

과제유형	<input type="checkbox"/> 일반형 <input type="checkbox"/> 기업 연계형					
교과목명	지능형 시스템					
과 제 명	도로 교통 표지판 분류&인식					
팀 명						
기 간	2022.00.00 ~ 2022.00.00					
지도교수	성 명	권택원		소속(학과)	융합IT	
	연락처			E-Mail		
참여기업	성 명			기업명		
	연락처			E-Mail		
결과물유형	<input type="checkbox"/> 시제품 <input type="checkbox"/> 논문 <input type="checkbox"/> 아이디어 <input type="checkbox"/> 기타()					
구 분	순번	성 명	학 번	학 과	전 화 번 호	역 할
참여팀원	1					팀장
	2	김규영	17681005	융합IT	01050070507	팀원
	3	장정우	17681026	융합IT	01032114464	팀원
	4					팀원
	5					팀원
	6					팀원
	7					팀원
	8					팀원
<p>2022년도 2학기 캡스톤디자인 과제 운영에 따른 결과보고서를 제출합니다.</p> <p style="text-align: center;">2022년 월 일</p> <p style="text-align: right;">지도교수 : (인)</p> <p>건양대학교 3단계 산학협력 선도대학 육성사업(LINC 3.0)단장 귀하</p>						

1. 과제 수행 개요 (1P 작성 / 맑은고딕, 11pt, 줄간격 160%)

가. 추진배경

딥러닝 교과목에서 진행한 최종 프로젝트에서 진행한 '자동차 번호판'인식 프로그램과 연관시켜 '교통 표지판 분류 및 인식'을 하면 자율주행 자동차들이 교통 법규를 지키며 안전한 운행을 할 것이라는 기대를 가지고 주제를 선정 했습니다.

나. 교과목 목적 및 목표

1. **목적** 수업을 통해 지능형 시스템을 이해하여 프로젝트를 성공적으로 수행한다.
2. **목표** 수업내 학습 과정을 통해 도로 교통 표지판 분류 및 인식 프로그램을 작성한다.

※ 교과목 운영 사업계획서 참고하시어 작성해주시면 됩니다.

2. 과제 수행 과정 및 방법 (1P 작성 / 맑은고딕, 11pt, 줄간격 160%)

가. 과제 수행 과정

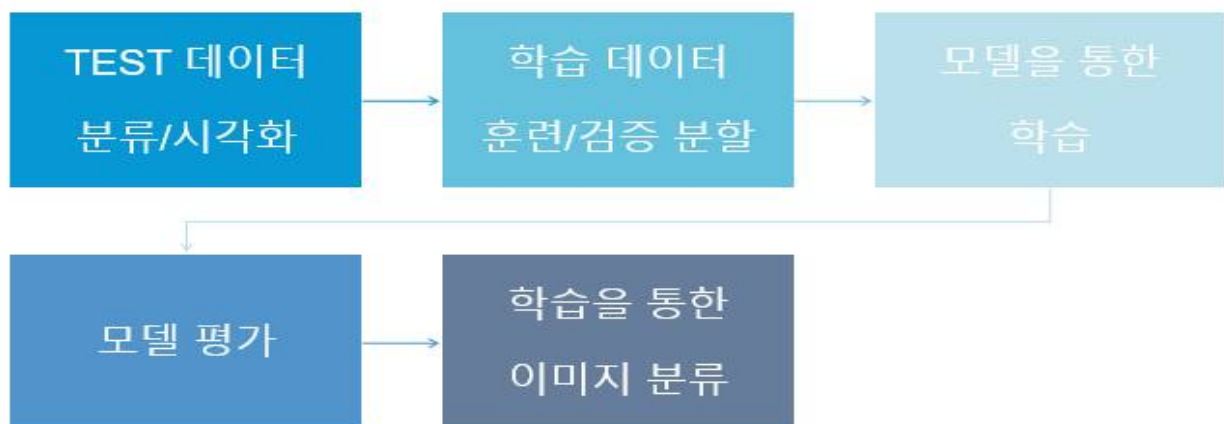
일요일	월요일	화요일	수요일	목요일	금요일	토요일
11/20	21	22	23	24	25	26
계획 수립	→		주제 선정	→		Kaggle 데이터 검색
27	28	29	30	12/1	2	3
개별 코딩 및 분석	→					
4	5	6	7	8	9	10
분석 및 평가 프로젝트 마무리	→					PPT/보고서 작성

공통: 주제 선정, 아이디어 공유, 데이터 선택

김규영 : 데이터 전 처리, 정확도 분석 시각화 PPT 작성

장정우: 데이터 학습 코드 작성, 정확도 분석, 과제 결과 보고서

나. 과제 수행 방법



3. 과제 개발 결과 (1P 작성 / 밝은고딕, 11pt, 줄간격 160%)

