

**꼼꼼함과 세심함으로
좋은 결과를 만드는 개발자
현호성 입니다**

CONTACT

Tel : 010-4320-22215
Email : hhs04321@naver.com
Address : 경기도 의정부시 의정로 46번길 26-9
동진빌라 다동 301호








Portfolio

꼼꼼함과 세심함으로 가치를 더하는 개발자 **현호성**입니다



- 항상 **새로운 기술 스택**을 배우기 위해 노력하며, **변화하는 기술 환경**에 적응하기 위해 노력하고 있습니다.
- 언제나 **Java와 Spring** 기반 개발 환경에 깊은 관심을 가지고 있으며, 이를 활용한 프로젝트 **개발과 학습**을 지속적으로 이어가고 있습니다
- 코드를 작성할 때는 다른 개발자들이 읽기 쉽고 **이해하기 편하도록 구조적이고 깔끔한 코드**를 작성하는 데 집중합니다.
- 협업을 통해 더 나은 결과를 도출하는 개발자가 되기 위해 끊임없이 **소통과 팀워크**의 가치를 실천하고 있습니다.

보유 기술 스택

Front-End	HTML5, CSS3, Javascript(ES6)
Back-End	Java, Spring, PHP, JPA, RESTfull API
데이터베이스	MySQL, MariaDB
OS	Linux(CentOS)
Embedded System	Raspberry Pi, Arduino
ETC	      



[깃허브](#)



[티스토리](#)



[Discord](#)

GRADUATION

2018 발곡고등학교 졸업
2018 대진대학교 컴퓨터공학과 입학
2024 대진대학교 컴퓨터공학과 졸업

AWARDS

2023 벤처아카데미스타트업 우수상

PROJECT

2021 겐구판
2022 RPCar
2023 RLStock
2024 Smart Makeup

CERTIFICATION

2024.06 정보처리기사
2024.09 SQL 개발자

개발 프로젝트

2021.05~2021.06

개인 프로젝트



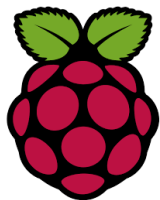
겜구판

프론트엔드 & 백엔드

HTML5, CSS, JavaScript,
PHP, MySQL을 사용해
게임 거래 웹
플랫폼의 **회원 관리**, **거래 기록 저**
장 등 다양한 기능 구현

2022.10~2022.12

팀 프로젝트



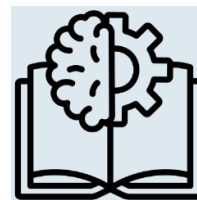
RPCar

임베디드 시스템

초음파 센서를 활용한 **실**
시간 장애물
탐지 및 **자동 회피**
알고리즘 구현

2023.08~2023.12

팀 프로젝트



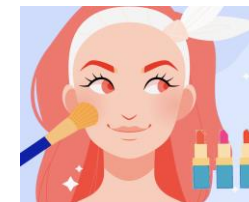
RLStock

인공지능

강화학습을 활용해
주식의 매수 및 매도 결정
을 돕는 모델을 구현

2024.09~2024.12

팀 프로젝트



Smart MakeUp

프론트엔드 & 백엔드

사용자의 **얼굴**을 카메라로
인식하고, 화장 효과를
실시간으로 적용하여
보여주는 **웹 서버**를 구현

개발자로서의 성장 로드맵

단계별 성장 과정과 목표

성장 비전

실전 경험

성장 단계

출발점

- 중학교 시절 HTML로 첫 웹 페이지를 만들며, 프로그래밍의 매력을 느끼게 되었습니다.

- **2018** C/C++, Python
- **2021** Java, HTML, CSS, Javascript, 자료구조
- **2022** 알고리즘, Android, 인공지능, 운영체제
- **2023** Database, Linux
- **2024** Spring Boot, FastAPI

- **RLStock**
 - A2C 알고리즘 기반 주식 가격 예측 및 거래 결정 시스템 개발
 - 시계열 데이터 분석 및 전처리
- **Smart MakeUP**
 - Spring Boot 기반의 화장 시뮬레이션 웹 서버 구현
 - Spring Boot와 FastAPI간의 서버 통신
 - 데이터베이스 설계 및 구축

- **웹 백엔드 전문가**
 - 안정적이고 확장 가능한 서버 구축에 강점을 가진 개발자
 - 대규모 트래픽을 처리할 수 있는 역량 보유
- **AI 전문가**
 - 데이터 분석 및 예측 모델 개발 역량을 갖추고, 실제 산업에 적용 가능한 AI 솔루션을 제공하는 전문가로 성장

RLStock

주가 예측 시스템

소개

강화 학습 알고리즘을 활용해 시장 데이터를 분석하고, 사용자의 주식 거래 결정을 지원하는 프로그램을 개발했습니다. 이를 통해 신뢰도 높은 주식 예측 서비스를 제공합니다

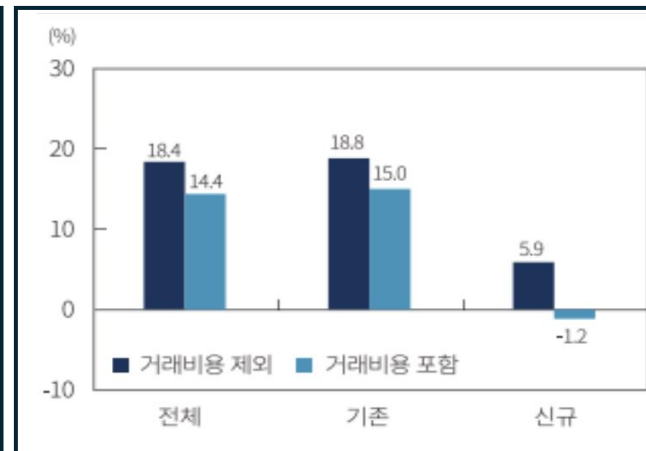
기간 / 인원

- **개발 기간** : 2023.08~2023.12
- **팀장** : 김기호
- **조원** : 정우창, 현호성
- **담당 기능**
 - 데이터 및 차트 분석
 - 발표 및 자료 수집
 - 데이터 학습

