

차량 간 통신을 이용한 안전 자율 주행

목차 CONTENTS

1. 프로젝트 및 구성원 소개

2. 프로젝트 과정

1. 최적화 과정
2. 4가지의 딥 러닝 모델 성능 비교
3. 상황 별 주행
4. 소스코드 설명

3. 시연 영상

4. 앞으로의 활용 방안

5. 데모 영상 & 질의응답

1 프로젝트소개

차량 간 통신을 이용한 안전 자율 주행

SBC기반의 RC카 프로그래밍

딥러닝 학습을 통한 자율 주행

통신을 사용한 안전 자율 주행

1 구성원소개

팀 명 : AICAR

20165158 임아영

20165127 박수빈

20155314 김택진

20145316 박진홍



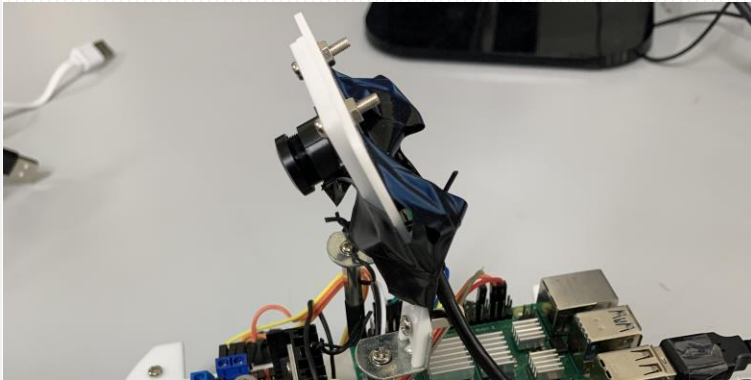
2 - 1 최적화 과정

정확한 데이터 수집과 학습을 위한 최적화 과정 필요

카메라 높이 최적화

텐서플로우 라이트 사용

OrangePi 에서 라즈베리 파이 V4로 변경 + 모터 제어 라이브러리 포팅



2 - 24가지의딥러닝모델성능비교

A 97%	Color Net (샘플로 주어진 신경망) Conv 신경망으로만 구성된 샘플(컬러 이미지)
B 93%	Gray Net (샘플로 주어진 신경망) 에서 입력 영상을 Gray Scale 로 변환 후 학습하는 모델로 변경
C 39%	Gray ANN (새로운 신경망 구현) Gray Scale 이미지를 사용, Fully-Connected Layer로만 구성, ANN 네트워크 구현
D 96%	Gray CNN (새로운 신경망 구현) Gray Scale 이미지를 사용, CNN 구조를 사용하는 네트워크 구현 (LeNet-5를 일부 변형한 형태)
E 97%	cnnKeras (새로운 신경망 구현) Gray Scale 이미지를 사용, CNN 구조를 사용하는 네트워크 구현, TF-Lite 버전 사용

```
Epoch: 29, Step: 401, Loss: 0.0846585
Train Accuracy: 0.97
Validation Accuracy: 0.95
begin: 2020-05-28_07-39-34
end: 2020-05-28_08-02-38
Run the command line:
--> tensorboard --logdir=./logs --port=6006
Then open http://0.0.0.0:6006/ into your web browser
```

2 - 2 - 1 CNN 학습모델

```
from matplotlib import pyplot as plt
%matplotlib inline
```

```
num_classes = cfg.NUM_KEYS
```

```
#x_train_images = tf.reshape(x_train_images, [len(x_train_images), cfg.final_height, cfg.final_width, 1])
x_train_images = np.array(x_train_images)
print(x_train_images.shape)
x_train_images.reshape(x_train_images.shape[0], cfg.final_height, cfg.final_width, 1)
print(x_train_images.shape)
plt.imshow(x_train_images[0])
plt.show()
```

```
#x_validation_images = tf.reshape(x_validation_images, [len(x_validation_images), cfg.final_height, cfg.final_width, 1])
x_validation_images = np.array(x_validation_images)
x_validation_images.reshape(x_validation_images.shape[0], cfg.final_height, cfg.final_width, 1)
print(x_validation_images.shape)
plt.imshow(x_validation_images[0])
plt.show()
```

```
#x_test_images = tf.reshape(x_test_images, [len(x_test_images), cfg.final_height, cfg.final_width, 1])
x_test_images = np.array(x_test_images)
x_test_images.reshape(x_test_images.shape[0], cfg.final_height, cfg.final_width, 1)
print(x_test_images.shape)
plt.imshow(x_test_images[0])
plt.show()
```

```
y_train_labels = keras.utils.to_categorical(y_train_labels, num_classes)
y_validation_labels = keras.utils.to_categorical(y_validation_labels, num_classes)
y_test_labels = keras.utils.to_categorical(y_test_labels, num_classes)
```

