포트폴리오

[최은규]

[010-9772-3004 / dlarhkd1211@naver.com]

Index

- Personal Profile
- Projects
  - AI 농산물 선별기
  - 마포 WIFI 신규 입지 선정
  - 기상요인에 따른 질병별 발병 위험도 예측
- Study
  - 1-Stage Detector YOLO
  - Pytorch
  - OpenCV
- In Progress
  - YOLO, OCR
  - Linux, Cloud Platform (Docker)

Personal Profile

Name 최은규

Birth 1994.12.11

Phone 010-9772-3004

E-mail dlarhkd1211@naver.com

Education

서울시립대학교 환경공학/빅데이터분석학 졸업
데이터산업진흥원 '데이터 사이언스 기반 지능 소프트웨어' 과정 수료

Certificate

ADsP
SQLD
정보처리기사
TOEIC SPEAKING 150
TOEIC 865

Program Language

Python, R

Activity

(사)인공지능연구소 AI 오픈랩 'Real-Time Object Detection'
마포형 청년 일자리사업 4차산업 빅데이터팀
서울시립대학교 창업동아리 'AI 농산물 분류기' 개발

Projects. 1 AI 농산물 선별기(1)

학습내용

데이터를 직접 수집 후 학습의 용이성 및 정확도 향상을 위한 Data Augmentation

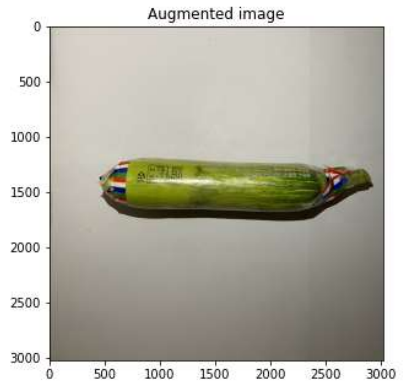
실습내용

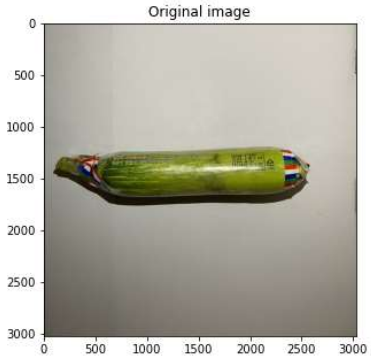
Python을 이용한 Data Augmentation

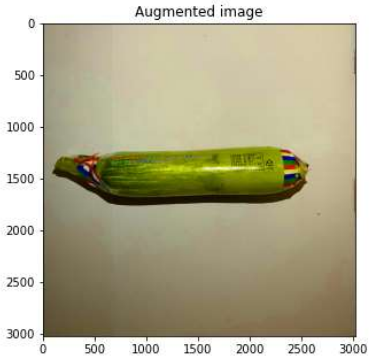
- Tensorflow.image
- 전환, 회전, 1채널화(GrayScale), 광도, 채도

결과물

```
def augment(res_img):
    image_string = tf.io.read_file(res_img)
    image = tf.image.decode_jpeg(image_string, channels = 3)

    flipped = tf.image.flip_left_right(image)
    write_tf(flipped, res_img, "flipped")

    rotated = tf.image.rot90(image)
    write_tf(rotated, res_img, "rotated")

    grayscaled = tf.image.rgb_to_grayscale(image)
    write_tf(grayscaled, res_img, "grayscaled")

    saturated = tf.image.adjust_saturation(image, 3)
    write_tf(saturated, res_img, "saturated")

    bright = tf.image.adjust_brightness(image, 0.4)
    write_tf(bright, res_img, "bright")
```







Projects. 1 AI 농산물 선별기(2)