가. 개발 내용

```
model = keras.models.Sequential([
    keras.layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', input_shape=(224, 224, 3)),
    keras.layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu'),
    keras.layers.Conv2D(filters=128, kernel_size=(3,3), activation='relu'),
    keras.layers.MaxPool2D(pool_size=(2, 2)),
    keras.layers.BatchNormalization(axis=-1),

    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(rate=0.5),

    keras.layers.Dense(43, activation='softmax')
])

lr = 0.001
epochs = 30

opt = Adam(lr=lr, decay=lr / (epochs * 0.5))
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])

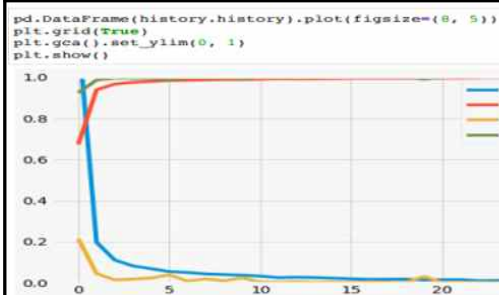
aug = ImageDataGenerator(
    rotation_range=10,
    zoom_range=0.15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.15,
    horizontal_flip=False,
    vertical_flip=False,
    fill_mode="nearest")

history = model.fit(aug.flow(X_train, y_train, batch_size=32), epochs=epochs)
```

Epoch 1/30
858/858 [=====] - 19s 23ms/step - loss: 1.218
0 - accuracy: 0.6729 - val_loss: 0.2146 - val_accuracy: 0.9243
Epoch 2/30
858/858 [=====] - 18s 21ms/step - loss: 0.197
4 - accuracy: 0.9384 - val_loss: 0.0455 - val_accuracy: 0.9856
Epoch 3/30
858/858 [=====] - 21s 24ms/step - loss: 0.111
3 - accuracy: 0.9662 - val_loss: 0.0144 - val_accuracy: 0.9953
Epoch 4/30
858/858 [=====] - 20s 23ms/step - loss: 0.082
3 - accuracy: 0.9746 - val_loss: 0.0179 - val_accuracy: 0.9943
Epoch 5/30
858/858 [=====] - 19s 22ms/step - loss: 0.068
9 - accuracy: 0.9795 - val_loss: 0.0235 - val_accuracy: 0.9934

cnn모델 구축 및 학습을 통해
교통표지판을 분류하도록 설계

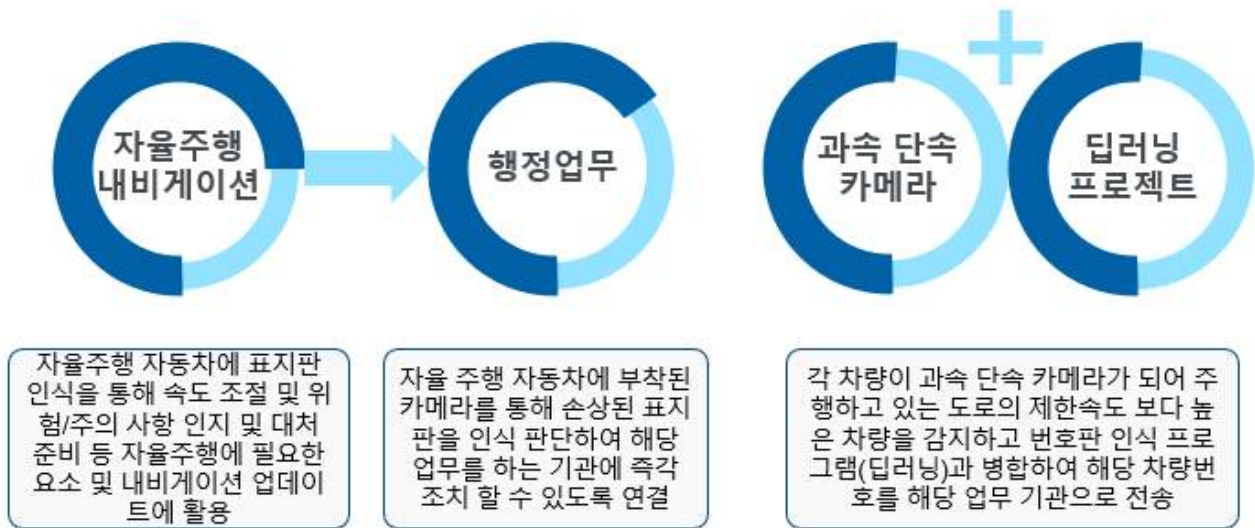
나. 성과 창출 내용



최종 무작위 교통표지판의 정답과 학습을 통해 도출한 결과값이 일치. 정확도는 99.5%로 상당히 높은 편이다.