CNN 신경망 학습

```
model = Sequential()
```

```
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1, 1), padding='same',
                activation='relu',
                input_shape=input_shape))
```

```
#model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
```

```
model.add(MaxPooling2D())
```

```
model.add(Conv2D(64, (2, 2), activation='relu', padding='same'))
```

```
#model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(MaxPooling2D())
```

```
model.add(Dropout(0.25))
```

```
model.add(Flatten())
```

```
model.add(Dense(1000, activation='relu'))
```

```
model.add(Dropout(0.5))
```

```
model.add(Dense(num_classes, activation='softmax'))
```

```
model.summary()
```

CNN 신경망 구현

2 - 3 상황별주행

3가지 상황

1. 음성 인식
(Start, Accident, Clear, Stop)
2. 화살표 인식 (유,무)
3. 사람 인식



2 - 4 학습후 제어 코드

```

if start_flag:
    if cfg.wheel == cfg.STOP:
        hw.motor_one_speed(cfg.normal_speed_right==0)
        hw.motor_two_speed(cfg.normal_speed_left==0)
    elif cfg.wheel == cfg.LEFT: #left turn
        hw.motor_one_speed(cfg.maxturn_speed)
        hw.motor_two_speed(cfg.minturn_speed)
    elif cfg.wheel == cfg.UP:
        #hw.motor_one_speed(cfg.normal_speed_right)
        #hw.motor_two_speed(cfg.normal_speed_left)
        if slow_flag == True:
            hw.motor_one_speed(int(cfg.normal_speed_right/2))
            hw.motor_two_speed(int(cfg.normal_speed_left/2))
        else:
            hw.motor_one_speed(cfg.normal_speed_right)
            hw.motor_two_speed(cfg.normal_speed_left)
    elif cfg.wheel == cfg.RIGHT: #right turn
        hw.motor_one_speed(cfg.minturn_speed)
        hw.motor_two_speed(cfg.maxturn_speed)
    elif cfg.wheel == cfg.UP_LEFT:
        hw.motor_one_speed(cfg.normal_speed_right)
        hw.motor_two_speed(round(cfg.normal_speed_left/cfg.slow_turn_factor))
    elif cfg.wheel == cfg.UP_RIGHT: #right turn
        hw.motor_one_speed(round(cfg.normal_speed_right/cfg.slow_turn_factor))
        hw.motor_two_speed(cfg.normal_speed_left)
    else:
        assert False

else:
    hw.motor_one_speed(0)
    hw.motor_two_speed(0)
    cfg.wheel = cfg.STOP

