

- Date 2020-09-18

-

Team13

System Testing Plan

for Smart Traffic Light System

201411285 유종혁

201411286 유환규

201611308 최준오

201714166 신예슬

- Test Identifier: STLS_1.1

기능/비 기능: 사진촬영

Input: N/A

Output: 도로 사진

Des: 도로 상황을 4 대의 카메라로 촬영하여 메인 컴퓨터에 전송할 수 있어야 한다. 사진 안에는 도로의 차량들과 사진 기준 도로의 왼쪽 보행자들을 담고 있어야 한다.

Test Case:

- 1) 4 대의 웹 캠으로 0.5 초에 각 도로마다 1 장씩 현재의 도로 상황 사진이 촬영하는지 확인한다.

- Test Identifier: STLS_1.2

기능/비 기능: 이미지 전송

Input: N/A

Output: N/A

Des: 이미지를 서버에 전송할 수 있어야 한다. 이때 4 개의 도로 상황 이미지가 항상 일정하게 정렬되어 전송하여야 한다.

Test Cases:

- 1) 촬영된 사진이 실시간으로 네트워크를 통해 서버에 도착하는지 확인한다.

- Test Identifier: STLS_1.3

기능/비 기능: Object Detection

Input: 도로 상황 사진

Output: N/A

Des: 전송 받은 이미지에서 도로 위 객체들을 식별하는지 확인한다.

Test Case:

- 1) RetinaNet 모델을 이용해 전송 받은 이미지에서 도로 위 자동차 객체의 Bounding Box 를 찾고, 객체를 식별하는지 확인한다.
- 2) RetinaNet 모델을 이용해 전송 받은 이미지에서 보행자 도로 위 사람 객체의 Bounding Box 를 찾고, 객체를 식별하는지 확인한다.
- 3) RetinaNet 모델을 이용해 전송 받은 이미지에서 도로 위 구급차 객체의 Bounding Box 를 찾고, 객체를 식별하는지 확인한다.

- Test Identifier: STLS_1.4

기능/비 기능: Object Counting 및 전송

Input: 객체 수를 담은 딕셔너리

Output: N/A

Des: 이미지에서 찾은 객체들의 수를 세고 신호 스케줄러로 보낸다.

Test Cases:

- 1) RetinaNet 모델로부터 얻어낸 자동차 객체의 Class 정보로 도로 위 자동차 객체의 수를 세고, Dictionary 에 저장되는지 확인한다.
- 2) RetinaNet 모델로부터 얻어낸 사람 객체의 Class 정보로 보행자 도로 위 사람 객체의 수를 세고, Dictionary 에 저장되는지 확인한다.
- 3) 각 객체들의 수가 저장된 Dictionary 가 신호 스케줄링을 진행하는 컴퓨터로 전송되는지 확인한다.

- Test Identifier: STLS_1.5

기능/비 기능: 기본 신호 스케줄링

Input: 도로의 객체 수

Output: N/A

Des: 사거리에서 모든 방향에서 차량(또는 보행자)이 존재한다고 가정했을 때 기본적으로 동작하는 신호 스케줄링이다.

Test Cases:

- 1) 도로를 위에서 아래로 보는 것을 기준으로 왼쪽도로, 오른쪽도로, 아래쪽도로, 위쪽도로 순으로 신호를 주는지 확인한다.
- 2) 도로 신호등의 기본값은 40 초, 보행자 신호등의 기본값은 20 초로 설정되었는지 확인한다.
- 3) 다음 신호로 바뀌기 전 2 초간 황색신호를 부여하는지 확인한다.
- 4) 황색신호가 켜지면 각 도로마다 자동차 수와 보행자 수로 가중치를 계산해 부여하는지 확인한다.
- 5) 다음 켜져야 할 도로 신호의 계산된 가중치가 상대적으로 크다면 기본값인 40 초에 추가 시간을 부여하고, 가중치가 작다면 시간을 단축시킨다.

