

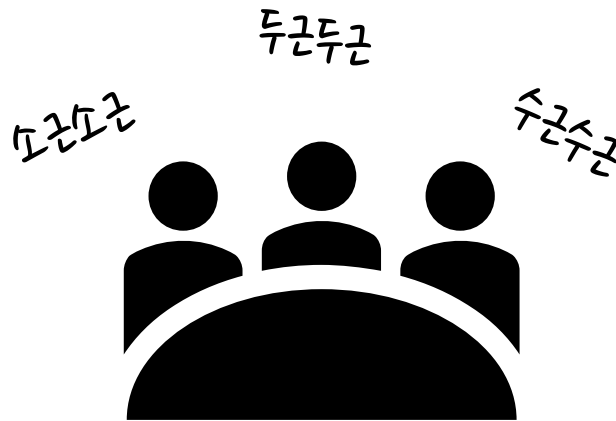
다양한 오픈소스 활용법

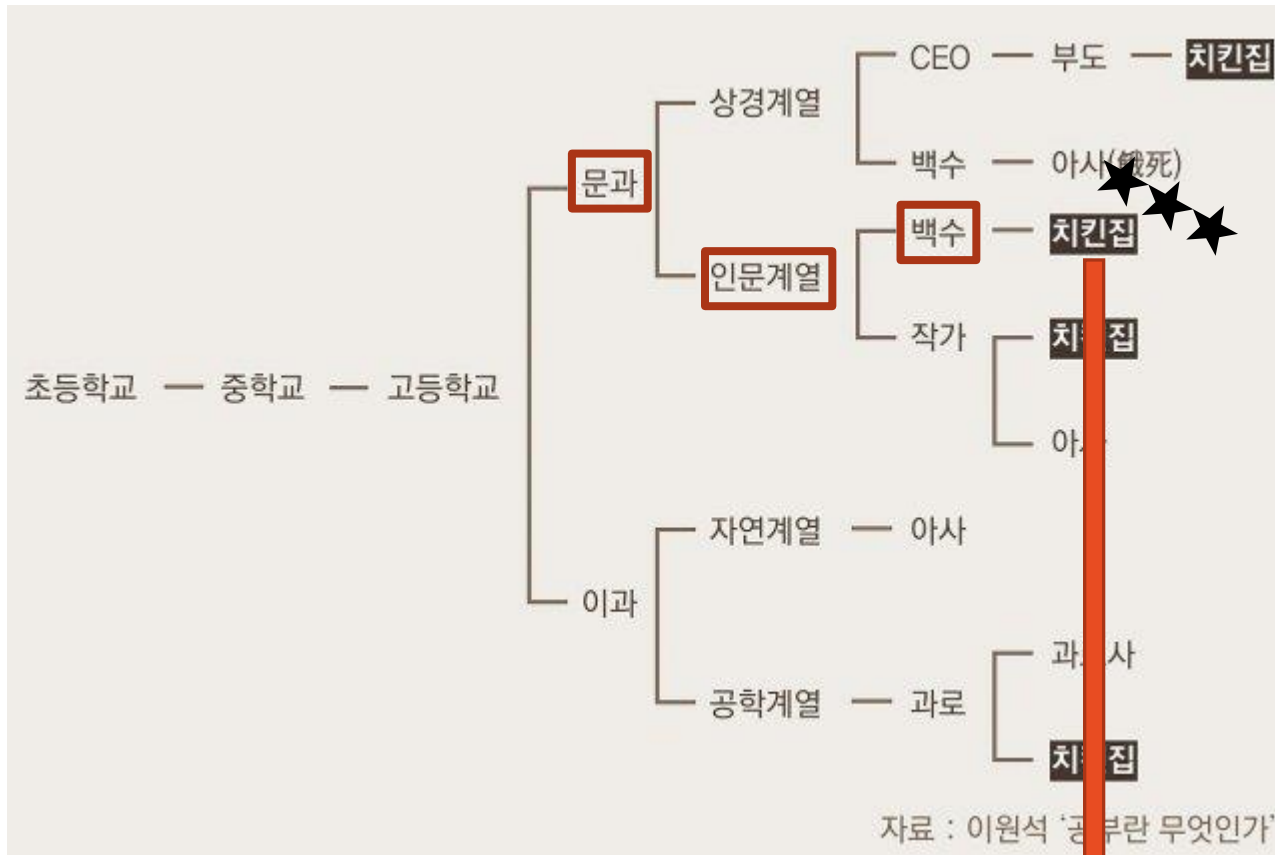
(주)꿀비 빅데이터팀 안수현

an.suhyun927@gmail.com

<https://github.com/yagippi27>

첫 번째로 든 생각은?





이과 : 졸업 → 취직 → 해고 → 치킨

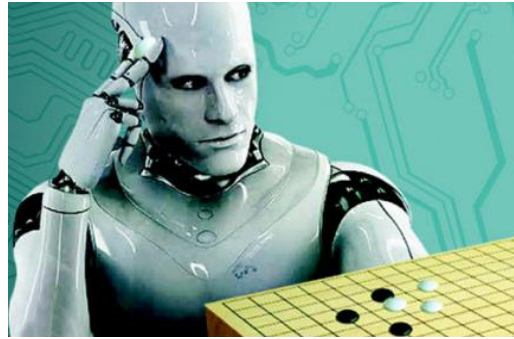
문과 : 졸업 → 빠른 치킨

빠른 스타트로 노하우가 다름

충남대 사회학과라는
치킨집의 운명이구나



전필	기초사회통계	3.0	A0
전선	사회통계분석	3.0	A+
전필	사회조사실습1	3.0	A+
전필	사회조사실습2	3.0	A+



추천서

성명 : 안수현
생년월일 :

상기인은 과학기술정보통신부와 정보통신기획평가원이 주관하고 한국생산성본부에서 운영하고 있는 국가 혁신성장 분야에 고급기술인재 양성을 목적으로 기획된 '빅데이터를 활용한 스마트데이터 전문가 양성과정'에 참여하여, 높은 교육성과를 창출하고 우수 역량 보유자로 판단되어 귀사에 관련 분야 전문가로 추천드립니다.

- 주관기관: 과학기술정보통신부, 정보통신기획평가원
- 수행기관: 한국생산성본부
