그 후 클라이언트로 응답을 보내 인식된 이미지 이름을 localStorage에 저장한다.

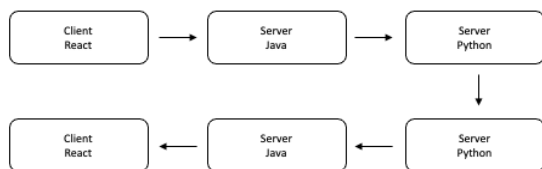


Fig. 3. Face Recognition Architecture 1

fig 4와 같이 localStorage에 저장된 이미지 이름을 Header에 담아 JAVA 서버로 요청을 보낸다. JAVA 서버에서는 Header에

담겨있는 이미지 이름을 통해 사용자를 검색하고 응답한다.

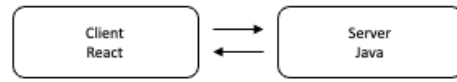


Fig. 4. Face Recognition Architecture 2

IV. Conclusions

현재 다양한 웹 사이트들에서 사용자를 위한 편의성을 제공하기 위해 다양한 방법으로 기능 간소화 방법을 모색하고 있다. 그러나 이러한 노력에도 불구하고, 회원가입 및 로그인에 대한 어려움은 여전히 사용자들에게 현실적인 문제로 남아있으며, 이로 인한 불편함이 더해지고 있다. 본 논문에서는 사용자 경험을 개선하기 위해 face_recognition 라이브러리를 활용하여 얼굴 인식 자동 로그인 시스템을 서비스한다.

이 얼굴 인식 시스템은 로그인 서비스뿐만 아니라 결제 시스템과 같이 다양한 사용자 개인 서비스에 적용될 수 있으며, 이를 통해 사용자들의 편의성이 크게 향상될 것으로 기대된다.

현재 얼굴 인식 시스템은 로컬서버에서 진행되고 있다. 추후 연구에서 구조를 확장하여 외부 서버에서 처리하며 Thread를 이용한 다중 접속 확인까지 진행할 계획이다.

REFERENCES

- [1] Spring Official Site (Site : <https://spring.io/>)
- [2] Face_recognition Github Site (Site :https://github.com/ageitgey/face_recognition)
- [3] JSON Official Site (Site :<https://www.json.org/json-ko.html>)