



## 시험 성적서

1. 발 급 번 호 : JNDL-24Q-0105
2. 접 수 일 : 2024년 03월 06일
3. 신 청 인  
회 사 명 : 주식회사 스포잇  
주 소 : 인천광역시 연수구 송도문화로 119, B1층 B1006호  
(송도동, 인천글로벌캠퍼스)
4. 시 험 품 목  
제 품 명 : Sv1  
모 델 명 : N/A
5. 시 험 기 간 : 2024년 03월 20일 ~ 2024년 04월 26일
6. 시 험 조 건 : 의뢰자 제시 조건
7. 성적서 용도 : 제출용

확 인	시험자	승인자
	성 명 : 박 준 형 (서명)	성 명 : 윤 경 상 (서명)

2024 년 04 월 30 일

**(주)제이앤디엘 대표이사 (인)**

주소 : 경기도 안양시 동안구 학의로 282, B동 317, 318호 (관양동,금강펜테리움IT타워)  
TEL: (031)382-2277 FAX: (031)382-4542 <http://www.jndcerti.com>

본 성적서는 신청자로부터 제공된 시험품에 한하여 평가한 결과로서 전체 제품에 대한 품질 및 성능을 보장하지 않습니다.

**■ 시험성적서 발급 내역**

- 이 문서의 개정내역이 표시됩니다.

발행일	시험성적서 발급번호	발행내역
2024년 04월 30일	JNDL-24Q-0105	최초 발급
-	-	-
-	-	-





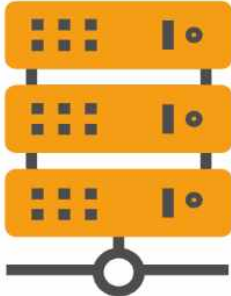


## 시험 내역


NO	시험 항목	Page
1	마킹 및 메모 영상 데이터 전송 성공률	6
2	영상데이터 변환 성공률	10
3	서비스 셧다운 Time	12
4	트레이너 추천 시스템의 추천 정확도	14
5	플랫폼 업로드 영상 및 기록 데이터 리포트 정확도	20



## 시험 신청제품 정보

◦ 소프트웨어 정보			
제 품 명(버전)		Sv1	
운영체제	서     버	CPU	Ubuntu Server
	앱	RAM	Google Chrome
	클라이언트	HDD	Windows 10 Professional
◦ 하드웨어 정보			
하드웨어 사     양	서     버	CPU	AWS Gravatation2 2 core
		RAM	4 GiB
		HDD	5 GB
	클라이언트	CPU	3.8 GHz 8-Core Intel Core i7
		RAM	32 GB
		HDD	1 TB
		GPU	AMD Radeon Pro 5700 8 GB
네트워크 환     경	AWS EC2를 통해 FLASK 프레임워크를 실행하여 웹사이트 호스팅		
<div><div></div><div></div><div></div></div> <div>클라이언트를 통한 웹사이트 접속</div> <div>AWS EC2 서버</div> <div>AWS S3 데이터베이스</div>			
시험환경 (장비 구성도)			



	
<p>운영체제 정보</p>	<p>하드웨어 정보</p>



## 마킹 및 메모 영상 데이터 전송 성공률

### 1. 시험 일자

- 2024. 04. 26

### 2. 시험 목표

- 99 % 이상

### 3. 사전 조건

- 현장에서 전달한 웹사이트 주소를 통해 클라이언트와 연결
- 화면 아래로 스크롤 해 영상 파일 객체 탐지 단락 확인
- UPLOAD 클릭

### 4. 시료

- 의뢰자가 제공한 1개의 이미지 데이터 (1\_13.jpg (568 KB))

### 5. 반복시험 횟수

- 100회

### 6. 시험 절차

- 파일 업로드 페이지에서 해당하는 1개의 영상 데이터를 삽입한 후, 파일 업로드를 클릭한다.
- UPLOADED IMAGES 버튼을 클릭해서, 이미지 파일을 다운로드한다.
- 해당 내용을 100회 반복하여, 이미지 파일이 성공적으로 다운로드 되었는지 확인한다.