개발 목적



기존 예측 서비스



수익률 그래프

- **프로젝트 목표** : 강화 학습을 활용해 원자재 가격을 예측하고, 이를 기반으로 매수 및 매도 결정을 자동화하는 시스템을 구축합니다. 이를 통해 사용자들이 신뢰성 있는 투자 결정을 내릴 수 있도록 지원하는 것을 목표로 합니다.
- **문제 상황** : 초보 투자자들은 시장의 불확실성과 정보 부족으로 인해 올바른 투자 결정을 내리기 어려워 손실을 보는 경우가 많습니다.
- **해결 방안** : 강화 학습 모델을 활용해 시장 데이터를 분석하고, 실시간으로 투자 전략을 제안하여 손실을 최소화합니다. 이를 통해 초보 투자자들이 신뢰할 수 있는 정보에 기반한 올바른 투자 결정을 내릴 수 있도록 돕습니다.

개발 환경

- ## 구현 결과

진행 과정

	9월				10월				11월				12월	
주제 선정 및 자료조사														
데이터 수집 및 전처리														
데이터 학습 및 백테스팅														
검토														
1차 구현 및 발표														
수정 및 검토														
2차 구현 및 발표														
수정 및 검토														
3차 구현														

RLStock

담당 기능

1. 필요 정보 수집 및 알고리즘 조사
 - 주요 역할 : 주식 시장 **데이터**(Yahoo Finance 등)를 **수집**하고 강화 학습에 필요한 **알고리즘**(A2C, A3C 등) 관련 **자료 조사**
2. 데이터 전처리 및 강화 학습 모델 준비
 - 주요 역할 : 수집된 **데이터**를 강화 학습에 활용할 수 있도록 **전처리**(결측치 처리, 데이터 정규화 및 표준화 등)
3. 모델 학습 및 성능 평가
 - 주요 역할 : 강화 학습 모델을 수집된 **데이터**를 기반으로 **학습**시키고 학습 과정과 결과를 **테스트**하여 성능 평가
4. 프로젝트 발표
 - 주요 역할 : 모델 개발 진행 상황, 성과 및 분석 결과를 청중에게 **발표**하고 발표를 위한 **증거 자료 준비**(그래프, 코드 예시, 논문 인용 등)