학습내용

농산물의 등급 분류를 위한 CNN 모델 생성

- 특, 1특, 2특, 불량

실습내용

Keras CNN Model

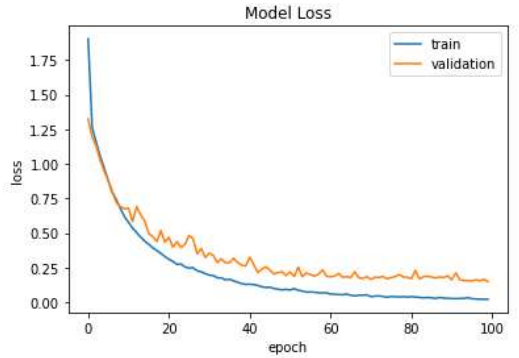
- Keras, Tensorflow
- Accuracy: 95.94

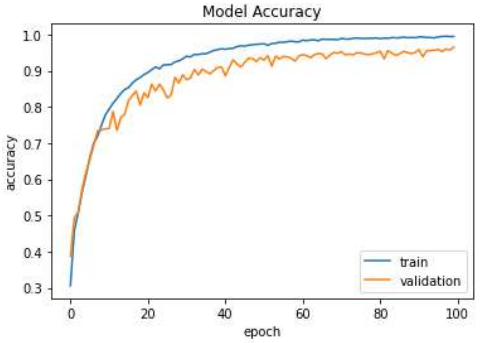
결과물

```
model = keras.Sequential([
    Conv2D(64, kernel_size = (3, 3), padding = 'same', input_shape = input_shape, activation = tf.nn.relu),
    MaxPooling2D(pool_size = (2, 2)),
    Dropout(0.05),

    Conv2D(32, kernel_size = (3, 3), padding = 'same', activation = tf.nn.relu),
    MaxPooling2D(pool_size = (2, 2)),
    Dropout(0.350000000000000003),

    Flatten(),
    Dense(352, activation = tf.nn.relu),
    Dropout(0.150000000000000002),
    Dense(num_classes, activation = tf.nn.softmax)
])
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 256, 256, 64)      1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 128, 64)      0
_________________________________________________________________
dropout (Dropout)            (None, 128, 128, 64)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 128, 32)      18464
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 64, 64, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 131072)            0
_________________________________________________________________
dense (Dense)                (None, 352)               46137696
_________________________________________________________________
dropout_2 (Dropout)          (None, 352)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 1412
=================================================================
Total params: 46,159,364
Trainable params: 46,159,364
Non-trainable params: 0
```





Projects. 1 AI 농산물 선별기(3)