### 7. 시험 결과

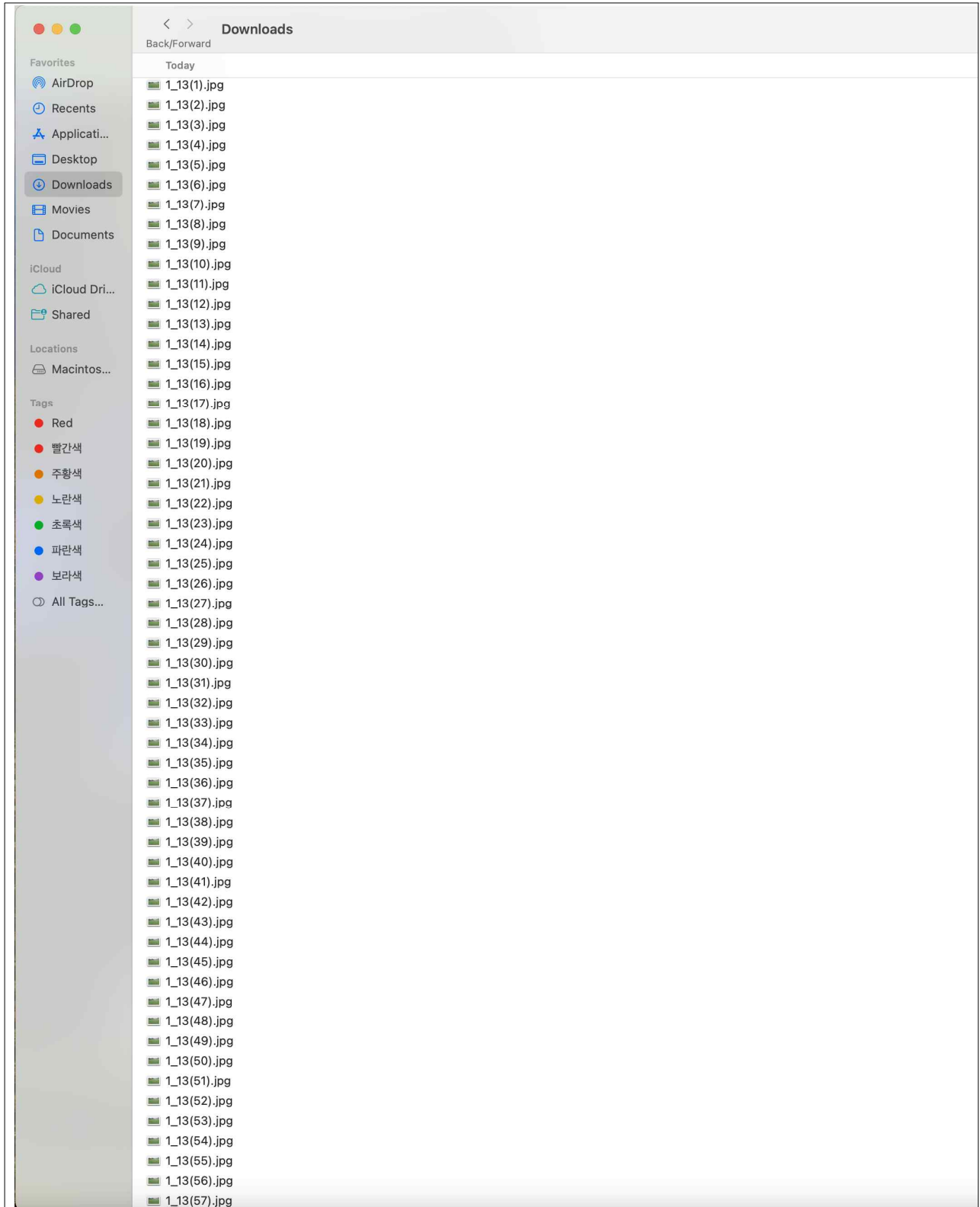
- 시험 결과 의뢰자가 제공한 1개의 이미지 데이터를 100회 모두 다운로드 된 것을 확인함.