- Test Identifier: STLS_1.6

기능/비 기능: 구급차 출현 스케줄링

Input: 도로의 객체 수

Output: N/A

Des: 도로에 구급차가 나타났을 때 신호 스케줄링은 현재 신호를 잠시 멈추게 하고 구급차가 나타난 도로 쪽에 신호를 주어야 한다.

Test Cases:

- 1) 현재 켜진 도로에 구급차가 있는 경우, 도로 신호의 남은 시간이 10 초 이하라면 남은 시간을 10 초로 연장하는지 확인한다.
- 2) 현재 켜진 도로에 구급차가 있는 경우, 도로 신호의 남은 시간이 10 초 보다 많이 남았다면 그대로 신호를 유지하는지 확인한다.
- 3) 현재 켜진 도로가 아닌 다른 도로에 구급차가 있는 경우에 켜진 도로의 남은 시간이 10 초 이하라면 그대로 둔 후 다음으로 켜질 신호를 구급차가 있는 도로 신호로 변경, 10 초의 시간을 부여하는지 확인한다.
- 4) 현재 켜진 도로가 아닌 다른 도로에 구급차가 있는 경우에 켜진 도로의 남은 시간이 10 초 보다 많이 남았다면 10 초로 줄인 후 다음으로 켜질 신호를

구급차가 있는 도로 신호로 변경, 10 초의 시간을 부여하고 다시 원래의 신호로 돌아와 남았던 시간만큼 부여하는지 확인한다.

- 5) 구급차가 통과해야 하는 도로에 보행자 신호가 켜져 있는 경우, 10 초 후에 현재 켜진 도로 신호를 적색 신호로 변경하고 보행자 신호가 끝날 때까지 기다린 후 곧바로 구급차 진행방향의 도로 신호를 부여하는지 확인한다.
- 6) 여러 대의 구급차가 도로에서 발견된 경우, 나타난 순서대로 큐에 저장되었다가 큐가 비워질 때까지 처리해주는지 확인한다.
- 7) 구급차가 나타나서 구급차의 진행방향으로 신호를 부여 받았다면 구급차가 지나가고 다시 원래의 스케줄대로 돌아가는지 확인한다.

- Test Identifier: STLS_1.7

기능/비 기능: 모든 도로에 자동차, 보행자가 없는 스케줄링

Input: 도로에 객체 수

Output: N/A

Des: 도로에 차나 보행자가 없을 경우 효율성을 높이기 위해서 도로들의 신호를 추가적으로 관리한다.

Test Cases:

- 1) 모든 도로에 자동차와 보행자가 없는 경우, 신호를 모두 적색 신호로 유지하는지 확인한다.
- 2) 새로운 차량이나 보행자가 나타난다면 처음으로 나타난 도로부터 시작해 신호 스케줄대로 돌아가는지 확인한다.
- 3) 다음 신호가 켜질 도로에 차량과 보행자가 없는 경우, 해당 도로에는 신호를 부여하지 않고 그 다음 도로의 신호로 넘어가는지 확인한다.

- Test Identifier: STLS_1.8

기능/비 기능: UI

Input: N/A

Output: UI

Des: 신호 스케줄러가 정한 신호 스케줄대로 UI가 동작한다.

Test Cases:

- 1) 신호 시스템의 결과가 UI에 반영되어 도로 신호, 보행자 신호, 부여된 시간 등을 볼 수 있는지 확인한다.
- 2) UI 업데이트 주기를 확인한다.
- 3) 한 신호등 지속시간이 끝나면 바로 다음 신호등 지속시간으로 이어지는지 확인한다.
- 4) 신호등의 현재상태에 따라 알맞게 신호등 색이 변하는 것을 확인한다. (통과신호: 초록, 변경 예고신호: 황색, 정지신호: 빨강)
- 5) 횡단보도의 현재상태에 따라 알맞게 신호등 색이 변하는 것을 확인한다. (통과신호: 초록, 정지신호: 빨강)

- Test Identifier: STLS_2.1

기능/비기능: 비기능

Input: 교통량 정보 및 각 신호등의 신호 정보

Output: 모니터를 통해 GUI로 신호등을 표현

Des: 모니터를 통해 화면을 분할하여 도로 신호와 보행자 신호를 한 번에 표시할 수 있어야 한다.

