파이썬 과제물

(기법 활용 프로그램 만들기)

2020. 11. 02

주제. 자율주행 데이터 서비스 사업

스프링고 – STEP

“주행상황 인지SW 성능” 검증을 위한 주행환경 데이터베이스, 평가모델 및 검증용 플랫폼 / **표준화 및 평가 프로세스 개발**

AV 평가 시나리오 개발

[상황 별 요구사항]

도로, 환경, 날씨 등 다양한 조건을 통한 시나리오 개발

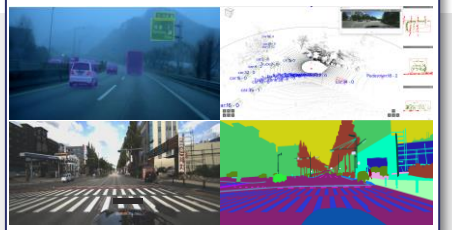
센서 융합 데이터 취득

SpringCloud AV fleet

SpringCloud AV Shuttles

자율주행 센서 차량을 이용한 센서융합데이터 취득

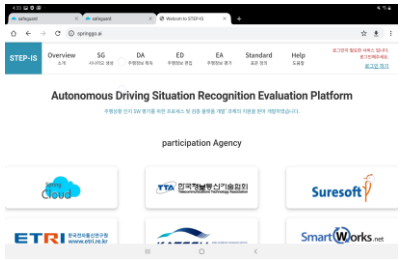
데이터 가공



평가용 데이터셋 구성

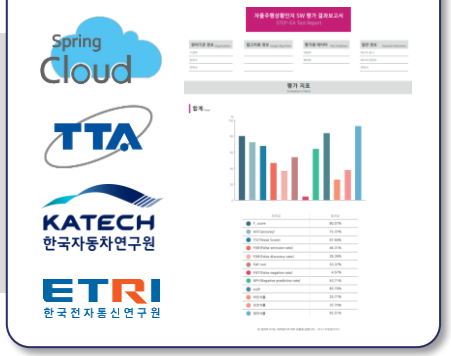
인지 SW 평가용 데이터셋 구성을 위한 데이터 가공

인지 SW 성능 평가 플랫폼



인지 SW 검증 및 평가용 플랫폼 개발

표준 평가 및 검증



개발 플랫폼을 통한 인지 SW 표준 평가 및 검증

코딩 Coding

```
#!/usr/bin/env python
import cv2
import numpy as np
import math
import matplotlib.pyplot as plt


def histogram_equilization(image):
enhanced_image = cv2.equalizeHist(image)
cv2.imshow("histogram equilization", enhanced_image)
return enhanced_image


def canny_edge_detector(image):
# Convert the image color to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
cv2.imshow("Input Grey image", gray_image)

# Use histogram equilization
enhanced_grey = histogram_equilization(gray_image)

# Reduce noise from the image
blur = cv2.GaussianBlur(enhanced_grey, (9, 9), 0)
cv2.imshow("Gaussian Smoothening", blur)

canny = cv2.Canny(blur, 50, 170)
cv2.imshow("Canny Edge", canny)

return canny


def region_of_interest(image):
height = image.shape[0]
polygons = np.array([
[(150, height), (850, height), (550, 250)]
])
mask = np.zeros_like(image)

# Fill poly-function deals with multiple polygon
cv2.fillPoly(mask, polygons, 255)
# cv2.imshow("mask", mask)
```

```
# Bitwise operation between canny image and mask image
masked_image = cv2.bitwise_and(image, mask)
return masked_image


def create_coordinates(image, line_parameters):
# print('line:', line_parameters)
slope, intercept = line_parameters
y1 = int(image.shape[0])
y2 = int(image.shape[0] / 2 + 100)
x1 = int((y1 - intercept) / slope)
x2 = int((y2 - intercept) / slope)
return np.array([x1, y1, x2, y2])


def average_slope_intercept(image, lines):
left_fit = []
right_fit = []
for line in lines:
x1, y1, x2, y2 = line.reshape(4)

# It will fit the polynomial and the intercept and slope
parameters = np.polyfit((x1, x2), (y1, y2), 1)
slope = parameters[0]
intercept = parameters[1]
if slope < 0:
left_fit.append((slope, intercept))
else:
right_fit.append((slope, intercept))

left_fit_average = np.average(left_fit, axis=0)
right_fit_average = np.average(right_fit, axis=0)
print('left:', left_fit_average, 'right', right_fit_average)
left_line = create_coordinates(image, left_fit_average)
right_line = create_coordinates(image, right_fit_average)
return np.array([left_line, right_line])


def display_lines(image, lines):
line_image = np.zeros_like(image)
if lines is not None:
```

```
for x1, y1, x2, y2 in lines:
cv2.line(line_image, (x1, y1), (x2, y2), (0, 255, 0), 10)
return line_image


# Path of dataset directory
cap = cv2.VideoCapture("test1.mp4")
while (cap.isOpened()):
_, frame = cap.read()
canny_image = canny_edge_detector(frame)
cropped_image = region_of_interest(canny_image)

lines = cv2.HoughLinesP(cropped_image, 2, np.pi / 180, 100,
np.array([]), minLineLength=40,
maxLineGap=5)

averaged_lines = average_slope_intercept(frame, lines)
line_image = display_lines(frame, averaged_lines)
combo_image = cv2.addWeighted(frame, 0.8, line_image, 1, 1)
cv2.imshow("results", combo_image)

if cv2.waitKey(0) & 0xFF == ord('q'):
break

# close the video file
cap.release()

# destroy all the windows that is currently on
cv2.destroyAllWindows()
```

원본 Image

기법1 Input Grey Image

기법2 Histogram equilization

기법3 Gaussian Smoothening

기법4 Canny Edge