학습내용

실시간 영상 내 Object(Zucchini) Detection을 위한
YOLOv3 Model training

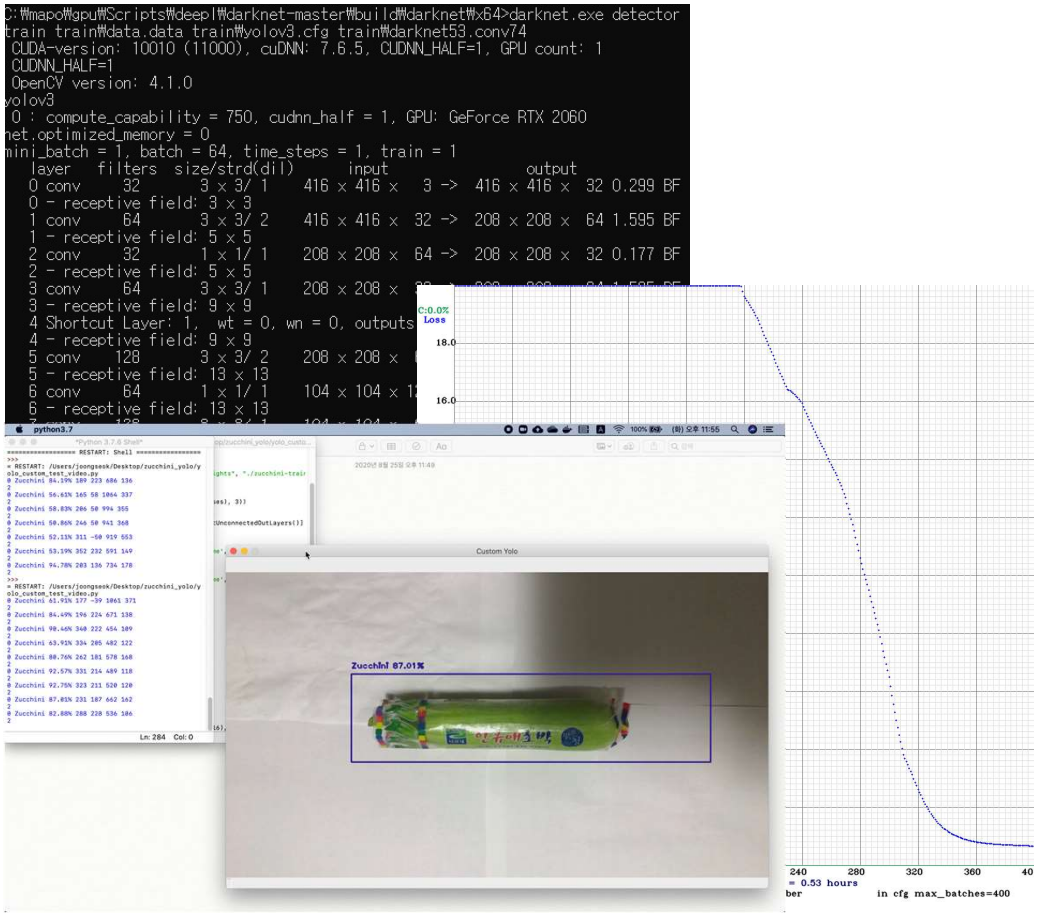
실습내용

Yolov3
- Class: 1 (Zucchini)

Final Result

Object Detection(YOLO)
-> Frame Capture(OpenCV) -> Classification(Keras)

결과물



Projects. 2 마포 WIFI 신규 입지 선정 (1)

학습내용

주소를 위도 및 경도로, 위도 및 경도를 주소로 변환하는 Geocoding

실습내용

Open API를 이용하여 각 주소에 대한 위도, 경도값 크롤링
- API World

결과물

```
for i in tqdm.trange(len(data)):
    n = data['name'][i]
    add = 'http://api.vworld.kr/req/address?service=address&request=getc
mat=json&type=road&key=DAEEBC94-6866-34E2-BEC6-27D059804F75'
    r = requests.get(add)
    #print(float(r.json()['response']['result']['point']['x']),r.json()[
    data['y'][i] = float(r.json()['response']['result']['point']['x'])
    data['x'][i] = float(r.json()['response']['result']['point']['y'])
```

	Unnamed: 0	code	populations	name	x	y	X	Y
0	0	1000458	1164943	광성로6길	37.549513	126.936671	194578	449861
1	1	1000459	1482185	대흥로21길	37.545634	126.938695	194970	450521
2	2	1000460	1115615	대흥로28길	37.547106	126.941870	195318	450623
3	3	1000461	703465	도화4길	37.537972	126.950394	195659	448858
4	4	1000462	1726932	도화길	37.541531	126.949875	195575	449041

Projects. 2 마포 WIFI 신규 입지 선정 (2)

학습내용

각 요인들에 대해 5점척도로 변환.

실습내용

주용도코드: 상권에 우선순위가 있기 때문에
상권지역 5, 주거지역 2.5
인구: QGIS Natural Break Value 활용
Wifi 설치여부에 따른 0, 1 부여

결과물

```
for i in range(len(df2)):
    if df2['lbl'][i] == None:
        df2['pop'][i] += 0
    elif df2['lbl'][i] == '01000' or df2['lbl'][i] == '02000':
        df2['pop'][i] += 2.5
    elif df2['lbl'][i] == '03000' or df2['lbl'][i] == '04000':
        df2['pop'][i] += 5

    if 0 < df2['val'][i] < 115:
        df2['pop'][i] += 1
    elif 115 <= df2['val'][i] < 252:
        df2['pop'][i] += 2
    elif 252 <= df2['val'][i] < 408:
        df2['pop'][i] += 3
    elif 408 <= df2['val'][i] < 644:
        df2['pop'][i] += 4
    elif 644 <= df2['val'][i] < 1212:
        df2['pop'][i] += 5
    else:
        df2['pop'][i] += 0
```

	index	lbl	geometry	val	pop	wifi	f_score
1774	1774	04000	POLYGON ((949000.000 1950600.000, 949000.000 1...	280.0	8.0	0	0.0
1775	1775	04000	POLYGON ((949000.000 1950700.000, 949000.000 1...	23.0	6.0	0	0.0

Projects. 2 마포 WIFI 신규 입지 선정 (3)