```

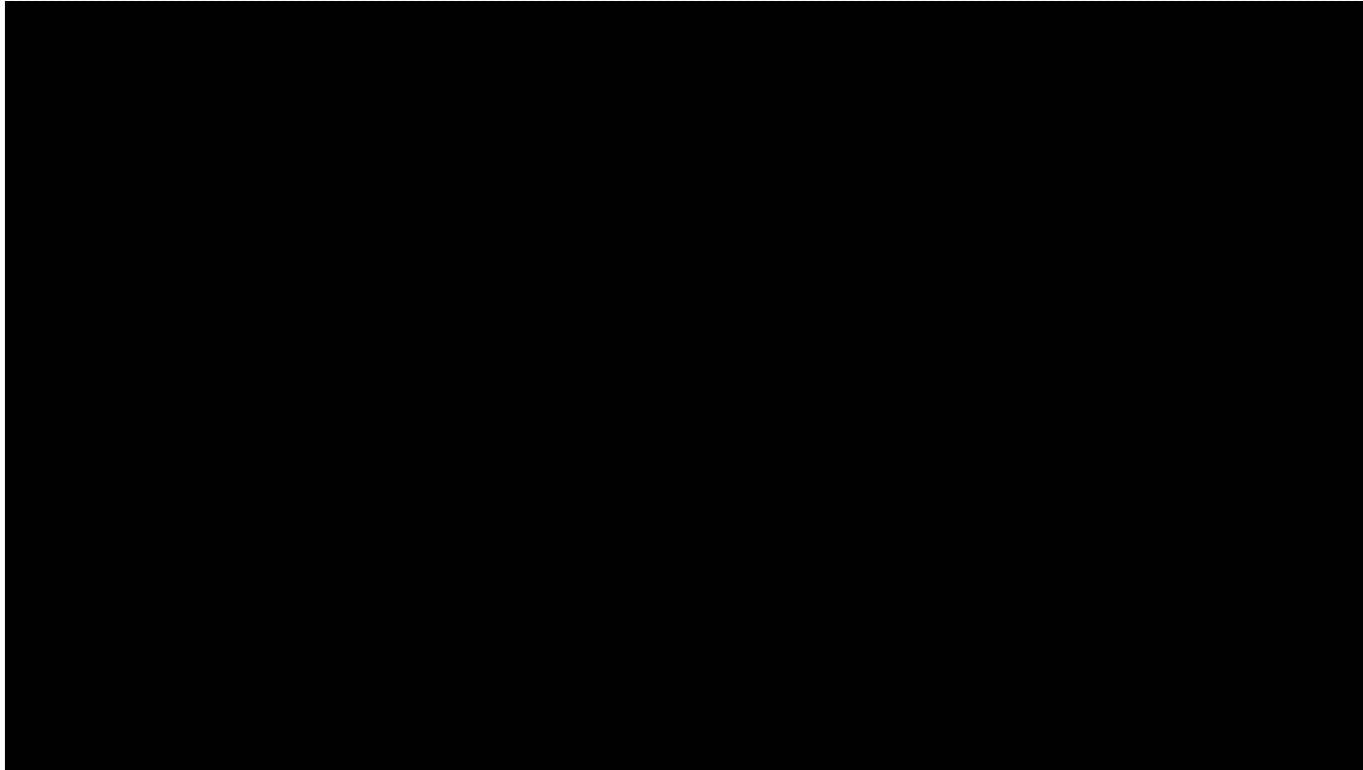
딤러닝 기반의 방향 제어

Flag를 사용하여 상황별로 다른 동작을 하도록 구현

모터 제어 라이브러리

3 - 1 음성인식(Start, Accident, Clear, Stop)을 이용한 차량제어시연

영상



3 - 2화살표인식,사람인식시연영상



← 딥러닝 기반
최적의 방향 예측

← 실시간 카메라 영상

4 활용방안



[물류창고 관리 자동화]

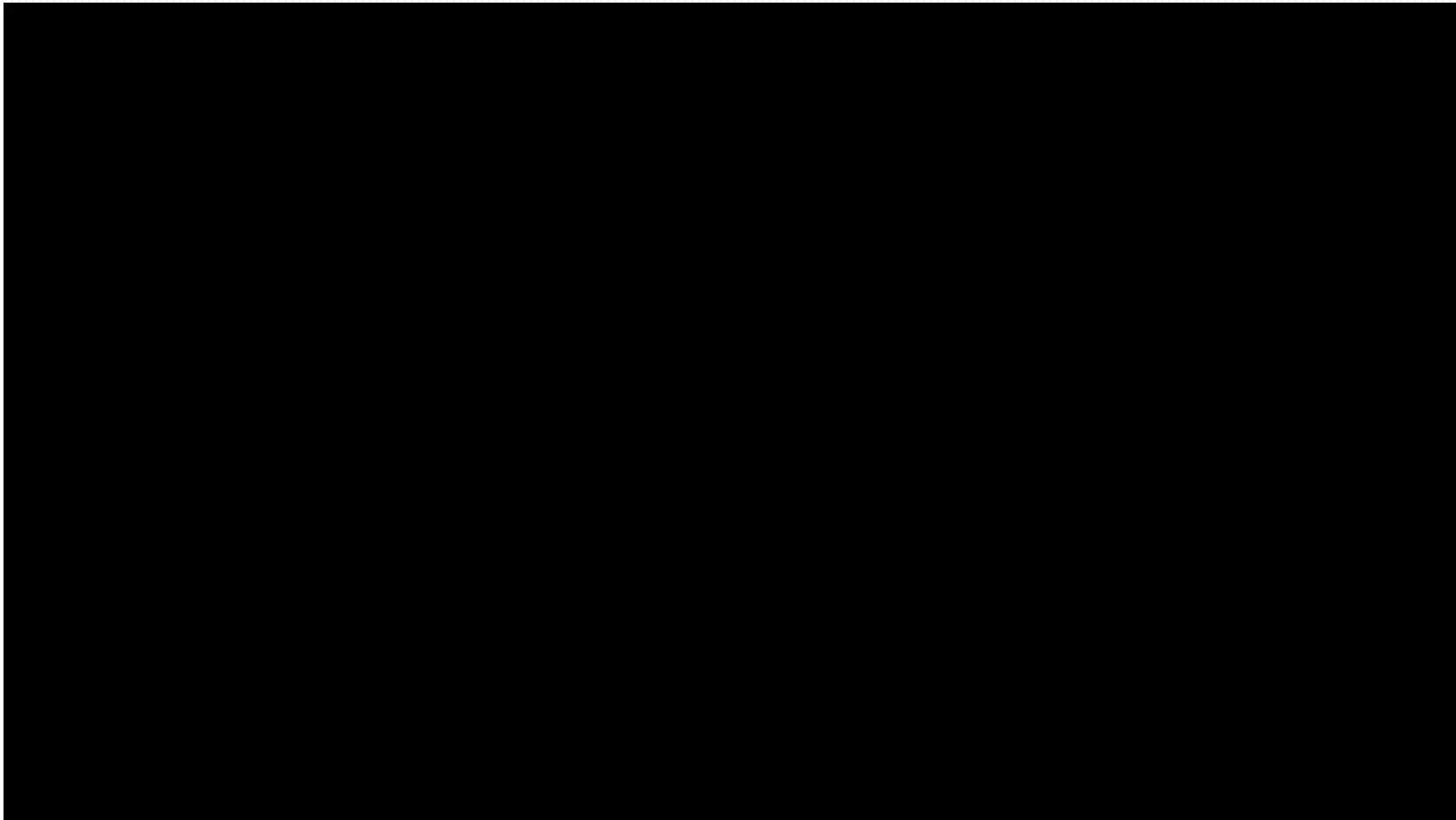


[물류배송 자동화]



[무인 대중교통]

5. 전체적인 데모 영상



질의 응답

감사합니다.