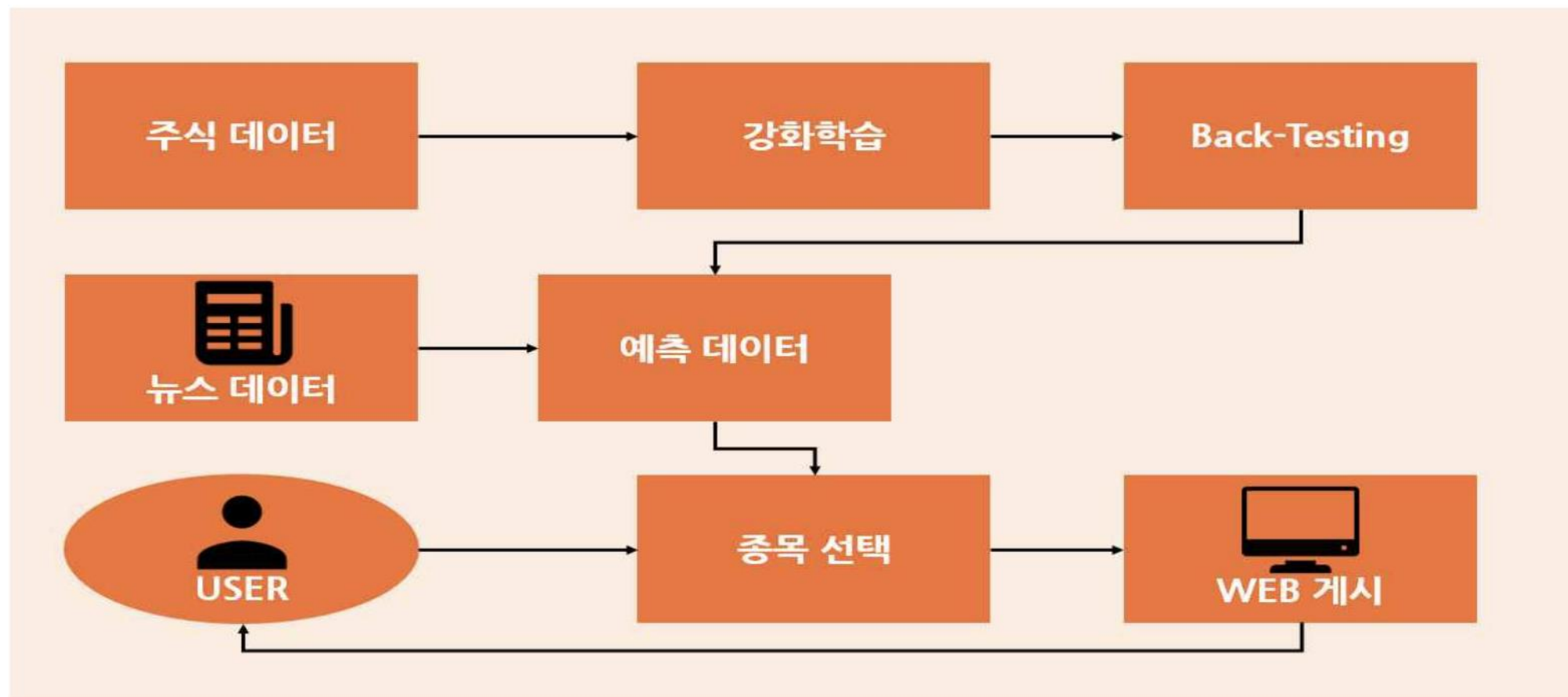
RLStock

강화학습 기반 투자 관리 시스템 구조



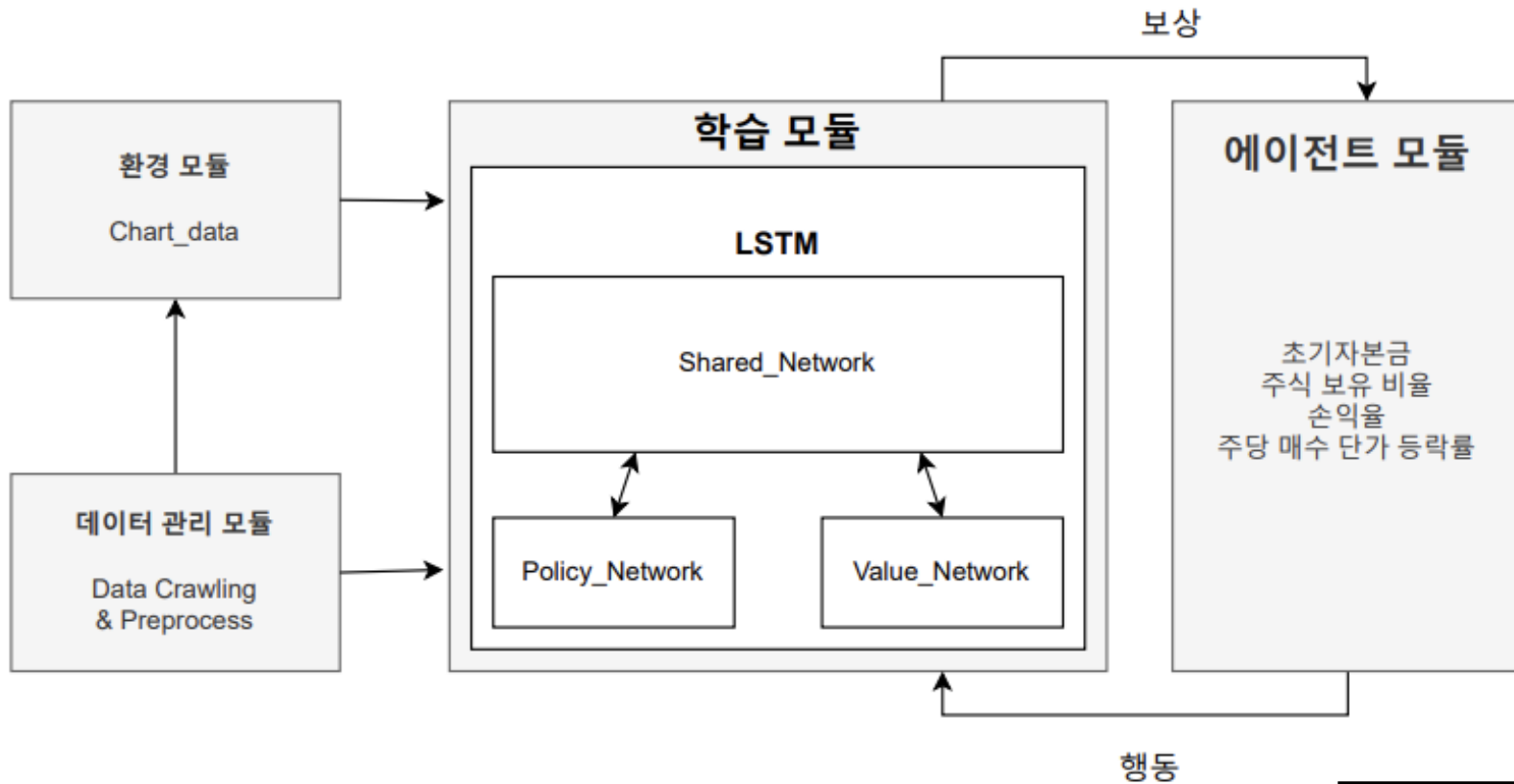
RLStock

데이터 흐름(강화학습 기반 투자 관리 시스템)



RLStock

System Architecture (학습 및 에이전트 구조)



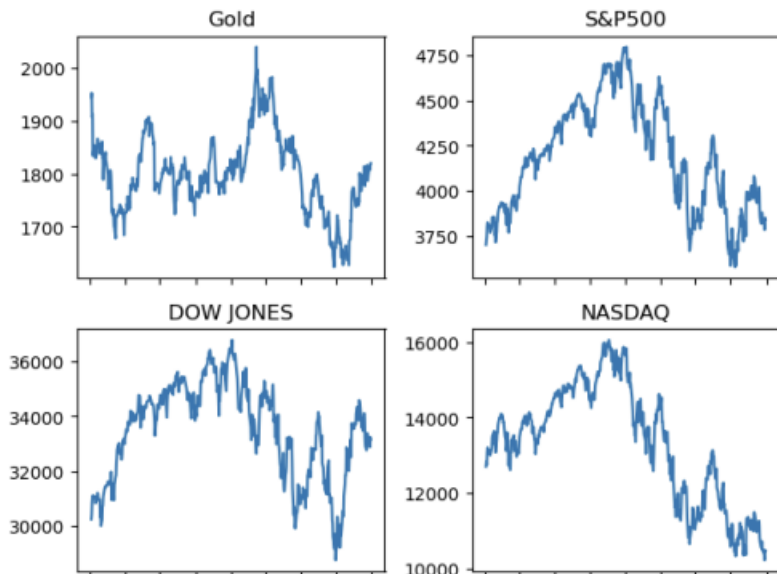
- **Shared_Network**: 데이터 특성을 추출하는 공통 네트워크
- **Policy_Network**: 행동을 결정하는 정책 네트워크
- **Value_Network**: 가치 평가를 담당하는 네트워크