▶ 시험결과



의뢰자가 제공한 이미지 데이터







이미지 데이터 100회 다운로드 된 스크린샷



## 영상데이터 변환 성공률

### 1. 시험 일자

- 2024. 03. 20

### 2. 시험 목표

- 99 % 이상

### 3. 사전 조건

- 현장에서 전달한 웹사이트 주소를 통해 클라이언트와 연결
- 화면 아래로 스크롤 해 영상 파일 객체 탐지 단락 확인
- UPLOAD 클릭

### 4. 시료

- 의뢰자가 제공한 100개의 영상 데이터

### 5. 반복시험 횟수

- 1회

### 6. 시험 절차

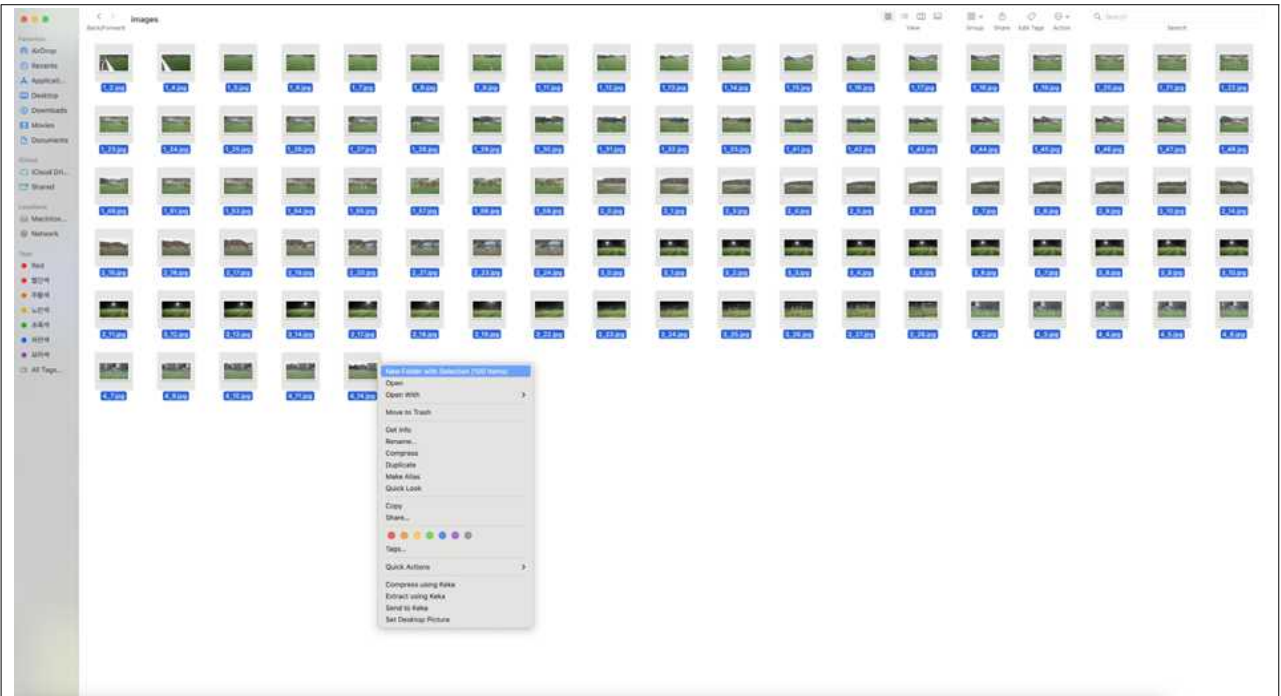
- 마킹 및 메모 영상 데이터 저장 전송률 시험을 마치고, UPLOADED IMAGES 밑에 있는 DETECTED IMAGES 버튼을 클릭해서 압축 파일을 다운로드 한다.
- 저장된 압축 파일을 압축 해제해서 100개의 탐지된 영상 데이터가 있는지 확인한다.
- 각 탐지 이미지마다 경계 상자 데이터가 담긴 pickle 파일을, 해당 이미지의 이상적인 경계 상자 데이터가 담긴 pickle 파일과 비교하여 iou를 계산한다.
- 한 이미지에서 발견되는 각 클래스 (player, referee, goalkeeper, ball) 들을 각각 구분하여 실제  $\cap$  예측 중복 영역과, 실제  $\cap$  예측 전체 영역을 각각 분자와 분모에 한 이미당 총 이상적인 경계 상자 데이터에 포함된 클래스의 개수 를 더해준다. 이미지의 갯수만큼 100번 반복하여 분자와 분모를 각각 축적하고, 분자를 분모로 나누고 100을 곱해 정확도를 나타낸다.
- 성공률 (%) = (모든 이미지의 실제  $\cap$  예측 중복 영역) / (모든 이미지의 실제  $\cap$  예측 전체 영역) \* 100 (%)