- 관련사업: 혁신성장 청년인재 집중양성사업
- 참여과정: 빅데이터를 활용한 스마트데이터 전문가 양성과정
- 교육내용: R, Python, 머신러닝, 딥러닝 기술 교육 및 산업맞춤형 데이터 분석 프로젝트 진행
- 교육기간: 2019년 5월 27일 (월) ~ 2019년 11월 8일 (금) [총 960시간]
- 문의처: 한국생산성본부 디지털융복합센터 (02-724-1224)

2019년 11월 4일

한국생산성본부 회장

본부 회장

이티를 이용한 전력수요 예측
119.11.08.

가원 및 한국생산성본부에서
'청년인재 집중양성' 과정인
'스마트데이터 전문가 양성과정'
에 같이 입성하였기에
수여합니다.

1월 08일

=

GGulb.



오픈소스에 기여하는 방법

오픈소스에 기여하고 싶으세요? 초보자와 숙련자를 위한 오픈소스 기여 가이드입니다.



오픈소스 프로젝트 시작하기

오픈소스의 세계에 대해 자세히 알아보고 여러분의 프로젝트를 시작할 준비를 해보세요.



프로젝트에 기여할 사람 찾기

당신의 오픈소스 프로젝트가 행복한 사용자들의 손길 아래 성장할 수 있게 만드세요.



마음을 끄는 커뮤니티 만들기

사람들이 프로젝트를 사용하고, 기여하고, 전파하도록 격려하는 커뮤니티를 만드세요.