1. 데이터 관리 모듈에서 데이터를 수집 및 전처리
2. 환경 모듈에서 학습 및 에이전트 테스트 실행
3. 학습 모듈에서 네트워크를 최적화하여 투자 행동 생성
4. 에이전트 모듈에서 투자 성과를 바탕으로 보상 제공

RLStock

정보 수집 및 데이터 전처리

종목 정보 및 경제지표 정보 수집



- Yahoo-Finance API로 원자재 주식 데이터,
S&P500, 다우존스, 나스닥 데이터를 수집

수집된 데이터 전처리

```
COLUMNS_TRAINING_DATA = [  
    'diffratio', 'open_lastclose_ratio', 'high_close_ratio', 'low_close_ratio', 'close_lastclose_ratio', 'volume_lastvolume_ratio',  
    'sma_close_ma5_ratio', 'sma_close_ma10_ratio', 'sma_close_ma20_ratio', 'sma_close_ma60_ratio', 'sma_close_ma120_ratio',  
    'sma_volume_ma5_ratio', 'sma_volume_ma10_ratio', 'sma_volume_ma20_ratio', 'sma_volume_ma60_ratio', 'sma_volume_ma120_ratio',  
    'ema_close_ma5_ratio', 'ema_close_ma10_ratio', 'ema_close_ma20_ratio', 'ema_close_ma60_ratio', 'ema_close_ma120_ratio',  
    'ema_volume_ma5_ratio', 'ema_volume_ma10_ratio', 'ema_volume_ma20_ratio', 'ema_volume_ma60_ratio', 'ema_volume_ma120_ratio',  
    'bhb_ratio', 'blb_ratio', 'rsi_ratio', 'sr_ratio', 'macd_ratio', 'obv_ratio', 'adi_ratio',  
    'sma_sp_close_ma5_ratio', 'sma_sp_close_ma10_ratio', 'sma_sp_close_ma20_ratio', 'sma_sp_close_ma60_ratio', 'sma_sp_close_ma120_ratio',  
    'sma_dji_close_ma5_ratio', 'sma_dji_close_ma10_ratio', 'sma_dji_close_ma20_ratio', 'sma_dji_close_ma60_ratio', 'sma_dji_close_ma120_ratio',  
    'ema_dji_close_ma5_ratio', 'ema_dji_close_ma10_ratio', 'ema_dji_close_ma20_ratio', 'ema_dji_close_ma60_ratio', 'ema_dji_close_ma120_ratio',  
    'sma_ixic_close_ma5_ratio', 'sma_ixic_close_ma10_ratio', 'sma_ixic_close_ma20_ratio', 'sma_ixic_close_ma60_ratio', 'sma_ixic_close_ma120_ratio',  
    'ema_ixic_close_ma5_ratio', 'ema_ixic_close_ma10_ratio', 'ema_ixic_close_ma20_ratio', 'ema_ixic_close_ma60_ratio', 'ema_ixic_close_ma120_ratio',  
    'sma_sp_volume_ma5_ratio', 'sma_sp_volume_ma10_ratio', 'sma_sp_volume_ma20_ratio', 'sma_sp_volume_ma60_ratio', 'sma_sp_volume_ma120_ratio',  
    'sma_dji_volume_ma5_ratio', 'sma_dji_volume_ma10_ratio', 'sma_dji_volume_ma20_ratio', 'sma_dji_volume_ma60_ratio', 'sma_dji_volume_ma120_ratio',  
    'ema_dji_volume_ma5_ratio', 'ema_dji_volume_ma10_ratio', 'ema_dji_volume_ma20_ratio', 'ema_dji_volume_ma60_ratio', 'ema_dji_volume_ma120_ratio',  
    'sma_ixic_volume_ma5_ratio', 'sma_ixic_volume_ma10_ratio', 'sma_ixic_volume_ma20_ratio', 'sma_ixic_volume_ma60_ratio', 'sma_ixic_volume_ma120_ratio',  
    'ema_ixic_volume_ma5_ratio', 'ema_ixic_volume_ma10_ratio', 'ema_ixic_volume_ma20_ratio', 'ema_ixic_volume_ma60_ratio', 'ema_ixic_volume_ma120_ratio',  
    'sp_close_diff', 'sp_volume_diff', 'dji_close_diff', 'dji_volume_diff', 'ixic_close_diff', 'ixic_volume_diff',  
    'buy_strength_ratio', 'sell_strength_ratio'  
]
```

$$Z = \frac{x - \mu}{\sigma}$$

x : 원 데이터(raw data)
z : 표준점수(standard score, z-score)
μ : 평균(mean)
σ : 표준편차(standard deviation)