### 7. 시험 결과

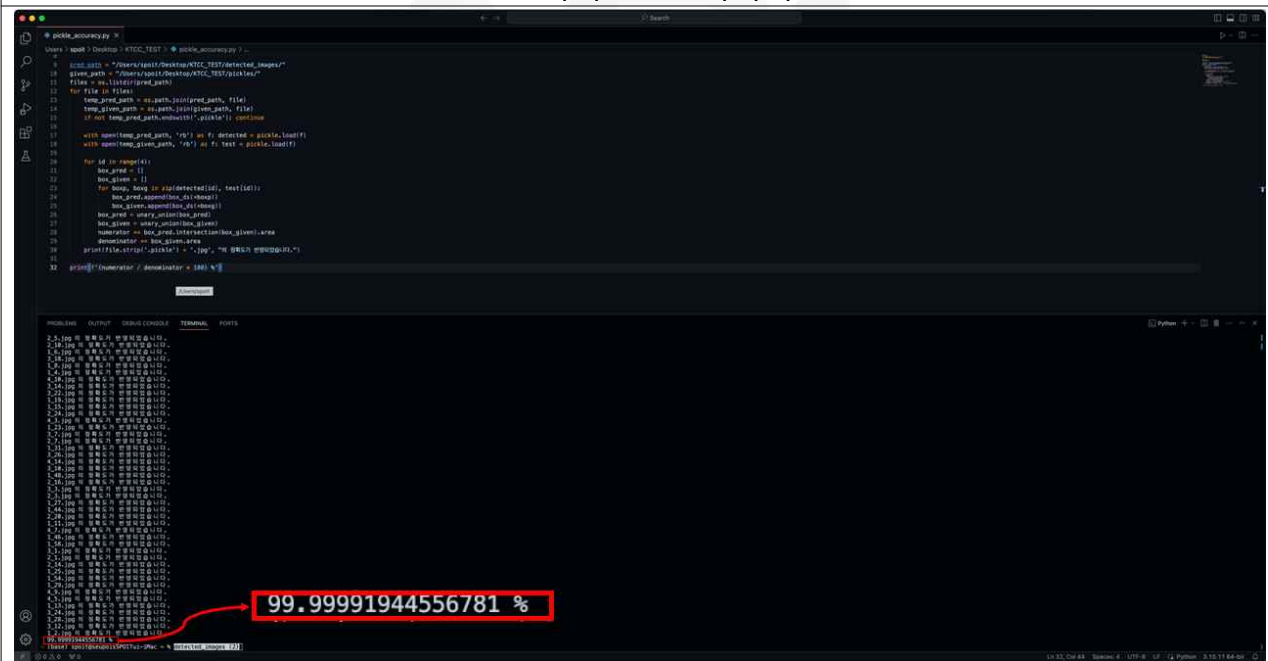
- 시험 결과 영상데이터 변환 성공률은 99.99 % 로 확인함.



## ▶ 시험결과



100개의 영상 데이터



성공률



## 서비스 셧다운 Time

### 1. 시험 일자

- 2024. 03. 20

### 2. 시험 목표

- 99.5 % 이상

### 3. 사전 조건

- 명령 프롬프트를 실행 후 커맨드라인을 입력할 준비를 마친다.

### 4. 시료

- 현장에서 전달받은 웹사이트 주소

### 5. 반복시험 횟수

- 1회

### 6. 시험 절차

- 전달받은 웹사이트에서 포트번호 (':8080')을 제외하고, 해당 명령어를 작성한다:
  - > ping -c 100 {유동적 ip 주소}
  - > ex) ping -c 100 3.35.140.156
- 정확도 (%) = 100 - packet loss (%)

### 7. 시험 결과

- 시험 결과 packet loss 는 0 % 였으므로 서비스 셧다운 time의 정확도는 100 % 로 확인함.