강의 개요

Colab 기반으로
영상 처리
오픈소스
사용 경험 쌓기

강의 목표



25분 x 2강

강의 길이



Python 기초 코딩
머신러닝
workflow

선행 지식



영상 처리에 대한
이해를 높이고자
하는 사람

학습 대상



강의 범위

1강

- 오픈 소스란?
- Git 그리고 Github
- 저작권에 대하여
- Colab 실습
- 2강을 위한 준비

2강

- 비식별화란?
- Dlib-models 설명
- License-plate Detection by YOLO
- 안면 비식별화 코딩
- 자동차 번호판 비식별화 코딩

오픈 소스를 위한 준비하기

오픈소스 활용 - Google 검색

google.com/search?newwindow=1&sxsrf=ALeKk02gcPWkDWA7Wer-B7OKCd7kyAxaA%3A1602090816194&ei=QPd9X-euC8ml

Google

오픈소스 활용

전체 뉴스 이미지 동영상 지도 더보기

검색결과 약 10,200,000개 (0.50초)

medium.com > what-learn > git-사용법-3-오픈소스와-... ▼

GIT 사용법 — (3) 오픈소스와 활용 방안 - Medium

2019. 3. 28. — 오픈 소스(Open Source)란 — '공개된 소스코드'를 의미한다. 특정한 소프트웨어를 개발한 개발자의 권리를 지키면서 누구나 소스코드를 확인할 수 ...

www.zdnet.co.kr > view ▼

[공개SW 활용 성공 사례/네이버] "오픈소스 사용 20년...모든 ...

2019. 10. 16. — 14일 성남시 분당구 정자동 본사에서 만난 김성관 네이버 책임리더(이사)는 "오픈소스를 잘 활용하기 위한 선순환 구조를 만들어가는 중"이라며 이 ...

platum.kr > archives ▼

오픈소스 활용 전에 고려해야 할 사항들 - 'Startup's Story Platform'

2015. 7. 16. — 요즘은 새로 만들기 보다는 기존에 있던 솔루션을 기반으로 커스터마이징, 혹은 업그레이드를 하든지 아니면 오픈소스를 이용하는 경우가 많은 듯 ...

www.ciokorea.com > news ▼

기업이 오픈소스를 좋아하는 10가지 이유 - CIO Korea

2019. 7. 9. — 2019년 4월에 공개된 레드햇의 기업 오픈소스 실태 조사에 따르면 IT 리더 중 ... 적지 않은 오픈소스 소프트웨어 자산을 활용하고 있음"을 발견했다.

spri.kr > post ▼

기술 혁신을 위한 오픈소스 활용 - SPRI - 소프트웨어정책연구소 ...

2018. 10. 26. — SW 분야에서 오픈소스가 중요해지면서 이를 활용하는 다양한 혁신 사례들이 나타나고 있음; • SW산업뿐만 아니라 전 산업 관점에서 오픈소스 ...

www.oss.kr > news > show ▼

기업이 오픈소스를 좋아하는 10가지 이유 - 공개SW 포털

2019. 7. 10. — 오픈소스 소프트웨어는 느리긴 하지만 기업 영역에 확실히 침투했고 많은 ... 적지 않은 오픈소스 소프트웨어 자산을 활용하고 있음"을 발견했다.

www.oss.kr > news > show ▼

[오픈소스가 힘이다]②국내 활용 어디까지 왔나 - 공개SW 포털

소용없다 이 말이야

오픈 소스란 무엇인가?

- 오픈 소스 소프트웨어(Open Source Software, OSS)
- 공개적으로 액세스할 수 있게 설계
- 누구나 자유롭게 확인, 수정, 배포가능한 코드
- 비용절감, 코드 변형 가능, 빠른 개선 속도, 불확실한 로드맵, 기술 지원 의무 없음, 저작권 정책
- 단순히 소스 코드가 공개된 소프트웨어로 무료로 사용할 수 있는 프리웨어(Freeware)와 다른 장민제. (2018. 06. 18). [단독] 한글과컴퓨터, 오픈소스 분쟁소송 '205만 달러'로 합의. 시사위크.
- Git 전문 웹호스팅 서비스인 Github에서 무궁무진한 오픈소스를 찾을 수 있음