4. 활용방안 및 기대효과 (1P 이내 작성 / 맑은고딕, 11pt, 줄간격 160%)

가. 개발 과제 활용 및 파급효과



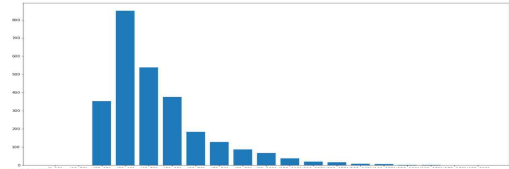
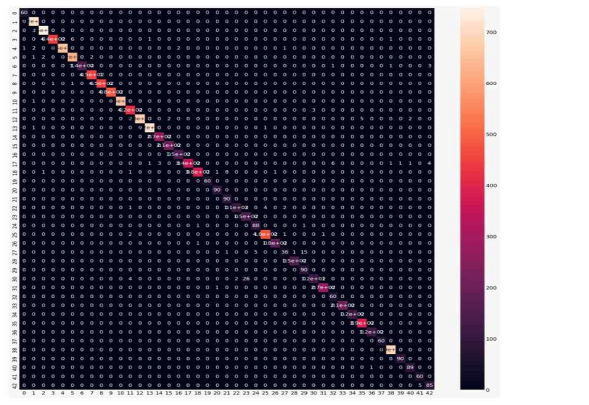

나. 지식재산권 *지식재산권 (특허, 실용신안, 상표, 디자인 등)이 있는 경우 기재*

지식재산권명(특허명 등)	출원인	출원일	출원번호
		20XX.XX.XX	00-0000-0000000

※ 작성 시 유의사항

- ① 서식통일: 맑은고딕 11pt, 줄간격 160%, 자간0, 글자색 검정
- ② **서술식**으로 작성해야 하며 수행한 과제 내용을 상세히 기재할 것
- ③ 전문용어의 경우 각주를 달아 부가 설명할 것을 권장
- ④ 작성한 내용은 지도교수님의 검토를 받아 제출할 것
- ⑤ 파일 제출 시 유의 사항 삭제 후 제출 요망
- ⑥ 기업 연계형의 경우 '상호협력협약서' 반드시 첨부 후 제출

부록 (과제 수행 및 제작 과정, 결과물)

<pre>In [69]: df_cutwidth = pd.cut(df_Train['Width'], np.arange(0,200,10)).value_counts(sort=False) fig, ax = plt.subplots(figsize=(20,10)) ax.bar(range(len(df_cutwidth)),df_cutwidth.values) ax.set_xticks(range(len(df_cutwidth))) ax.set_xticklabels(df_cutwidth.index) fig.show()</pre>  <pre>In [2]: data_dir = '../input/gtsrb-german-traffic-sign' train_path = '../input/gtsrb-german-traffic-sign/Train' test_path = '../input/gtsrb-german-traffic-sign/' # Resizing the images to 30x30x3 IMG_HEIGHT = 30 IMG_WIDTH = 30 channels = 3</pre>	<pre># Label Overview classes = (0:'Speed limit (20km/h)', 1:'Speed limit (30km/h)', 2:'Speed limit (50km/h)', 3:'Speed limit (60km/h)', 4:'Speed limit (70km/h)', 5:'Speed limit (80km/h)', 6:'End of speed limit (80km/h)', 7:'Speed limit (100km/h)', 8:'Speed limit (120km/h)', 9:'No passing', 10:'No passing veh over 3.5 tons', 11:'Right-of-way at intersection', 12:'Priority road', 13:'Yield', 14:'Stop', 15:'No vehicles', 16:'Veh > 3.5 tons prohibited', 17:'No entry', 18:'General caution', 19:'Dangerous curve left', 20:'Dangerous curve right', 21:'Double curve', 22:'Bumpy road', 23:'Slippery road', 24:'Road narrows on the right', 25:'Road work', 26:'Traffic signals', 27:'Pedestrians', 28:'Children crossing', 29:'Bicycles crossing', 30:'Beware of ice/snow', 31:'Wild animals crossing', 32:'End speed + passing limits', 33:'Turn right ahead', 34:'Turn left ahead', 35:'Ahead only', 36:'Go straight or right', 37:'Go straight or left', 38:'Keep right', 39:'Keep left', 40:'Roundabout mandatory', 41:'End of no passing', 42:'End no passing veh > 3.5 tons')</pre>
<p>이미지 크기 분포 확인 후 30 사이즈로 데이터 정렬</p>	<p>클래스 번호를 사전의 교통 표지판 이름에 매핑</p>
<pre>In [95]: lr = 0.001 epochs = 30 opt = Adam(lr=lr, decay=lr / (epochs * 0.5)) model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy']) /Users/dinogos/opt/anaconda3/lib/python3.9/site-packages/keras/optimizers/optimizer_v2/adam.py:110: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead. super(Adam, self)._init__(name, **kwargs)</pre>	<pre>In [134]: pd.DataFrame(history.history).plot(figsize=(8, 5)) plt.grid(True) plt.gca().set_ylim(0, 1) plt.show()</pre> 
<p>'Soft max', 'categorical_crossentropy' 다중분류 손실함수를 이용하여 30번의 학습</p>	<p>정확도와 손실도</p>
<pre>In [155]: import seaborn as sns df_cm = pd.DataFrame(cf, index = classes, columns = classes) plt.figure(figsize = (20,20)) sns.heatmap(df_cm, annot=True)</pre> <pre>Out[155]: <AxesSubplot></pre> 	
<p>confusion matrix</p>	<p>정확도 분류</p>