## ▶ 시험결과

```

Last login: Tue Mar 19 14:53:42 on ttys000
[chaos] nano@supersapiens:~$ ping -c 100 13.125.243.176
PING 13.125.243.176 (13.125.243.176): 36 data bytes
64 bytes from 13.125.243.176: icmp_seq=0 ttl=60 time=5.202 ms
64 bytes from 13.125.243.176: icmp_seq=1 ttl=60 time=5.117 ms
64 bytes from 13.125.243.176: icmp_seq=2 ttl=60 time=5.644 ms
64 bytes from 13.125.243.176: icmp_seq=3 ttl=60 time=5.740 ms
64 bytes from 13.125.243.176: icmp_seq=4 ttl=60 time=5.852 ms
64 bytes from 13.125.243.176: icmp_seq=5 ttl=60 time=5.673 ms
64 bytes from 13.125.243.176: icmp_seq=6 ttl=60 time=5.844 ms
64 bytes from 13.125.243.176: icmp_seq=7 ttl=60 time=5.654 ms
64 bytes from 13.125.243.176: icmp_seq=8 ttl=60 time=5.589 ms
64 bytes from 13.125.243.176: icmp_seq=9 ttl=60 time=48.746 ms
64 bytes from 13.125.243.176: icmp_seq=10 ttl=60 time=5.374 ms
64 bytes from 13.125.243.176: icmp_seq=11 ttl=60 time=5.728 ms
64 bytes from 13.125.243.176: icmp_seq=12 ttl=60 time=5.602 ms
64 bytes from 13.125.243.176: icmp_seq=13 ttl=60 time=5.810 ms
64 bytes from 13.125.243.176: icmp_seq=14 ttl=60 time=5.732 ms
64 bytes from 13.125.243.176: icmp_seq=15 ttl=60 time=5.714 ms
64 bytes from 13.125.243.176: icmp_seq=16 ttl=60 time=5.858 ms
64 bytes from 13.125.243.176: icmp_seq=17 ttl=60 time=5.643 ms
64 bytes from 13.125.243.176: icmp_seq=18 ttl=60 time=5.516 ms
64 bytes from 13.125.243.176: icmp_seq=19 ttl=60 time=5.753 ms
64 bytes from 13.125.243.176: icmp_seq=20 ttl=60 time=5.513 ms
64 bytes from 13.125.243.176: icmp_seq=21 ttl=60 time=5.801 ms
64 bytes from 13.125.243.176: icmp_seq=22 ttl=60 time=5.613 ms
64 bytes from 13.125.243.176: icmp_seq=23 ttl=60 time=5.725 ms
64 bytes from 13.125.243.176: icmp_seq=24 ttl=60 time=5.515 ms
64 bytes from 13.125.243.176: icmp_seq=25 ttl=60 time=5.816 ms
64 bytes from 13.125.243.176: icmp_seq=26 ttl=60 time=5.922 ms
64 bytes from 13.125.243.176: icmp_seq=27 ttl=60 time=5.806 ms
64 bytes from 13.125.243.176: icmp_seq=28 ttl=60 time=5.766 ms
64 bytes from 13.125.243.176: icmp_seq=29 ttl=60 time=5.608 ms
64 bytes from 13.125.243.176: icmp_seq=30 ttl=60 time=5.844 ms
64 bytes from 13.125.243.176: icmp_seq=31 ttl=60 time=6.489 ms
64 bytes from 13.125.243.176: icmp_seq=32 ttl=60 time=57.044 ms
64 bytes from 13.125.243.176: icmp_seq=33 ttl=60 time=5.749 ms
64 bytes from 13.125.243.176: icmp_seq=34 ttl=60 time=5.818 ms
64 bytes from 13.125.243.176: icmp_seq=35 ttl=60 time=8.116 ms
64 bytes from 13.125.243.176: icmp_seq=36 ttl=60 time=5.790 ms
64 bytes from 13.125.243.176: icmp_seq=37 ttl=60 time=5.136 ms
64 bytes from 13.125.243.176: icmp_seq=38 ttl=60 time=5.945 ms
64 bytes from 13.125.243.176: icmp_seq=39 ttl=60 time=5.625 ms
64 bytes from 13.125.243.176: icmp_seq=40 ttl=60 time=5.840 ms
64 bytes from 13.125.243.176: icmp_seq=41 ttl=60 time=7.822 ms
64 bytes from 13.125.243.176: icmp_seq=42 ttl=60 time=5.755 ms
64 bytes from 13.125.243.176: icmp_seq=43 ttl=60 time=5.723 ms
64 bytes from 13.125.243.176: icmp_seq=44 ttl=60 time=5.584 ms
64 bytes from 13.125.243.176: icmp_seq=45 ttl=60 time=5.679 ms
64 bytes from 13.125.243.176: icmp_seq=46 ttl=60 time=18.348 ms
64 bytes from 13.125.243.176: icmp_seq=47 ttl=60 time=5.131 ms
64 bytes from 13.125.243.176: icmp_seq=48 ttl=60 time=14.650 ms
64 bytes from 13.125.243.176: icmp_seq=49 ttl=60 time=5.682 ms