Git과 Github

- 소스코드 관리를 위한 분산 버전 관리 시스템
- 2005년 리누스 토발즈가 고안한 서비스
- 파일 변화 기록 및 특정 시점의 버전 불러오기
- 코드 수정자 추적 가능
- Git 전문 웹호스팅 플랫폼
- 2008년 3명의 개발자에 의해 시작
- 깃을 웹에서 편하게 쓸 수 있도록 만든 도구
- 그래픽 기반으로 서비스 직관적 이해 가능
- Github 서버 내 오픈 소스 사용 가능



라이선스에 대하여

- 오픈 소스마다 라이선스가 존재
- 각각의 오픈 소스의 라이선스를 확인하고 그에 따른 규정을 지키는 것이 좋음

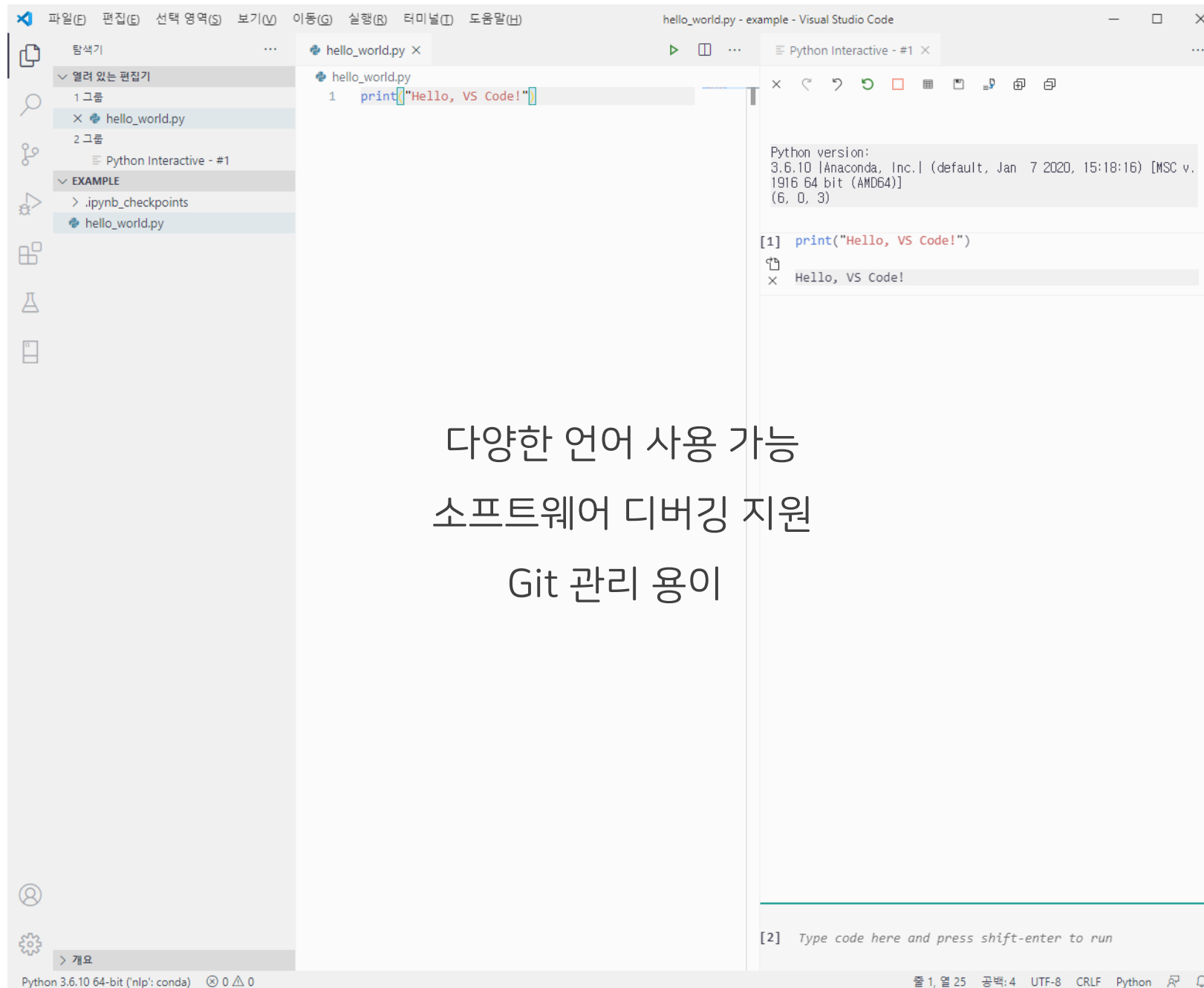
 davisking/dlib is licensed under the Boost Software License 1.0 A simple permissive license only requiring preservation of copyright and license notices for source (and not binary) distribution. Licensed works, modifications, and larger works may be distributed under different terms and without source code.	Permissions <ul style="list-style-type: none">✓ Commercial use✓ Modification✓ Distribution✓ Private use	Limitations <ul style="list-style-type: none">✗ Liability✗ Warranty	Conditions <ul style="list-style-type: none">① License and copyright notice for source
 alitourani/yolo-license-plate-detection is licensed under the GNU General Public License v3.0 Permissions of this strong copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.	Permissions <ul style="list-style-type: none">✓ Commercial use✓ Modification✓ Distribution✓ Patent use✓ Private use	Limitations <ul style="list-style-type: none">✗ Liability✗ Warranty	Conditions <ul style="list-style-type: none">① License and copyright notice① State changes① Disclose source① Same license