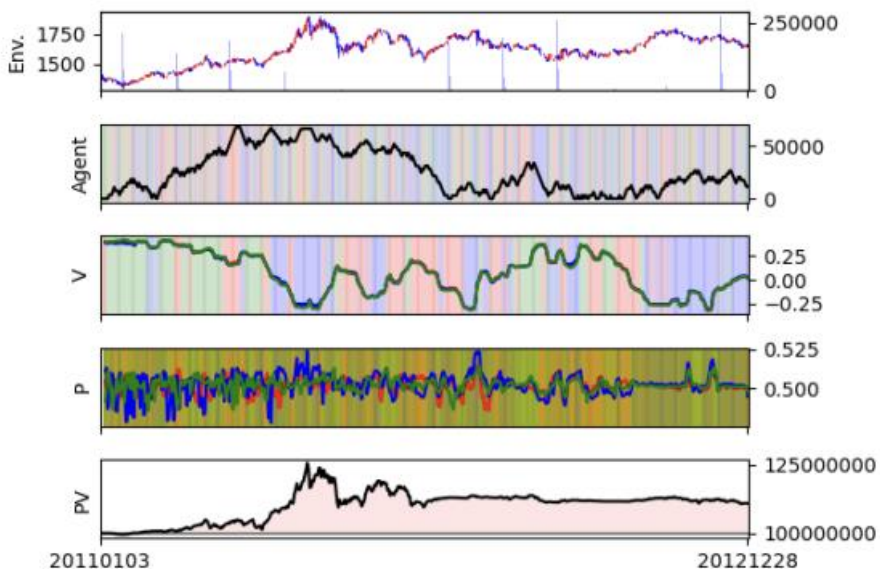
- 금 가격, 주식 시장 지수(S&P 500, 다우존스, 나스닥) 등은 서로 다른 특성과 범위를 가진 데이터이기 때문에 Z-SCORE 표준화를 통하여 데이터의 평균을 0, 표준 편차를 1로 변환

RLStock

학습 결과와 테스트 성능

학습 결과

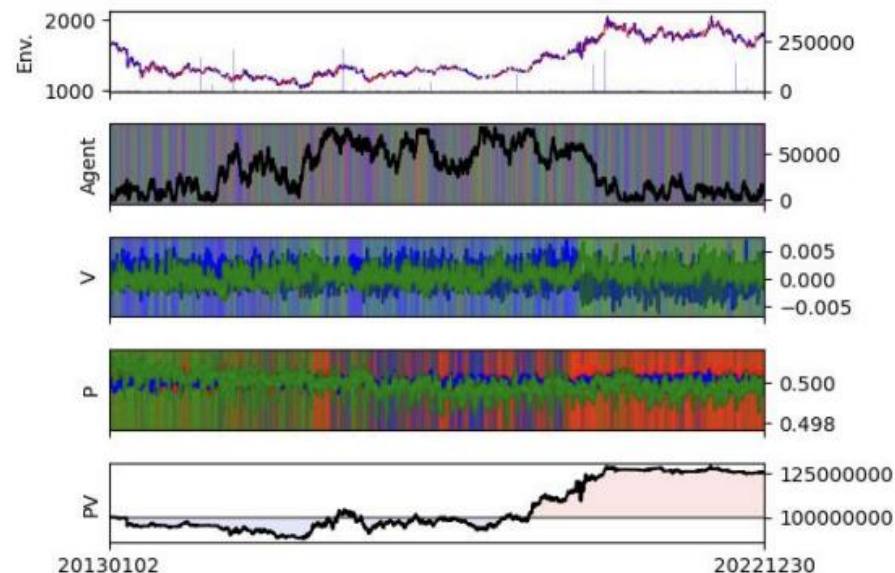
[GC=F] RL:a2c NET:lstm LR:0.001 DF:0.9
EPOCH:500/500 EPSILON:0.00



- **Env** : 실제 주식 거래소의 주가 데이터
- **Agent** : 주식 거래를 수행하는 가상 시스템으로, 실시간으로 매수, 매도, 관망 결정
 - 검은 실선 : 보유 중인 주식 수
 - 파란색 선 : 매도 신호
 - 빨간색 선 : 매수 신호
 - 초록색 선 : 관망 상태

테스트 성능

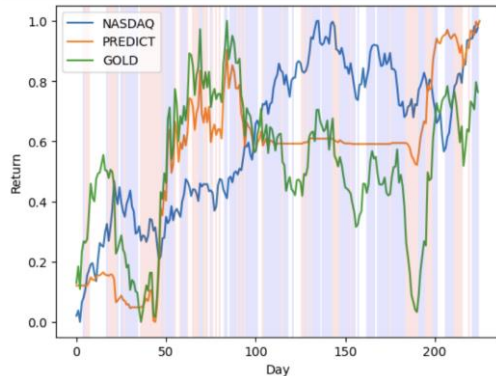
[GC=F] RL:a2c NET:lstm LR:0.001 DF:0.9
EPOCH:1/1 EPSILON:0.00



- **V** : 가치 신경망, 주식 거래의 가치를 평가하는 신경망
- **P** : 정책 신경망, 주식 거래 정책을 학습하고 최적화하는 신경망으로, 매수, 매도, 관망을 결정하는 역할
- **PV** : 주식 거래의 초기 자본금 변화 추이

RLStock

프로젝트 결과 및 기술적 통찰



월별 종목률	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월
금	4.89%	-5.13%	7.13%	0.35%	-1%	-2.88%	2.55%	-0.1%	-4.69%	8.46%	0.65%
예측	0.4%	-1%	5.56%	0.13%	-0.02%	-0.42%	0.17%	-0.11%	-0.14%	4.23%	0.59%
나스닥	11.53%	-3.05%	7.39%	0.3%	5.92%	5.24%	3.83%	-1.74%	-5.78%	-3.42%	9.21%

- **결과** : 프로젝트의 예측 시스템은 **총 9.74%의 수익률**을 기록하며, 신규 투자자들이 손실을 입지 않도록 시스템의 **안전성**과 **신뢰성**이 입증되었습니다.
- **개선 사항** : 강화학습 모델은 데이터 학습량 증가에 따라 성능이 개선되며, 향후 **20년치 데이터**를 추가 학습하여 **예측 정확도**를 높이는 것을 목표로 하고 있습니다.