64 bytes from 13.125.243.176: icmp_seq=50 ttl=60 time=5.111 ms
64 bytes from 13.125.243.176: icmp_seq=51 ttl=60 time=9.843 ms
64 bytes from 13.125.243.176: icmp_seq=52 ttl=60 time=5.224 ms
64 bytes from 13.125.243.176: icmp_seq=53 ttl=60 time=5.980 ms
64 bytes from 13.125.243.176: icmp_seq=54 ttl=60 time=12.844 ms
64 bytes from 13.125.243.176: icmp_seq=55 ttl=60 time=5.273 ms
64 bytes from 13.125.243.176: icmp_seq=56 ttl=60 time=5.920 ms
64 bytes from 13.125.243.176: icmp_seq=57 ttl=60 time=5.664 ms
64 bytes from 13.125.243.176: icmp_seq=58 ttl=60 time=5.583 ms
64 bytes from 13.125.243.176: icmp_seq=59 ttl=60 time=5.623 ms
64 bytes from 13.125.243.176: icmp_seq=60 ttl=60 time=5.655 ms
64 bytes from 13.125.243.176: icmp_seq=61 ttl=60 time=5.709 ms
64 bytes from 13.125.243.176: icmp_seq=62 ttl=60 time=5.534 ms
64 bytes from 13.125.243.176: icmp_seq=63 ttl=60 time=5.812 ms
64 bytes from 13.125.243.176: icmp_seq=64 ttl=60 time=5.594 ms
64 bytes from 13.125.243.176: icmp_seq=65 ttl=60 time=5.594 ms
64 bytes from 13.125.243.176: icmp_seq=66 ttl=60 time=5.955 ms
64 bytes from 13.125.243.176: icmp_seq=67 ttl=60 time=5.763 ms
64 bytes from 13.125.243.176: icmp_seq=68 ttl=60 time=5.801 ms
64 bytes from 13.125.243.176: icmp_seq=69 ttl=60 time=5.639 ms
64 bytes from 13.125.243.176: icmp_seq=70 ttl=60 time=5.122 ms
64 bytes from 13.125.243.176: icmp_seq=71 ttl=60 time=5.186 ms
64 bytes from 13.125.243.176: icmp_seq=72 ttl=60 time=9.344 ms
64 bytes from 13.125.243.176: icmp_seq=73 ttl=60 time=5.549 ms
64 bytes from 13.125.243.176: icmp_seq=74 ttl=60 time=5.849 ms
64 bytes from 13.125.243.176: icmp_seq=75 ttl=60 time=5.841 ms
64 bytes from 13.125.243.176: icmp_seq=76 ttl=60 time=5.766 ms
64 bytes from 13.125.243.176: icmp_seq=77 ttl=60 time=5.696 ms
64 bytes from 13.125.243.176: icmp_seq=78 ttl=60 time=14.876 ms
64 bytes from 13.125.243.176: icmp_seq=79 ttl=60 time=5.780 ms
64 bytes from 13.125.243.176: icmp_seq=80 ttl=60 time=5.505 ms
64 bytes from 13.125.243.176: icmp_seq=81 ttl=60 time=5.719 ms
64 bytes from 13.125.243.176: icmp_seq=82 ttl=60 time=5.789 ms
64 bytes from 13.125.243.176: icmp_seq=83 ttl=60 time=5.722 ms
64 bytes from 13.125.243.176: icmp_seq=84 ttl=60 time=5.694 ms
64 bytes from 13.125.243.176: icmp_seq=85 ttl=60 time=5.200 ms
64 bytes from 13.125.243.176: icmp_seq=86 ttl=60 time=5.608 ms
64 bytes from 13.125.243.176: icmp_seq=87 ttl=60 time=5.708 ms
64 bytes from 13.125.243.176: icmp_seq=88 ttl=60 time=5.616 ms
64 bytes from 13.125.243.176: icmp_seq=89 ttl=60 time=5.524 ms
64 bytes from 13.125.243.176: icmp_seq=90 ttl=60 time=5.825 ms
64 bytes from 13.125.243.176: icmp_seq=91 ttl=60 time=5.814 ms
64 bytes from 13.125.243.176: icmp_seq=92 ttl=60 time=5.722 ms
64 bytes from 13.125.243.176: icmp_seq=93 ttl=60 time=5.245 ms
64 bytes from 13.125.243.176: icmp_seq=94 ttl=60 time=5.805 ms
64 bytes from 13.125.243.176: icmp_seq=95 ttl=60 time=8.121 ms
64 bytes from 13.125.243.176: icmp_seq=96 ttl=60 time=5.210 ms
64 bytes from 13.125.243.176: icmp_seq=97 ttl=60 time=7.823 ms
64 bytes from 13.125.243.176: icmp_seq=98 ttl=60 time=7.328 ms
64 bytes from 13.125.243.176: icmp_seq=99 ttl=60 time=5.750 ms

--- 13.125.243.176 ping statistics ---
100 packets transmitted, 100 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.847/5.887/27.472/3.137 ms
[chaos] nano@supersapiens:~$