Test Cases:

- 1) 프로그램 실행 후 MainUI에서 도로 신호 UI로 이동하여 도로신호등 표시 여부를 확인한다
- 2) 프로그램 실행 후 MainUI에서 횡단보도 UI로 이동하여 횡단보도등 표시 여부를 확인한다.

- Test Identifier: STLS_2.2

기능/비기능: 비기능

Input: 4 장의 도로 이미지

Output: Test 측정시간

Des: 딥러닝을 이용한 Object Detect 의 Processing 시간을 확인한다.

Test Cases:

- 1) 딥러닝 모델 RetinaNet 에 4 개의 이미지를 Input 으로 주었을 때 수행시간을 확인한다.

- Test Identifier: STLS_2.3

기능/비기능: 비기능

Input: 객체 수 디렉터리 데이터

Output: 테스트 시간

Des: 분석데이터를 이용한 신호 스케줄링 걸리는 시간을 확인한다.

Test Cases:

- 1) 객체 수를 담고 있는 임시 디렉터리 데이터를 신호 스케줄링 프로그램에 전송하여 UI 에 실시간 신호가 반영되는 때까지 걸리는 시간을 측정한다.

- Test Identifier: STLS_2.4

기능/비기능: 비기능

Input: 구급차 포함된 도로 이미지

Output: 구급차 신호변경 측정시간

Des: 구급차 통과시 신호 변경 요청을 반영하는데 필요한 시간을 측정한다.

Test Cases:

- 1) 구급차에 대한 객체 정보가 들어왔을 때 구급차 예외상황을 처리하는 데 필요한 시간을 측정한다.

- Test Identifier: STLS_2.5

기능/비기능: 비기능

Input: 상황별 웹캠 이미지

Output: 객체 정확도

Des: Object detect 의 정확도를 측정한다.

Test Cases:

- 1) 일반 자동차 검출 정확도를 확인한다. 자동차가 있는 이미지를 넣었을 때 정확히 그 대수에 맞는 양이 검출되는지 확인한다.
- 2) 구급차 검출 정확도를 확인한다. 구급차가 있는 IMAGE 를 넣었을 때 구급차가 검출되는지 확인한다.
- 3) 사람 검출 정확도를 확인한다. 사람이 있는 IMAGE 를 넣었을 때 이미지에 있는 사람 수만큼의 객체가 검출되는지 확인한다.

- Test Identifier: STLS_2.6

기능/비기능: 비기능

Input: 딥셔너리 데이터

Output: 교통량 측정값

Des: 기존 신호시스템보다 실시간 신호시스템 반영 시 개선 비율을 측정한다.

Test Cases:

- 1) 기본 동작 스케줄링의 교통량과 실시간 신호시스템의 차량과 보행자의 평균 대기시간을 비교 측정하여 개선 비율을 측정한다

- Pass/Fail Criteria

1. Object Detection

- 1) STLS_1.3

: Test Case 1~3 전부 통과

- 2) STLS_1.4

: Test Case 1~3 전부 통과

- 3) STLS_2.2

: Test Case 1 을 0.5 초 이내로 통과

- 4) STLS_2.5

: Test Case 1~3 을 각각 90% 이상으로 통과

2. 신호 스케줄링 시스템

- 1) STLS_1.5

: Test Case 1~5 전부 통과

- 2) STLS_1.6

: Test Case 1~7 전부 통과

- 3) STLS_1.7

: Test Case 1~3 전부 통과

- 4) STLS_2.4

: Test Case 1 이 1 초 안에 수행되면 통과

5) STLS_2.6

: Test Case 1 이 10% 이상 개선율을 보이면 통과

3. 신호 UI

1) STLS_1.8

: Test Case 2 번의 주기가 1 초 미만이면 통과

Test Case 1, 3~5 통과

2) STLS_2.1

: Test Case 1~1 통과