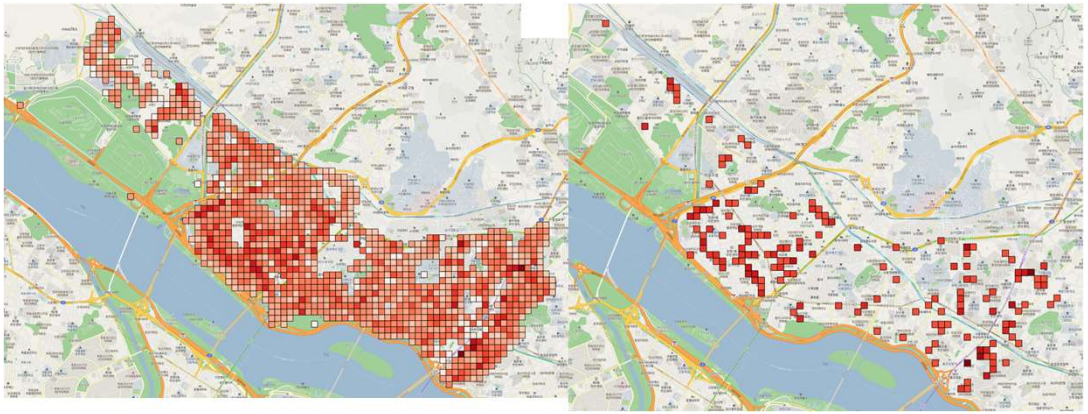
학습내용

가중치들의 합을 통해 우선 설치 지역 도출

실습내용

마포구 전체를 100m 격자로 나누고, 각 격자 별 중요도 매핑
상위 150곳을 QGIS를 통해 시각화하고,
Geocoding하여 주소값을 구청에 제공

결과물



	index	lbl	val	popu	pop	wifi	f_score	address	group
2504	2504	04000	378.0	4.0	12.0	1	12.0	대한민국 서울특별시 마포구 아현동 420-1	주요 상권
2265	2265	03000	0.0	7.0	12.0	1	12.0	대한민국 서울특별시 마포구 도화동 250-10	주요 상권
2314	2314	04000	272.0	3.0	11.0	1	11.0	대한민국 서울특별시 마포구 도화동 548	주요 상권
2283	2283	01000	256.0	5.0	10.5	1	10.5	대한민국 서울특별시 마포구 염리동 9-43	주요 상권
1328	1328	03000	104.0	4.0	10.0	1	10.0	대한민국 서울특별시 마포구 망원2동 482-5	주요 상권

Projects. 3 기상요인에 따른 질병별 발병 위험도 예측

학습내용

기상 요인(기온, 강수량, 습도 등) 및
대기 상태(PM10, PM2.5, O3, CO2 등) 에 따라
노인(65세 이상)들의 질병별 진료 횟수를 분석하고 이를
학습하여 위험도를 나타낼 수 있는 모델 개발 및 웹 서비스

실습내용

각 날짜별 기상요인, 대기상태 및 질병 종류별 진료 건수를
하나의 데이터 column에 생성.
이를 RandomForest를 이용하여 모델을 만들고,
Shinyapps를 이용하여 Web으로 서비스 제공

결과물

Study. 1 1-Stage Detector YOLO

학습내용

Object Detection 기법 중 실시간 객체탐지 모델인
YOLO에 대한 논문 리뷰

실습내용

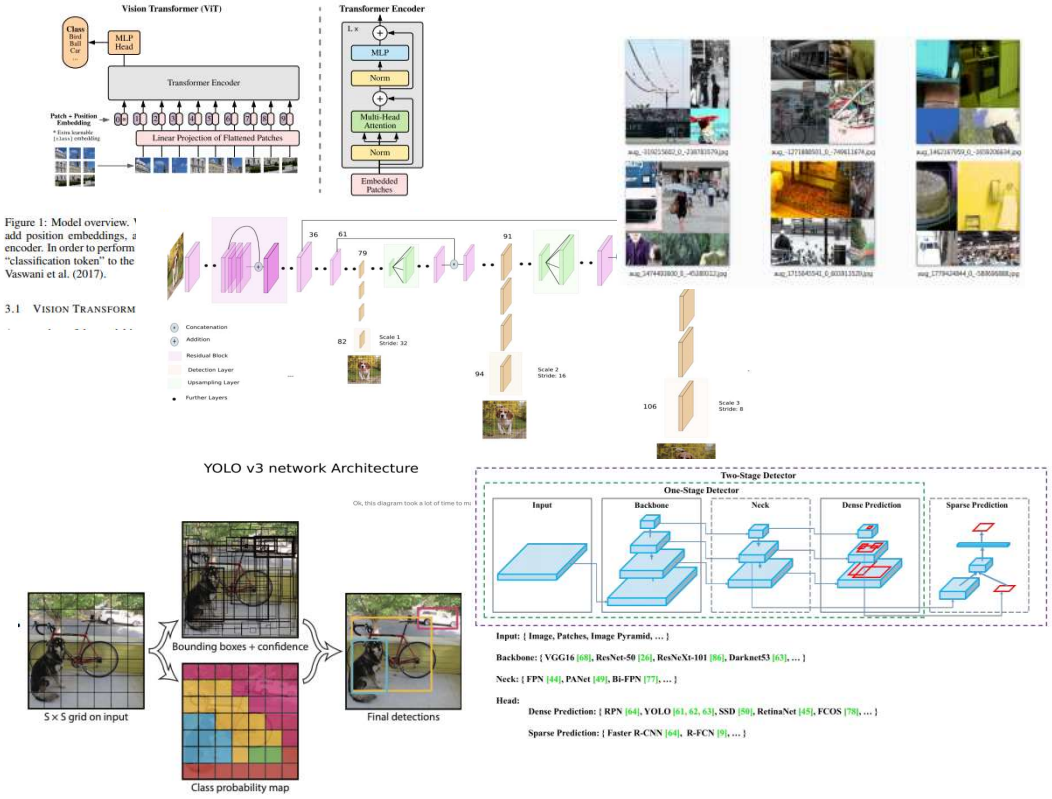
You Only Look Once: Unified, Real-Time Object Detection (https://arxiv.org/abs/1506.02640)