배운 점

- **기술적인 배운점**
 - 강화학습 모델을 통해 **알고리즘 학습**과 최적화된 학습 환경 설정 방법을 배워, 머신러닝 성능 향상에 대한 이해를 높였습니다.
- **문제 해결**
 - 알고리즘 구현 중 **자료 부족**으로 어려움을 겪었으나, **해외 학술 자료**를 번역 및 분석하여 알고리즘 개선에 활용했습니다.
 - 데이터 수집 후 **가공 과정**에서 어려움을 겪었으나, 교수님의 조언을 바탕으로 전처리 기법을 적용하여 데이터를 모델에 **적합한 형태로 변환**하였습니다.
- **팀워크와 협업**
 - **RLStock 프로젝트**에서는 처음 만난 팀원들과 강화학습 알고리즘 선택, 학습할 데이터 범위(1년치 vs 2년치) 등에 대해 **의견 차이**가 있었습니다. **주기적인 회의**를 통해 각자의 의견을 조율하고, 데이터 분석과 실험을 통해 최적의 해결책을 도출했습니다. 이 과정에서 효과적인 **소통과 협업**이 프로젝트 성공의 핵심임을 깨달았고, 갈등을 해결하며 **팀워크의 중요성**을 실감할 수 있었습니다.

Smart MakeUP

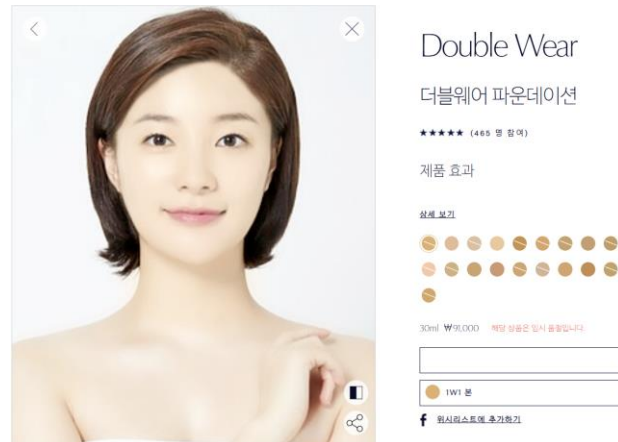
소개

사용자가 화장품을 구매하거나 직접 화장할 필요 없이, **미리 화장된 모습**을 보여주는 시스템을 개발한 프로젝트입니다. **Spring Boot**와 **FastAPI**를 활용하여 서버를 구축하였습니다. 사용자가 화장품 구매나 직접 테스트 없이 **미리 화장된 모습을 확인할 수** 있는 시스템입니다.

기간 / 인원

- **개발 기간**: 2024.09~2024.12
- **팀장**: 현호성
- **조원**: 한기윤, 최준원
- **담당 기능**
 - DB 설계 및 구현
 - Spring Boot를 활용 웹 개발
 - 기술서 등 문서 작업

개발 목적



- **프로젝트 목표**: 사용자 친화적인 웹 환경에서 **실시간 화장 시뮬레이션** 제공
- **문제 상황**: 다수의 화장품을 동시에 **시뮬레이션** 할 수 없는 기존 서비스의 한계
- **해결 방안**: **OpenCV**를 이용해 사용자의 **얼굴 이미지**를 분석하고, 각 **화장품의 효과**를 **실시간**으로 합성하여 시뮬레이션을 구현

Smart MakeUP

개발 환경

- **언어** : Python, Java, HTML5, CSS3, Javascript, Thymeleaf
- **프레임워크** : Spring boot, Boot Strap
- **API, 라이브러리** : OpenCV, Scikit-Learn, Pandas, Lombok, Matplotlib, Keras, Mediapipe, JPA
- **Database** : MySQL
- **IDE** : Visual Studio Code
- **Version Control** : Git

구현 결과

[https://github.com/hosunghyun/Smart Makeup Web](https://github.com/hosunghyun/Smart_Makeup_Web)

진행 과정

절차	추진일정				기간
	9월	10월	11월	12월	
프로젝트 주제 선정					09/19-09/19
프로젝트 계획서 작성					09/20-09/23
업무 기능도					09/24-09/26
단위업무 기술서					09/27-09/30
업무 흐름도					10/01-10/03
구현					10/04-11/15
버그 수정					11/16-12/04

Smart MakeUP

담당 기능

1. 데이터베이스 설계

- 주요 역할 : 프로젝트 요구 사항을 바탕으로 관계형 데이터베이스 설계 및 구축
- 사용한 기술
 - Database : MySQL
 - 설계 도구 : MySQL Workbench, Drawio, DBDiagram
 - 설계 결과 : 사용자, 제품, 주문 내역 등 주요 테이블 및 관계 설정