```

--- 13.125.243.176 ping statistics ---  
100 packets transmitted, 100 packets received, 0.0% packet loss



# 트레이너 추천 시스템의 추천 정확도

## 1. 시험 일자

- 2024. 03. 20

## 2. 시험 목표

- 90 % 이상

## 3. 사전 조건

- 현장에서 전달한 웹사이트 주소를 통해 클라이언트와 연결
  - > 화면 아래로 스크롤 해 코치 추천 및 영상 분석 단락 확인
  - > UPLOAD 클릭
- 경기 데이터의 이름은 "(선수 이름)\_(경기 포지션).csv" 으로 정해진다.

## 4. 시료

- Abdalla Mahmoud El Said Bekhit\_Center Attacking Midfield.csv

## 5. 반복시험 횟수

- 100회

## 6. 시험 절차

- 파일 업로드 페이지에서 시료 데이터를 삽입한 후, 파일 업로드를 클릭한다.  
해당 경기 데이터는 같은 데이터를 100회 반복해서 업로드 / 분석한다.
- 해당 경기 데이터의 예측된 포지션이 안정적인 결과를 보이는지 계산한다. 가장 빈도가 잦은 결과의 빈도 수를 100회로 나눈 후, 100을 곱해 정확도를 계산한다.
- 정확도 (%) = (분석된 결과 중 가장 빈도가 잦은 결과 횟수) / (100) \* 100 (%)

## 7. 시험 결과

- 시험 결과 100회 모두 SIDE WING으로 예측 포지션 및 포지션에 맞는 트레이너(남준재 코치) 추천으로 분석한 것을 확인함.





## ▶ 시험결과 - 예측 포지션

[illegible]