YOLO9000: Better, Faster, Stronger (https://arxiv.org/abs/1612.08242)

YOLOv3: An Incremental Improvement (https://arxiv.org/abs/1804.02767)

YOLOv4: Optimal Speed and Accuracy of Object Detection (https://arxiv.org/abs/2004.10934)

결과물



Study. 2 Pytorch

학습내용

Deeplearning Framework인 Pytorch를 이용하여
딥러닝의 전반적인 내용 학습

실습내용

Linear Regression, ANN
CNN Model(VGG16, GoogLeNet, ResNet)
Overfitting

결과물

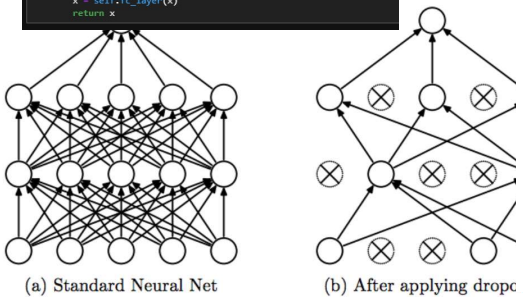
```python
def conv_2_block(in_dim, out_dim):
    model = nn.Sequential(
        nn.Conv2d(in_dim, out_dim, kernel_size = 3, padding = 1),
        nn.ReLU(),
        nn.Conv2d(out_dim, out_dim, kernel_size = 3, padding = 1),
        nn.ReLU(),
        nn.MaxPool2d(2,2)
    )
    return model

def conv_3_block(in_dim, out_dim):
    model = nn.Sequential(
        nn.Conv2d(in_dim, out_dim, kernel_size = 3, padding = 1),
        nn.ReLU(),
        nn.Conv2d(out_dim, out_dim, kernel_size = 3, padding = 1),
        nn.ReLU(),
        nn.Conv2d(out_dim, out_dim, kernel_size = 3, padding = 1),
        nn.ReLU(),
        nn.MaxPool2d(2,2)
    )
```

```python
class VGG(nn.Module):
    def __init__(self, base_dim, num_classes = 2):
        super(VGG, self).__init__()
        self.feature = nn.Sequential(
            conv_2_block(3, base_dim),
            conv_2_block(base_dim, 2 * base_dim),
            conv_3_block(2 * base_dim, 4 * base_dim),
            conv_3_block(4 * base_dim, 8 * base_dim),
            conv_3_block(8 * base_dim, 8 * base_dim)

        )

        self.fc_layer = nn.Sequential(
            nn.Linear(8 * base_dim * 7 * 7, 100),
            nn.ReLU(True),
            nn.Linear(100, 20),
            nn.ReLU(True),
            nn.Linear(20, num_classes)
        )

    def forward(self, x):
        x = self.feature(x)
        x = x.view(x.size(0),-1)
        x = self.fc_layer(x)
        return x
```

```python
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.layer = nn.Sequential(
            nn.Conv2d(1,16,5),
            nn.ReLU(),
            nn.Conv2d(16,32,5),
            nn.ReLU(),
            nn.MaxPool2d(2,2),
            nn.Conv2d(32,64,5),
            nn.ReLU(),
            nn.MaxPool2d(2,2)
        )
        self.fc_layer = nn.Sequential(
        nn.Linear(64*3*3,100),
        nn.ReLU(),
        nn.Linear(100, 10)
        )

    def forward(self, x):
        out = self.layer(x)
        out = out.view(batch_size, -1)
        out = self.fc_layer(out)
        return out
```

```python
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
model = CNN().to(device)
loss_func = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr = learning_rate)
```

```python
loss_arr = []
for i in range(num_epoch):
    for j,[image, label] in enumerate(train_loader):

        x = image.to(device)
        y_ = label.to(device)

        optimizer.zero_grad()
```