2. Spring Boot와 데이터베이스 연동

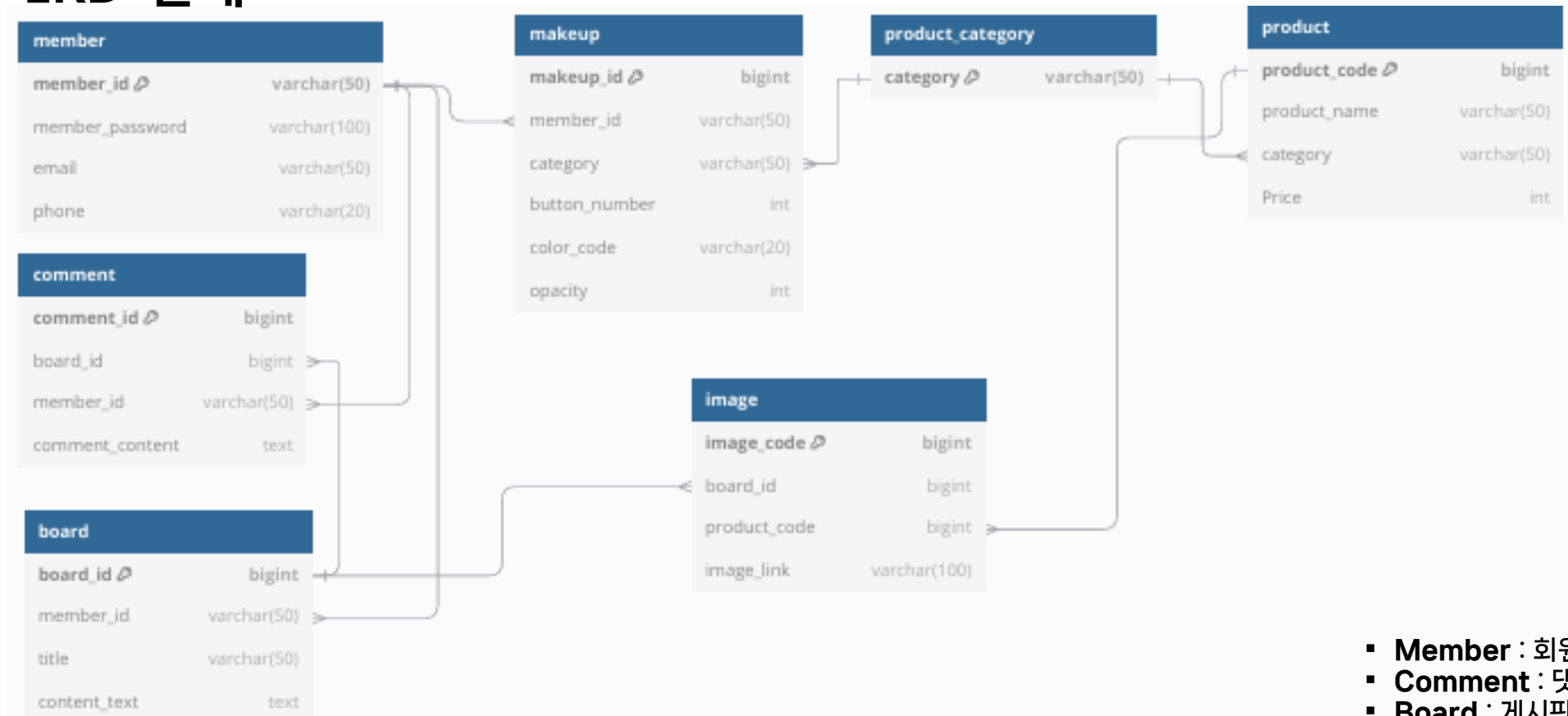
- 주요 역할 : Spring Boot 애플리케이션에서 JPA(Java Persistence API)와 데이터베이스 연동
- 구현 기능
 - 활용하여 데이터베이스와 연동 및 CRUD 기능 구현
 - Hibernate를 활용한 효율적인 데이터베이스 관리
- 사용한 기술 : Spring Boot, Spring Data JPA, Hibernate 등

3. UI/UX 구현 및 동적 웹 페이지 개발

- 주요 역할 : 사용자 친화적인 UI/UX 구현과 프로젝트 전체 웹 페이지 개발.
 - 주요 기능
 - 회원가입, 로그인, 제품 추천, 화장하기 페이지
 - 반응형 디자인 구현(Bootstrap)
- 사용한 기술 : HTML, CSS, JavaScript, Bootstrap, Thymeleaf

Smart MakeUP

ERD 설계

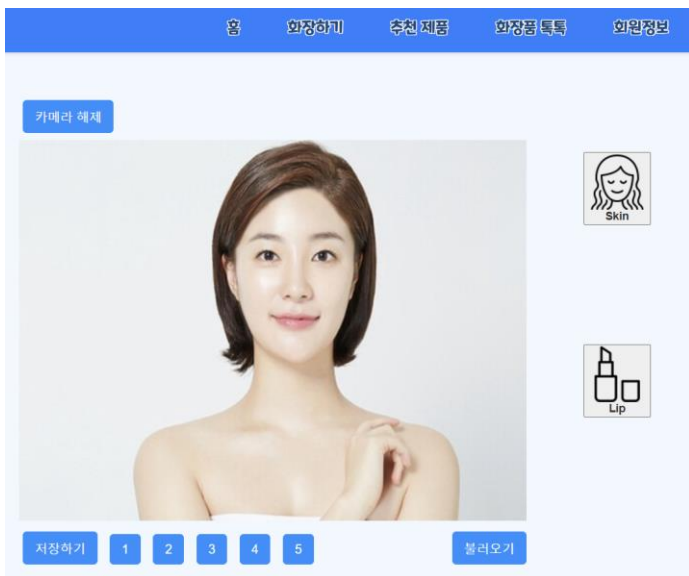


- **Member** : 회원 정보 저장
- **Comment** : 댓글 저장
- **Board** : 게시판 글 저장
- **Image** : 게시판 등의 이미지 파일이 링크 저장
- **MakeUP** : 사용자의 화장 정보(색상, 투명도) 저장
- **Product** : 화장품 정보 저장
- **Product Category** : 화장품 종류를 저장
(makeup과 product m:n 관계 해결)