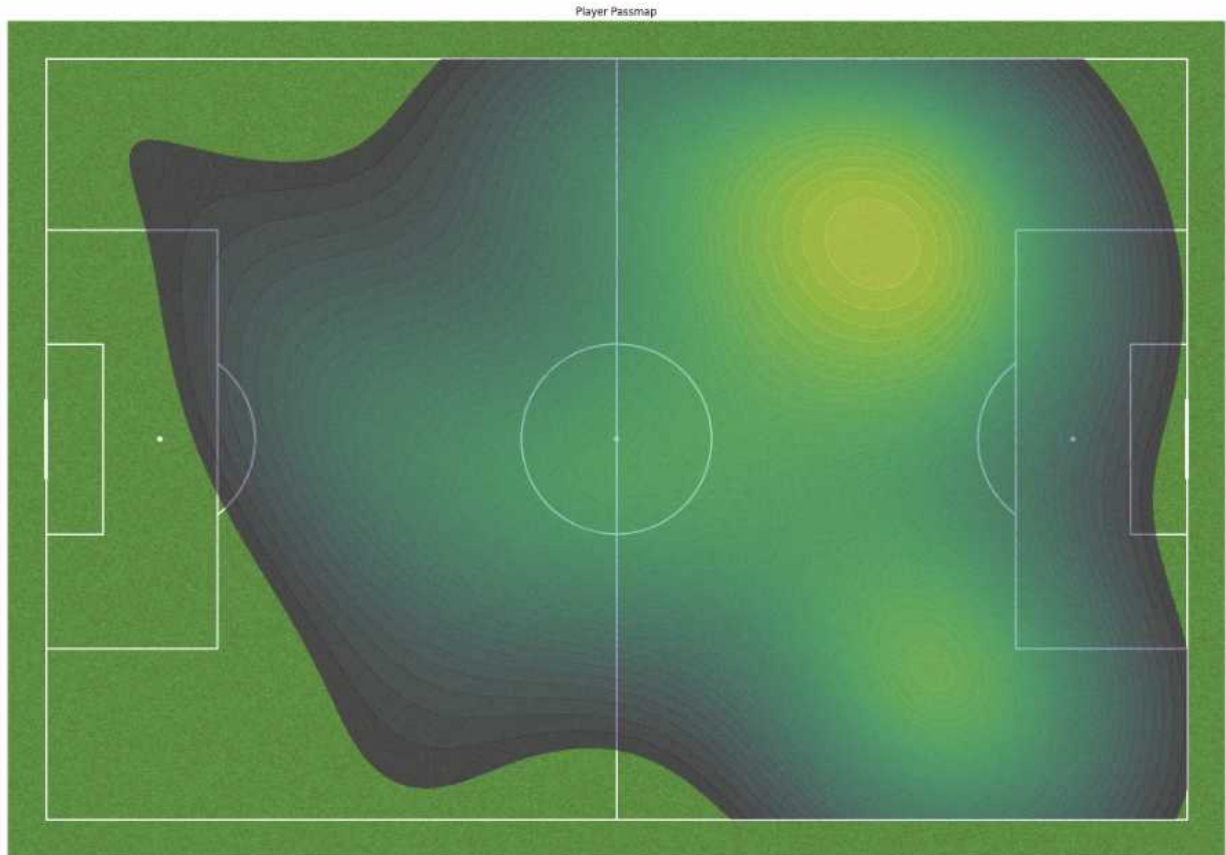


[illegible]





▶ 시험결과 - 포지션에 맞는 트레이너 추천 일부



**ABDALLA MAHMOUD EL SAID BEKHIT (1) 선수에게 추천드리는 포지션은 SIDE WING입니다.**

남준재 코치

Midfield, Side Wing 담당 코치

#1



**ABDALLA MAHMOUD EL SAID BEKHIT (50) 선수에게 추천드리는 포지션은 SIDE WING입니다.**

남준재 코치

Midfield, Side Wing 담당 코치

#50



**ABDALLA MAHMOUD EL SAID BEKHIT (100) 선수에게 추천드리는 포지션은 SIDE WING입니다.**

남준재 코치

Midfield, Side Wing 담당 코치

#100



## 플랫폼 업로드 영상 및 기록 데이터 리포트 정확도

### 1. 시험 일자

- 2024. 04. 26

### 2. 시험 목표

- 99 % 이상

### 3. 사전 조건

- 현장에서 전달한 웹사이트 주소를 통해 클라이언트와 연결
  - > 화면 아래로 스크롤 해 코치 추천 및 영상 분석 단락 확인
  - > UPLOAD 클릭
- 경기 데이터의 이름은 "(선수 이름)\_(경기 포지션).csv" 으로 정해진다.

### 4. 시료

- Abdalla Mahmoud El Said Bekhit\_Center Attacking Midfield.csv

### 5. 반복시험 횟수

- 100회

### 6. 시험 절차

- 파일 업로드 페이지에서 해당하는 1개의 선수 기록 데이터를 삽입한 후, 파일 업로드를 클릭한다.
- UPLOADED IMAGES 버튼을 클릭해서, 선수 기록 데이터 파일을 다운로드 한다.
- 해당 내용을 100회 반복하여, 선수 기록 데이터 파일이 성공적으로 다운로드 되었는지 확인한다.

### 7. 시험 결과

- 시험 결과 의뢰자가 제공한 1개의 선수 기록 데이터를 100회 모두 다운로드 된 것을 확인함.



## ▶ 시험결과