(a) Standard Neural Net (b) After applying dropo

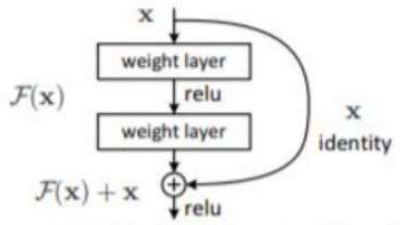


Figure 2. Residual learning: a building block.

Study. 3 OpenCV

학습내용

실시간 컴퓨터비전 라이브러리인 OpenCV

Image 처리 및 라이브러리를 활용한 프로그램 실습

YOLO weight 파일을 이용한 객체 탐지

실습내용

Image Augmentation (Resize, Flip, Rotate etc)

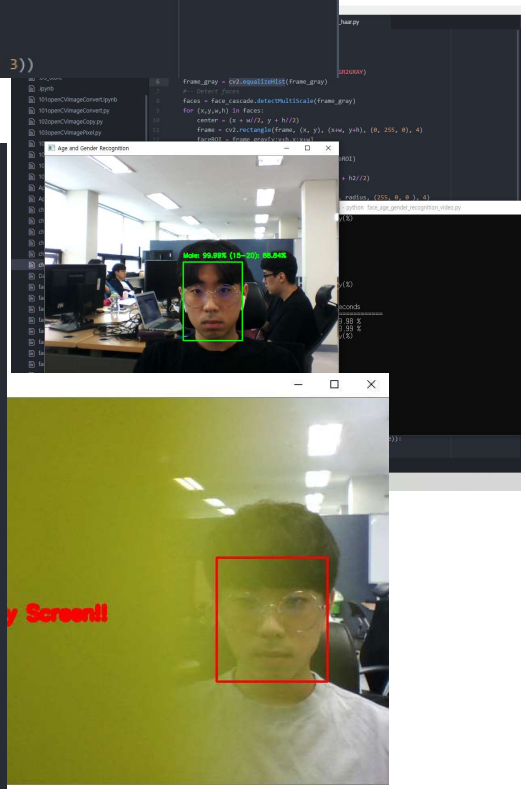
Face Detection Model 실습

- 특정 위치에 있는 얼굴만 인식하여 저장하는 프로그램 개발

결과물

```python
# Load Yolo
net = cv2.dnn.readNet("./model/custom-train-yolo_final.weights", "./custom/custom-train-yolo.cfg")
classes = []
with open("./custom/classes.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
print(classes)
color_lists = np.random.uniform(0, 255, size=(len(classes), 3))
```

```python
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > min_confidence:
            print(detection)
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)

            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            names.append(classes[class_id])
            colors.append(color_lists[class_id])

indexes = cv2.dnn.NMSBoxes(boxes, confidences, min_confidence, 0.4)
font = cv2.FONT_HERSHEY_PLAIN
for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = '{} {:,.2%}'.format(names[i], confidences[i])
        color = colors[i]
        print(i, label, color, x, y, w, h)
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        cv2.putText(img, label, (x, y - 10), font, 1, color, 2)
```



In Progress 1 YOLO, OCR

학습내용

YOLOv4를 이용하여 차량을 인식하고,
인식된 차량의 번호판에서 차량 번호를 인식하는
프로그램 개발

실습내용

- OpenCV를 이용하여 GrayScale 변환 후
  Threshold값을 이용하여 이미지 이원화
- 글자 인식에 용이하도록 이미지 변환 후 Tesseract를
  이용하여 이미지 내 텍스트 추출

결과물

In Progress 2 Linux, Cloud Platform(Docker)

학습내용

컨테이너 기반의 오픈소스 가상화 플랫폼 Docker 학습

개발 및 분석 환경에서 주로 사용하는 OS인 Linux 학습

실습내용

Docker를 이용한 Ubuntu, Centos 등 container 생성

- Oracle 및 MySQL 설치

Linux 명령어 학습

결과물