Smart MakeUP

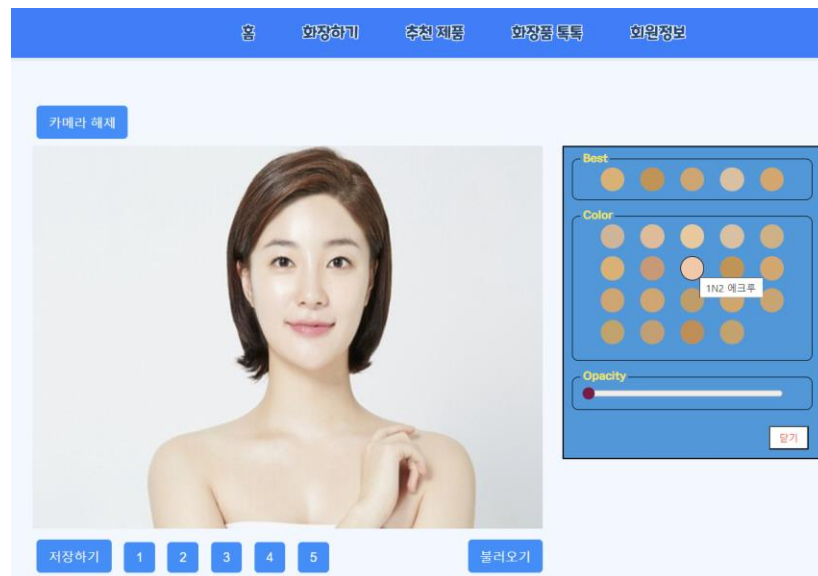
메이크업

카메라에 연결된 사용자 얼굴에 화장품 적용



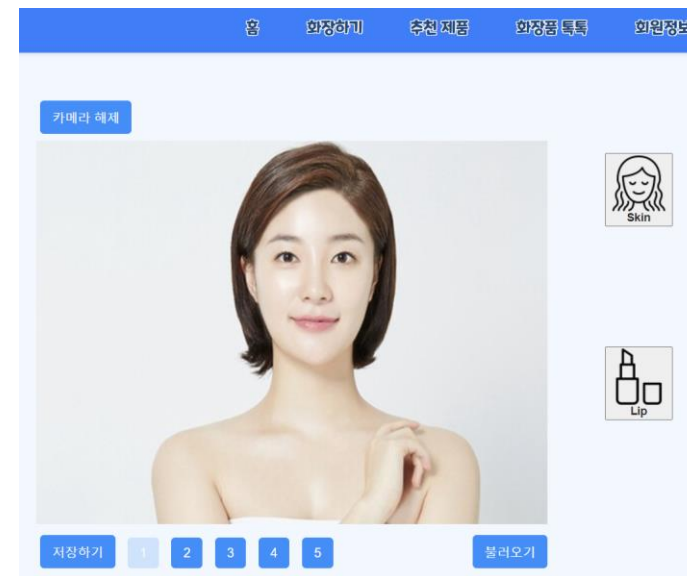
카메라 연결 기능

- 사용자의 웹캠과 연결하여 실시간으로 얼굴을 인식합니다.



화장하기 기능

- 베이스, 컬러, 투명도를 조정하여 원하는 메이크업을 적용합니다.



저장하기 및 블러오기 기능

- 적용된 메이크업 설정을 저장하고, 나중에 다시 불러옵니다

Smart MakeUP

프로젝트 결과 및 기술적 통찰



- **결과** : 사용자가 다양한 화장품을 적용한 모습을 실시간으로 확인할 수 있는 시스템을 구현하여, 여러 화장품을 적용한 사용자의 얼굴을 실시간으로 출력하는 웹 페이지를 성공적으로 구현했습니다.
- **개선 사항**
 - Python은 무한 반복문을 사용하여 영상 처리를 수행하는 과정에서 속도 저하 문제가 발생하고 있습니다. 이를 해결하기 위해, 향후 C++로 코드를 전환하여 더 효율적인 메모리 관리와 빠른 실행 속도를 통해 성능을 개선할 계획입니다.

배운 점

- **기술적인 배운점**
 - Spring Boot와 FastAPI를 활용하여 웹 서버를 구축하는 과정에서 웹 애플리케이션의 서버 사이드 개발에 대한 이해를 깊게 할 수 있었습니다. 특히, 서버 간 데이터 연동 및 API 설계에 대한 중요한 경험을 얻었습니다.
- **문제 해결**
 - FastAPI 서버가 동기 방식으로 구현되어 있었기 때문에 실시간 영상 처리에서 지연 시간이 증가하고, 리소스가 차단되며, 실시간 처리 성능이 부족한 문제가 발생했습니다. 이를 해결하기 위해 기존 동기 방식의 코드를 비동기 처리 방식으로 변경하여 성능을 최적화했습니다.
- **협업 및 팀워크**
 - Smart MakeUP 프로젝트를 진행하며 역할 분담의 중요성을 깊이 체감했습니다. 이번 프로젝트에서는 명확한 역할 분담을 통해 각자가 서버 구축, 영상 처리, 서버 통신 등 자신의 역할에 집중할 수 있었습니다. 또한, 주기적인 회의를 통해 진행 상황을 공유하고 의견을 조율하면서 팀워크가 강화되었고, 결과적으로 프로젝트의 완성도와 효율성을 크게 높일 수 있었습니다.

End of Portfolio

지금까지 제 포트폴리오를 읽어 주셔서 진심으로
감사합니다. 앞으로도 끊임없이 성장하며
더 나은 결과를 만들어 나가겠습니다.