참고해보기 : [\[춘식이의 코드이야기\] 대표 오픈소스 라이선스, 한 눈에 보기!](#)

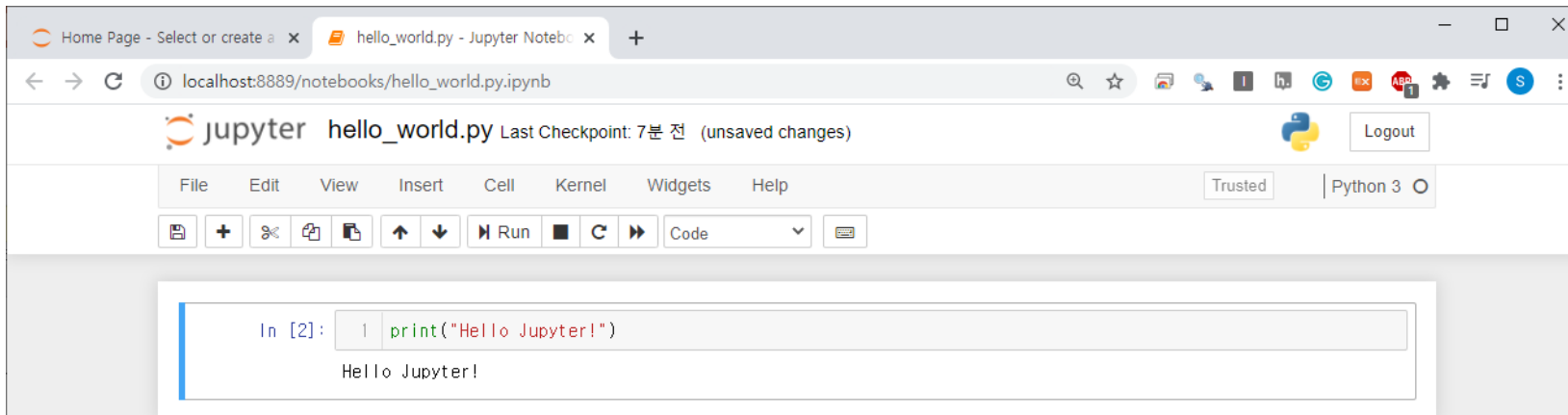
오픈 소스 사용이
쉬운 툴이 뭐가 있을까? ° ✨ (°▽°) ✨ °

Python을 이용하기 위한 여러 tool





다양한 언어 사용 가능
소프트웨어 디버깅 지원
Git 관리 용이



Home Page - Select or create x hello_world.py - Jupyter Notebook x

localhost:8889/notebooks/hello_world.py.ipynb

jupyter hello_world.py Last Checkpoint: 7분 전 (unsaved changes) Logout

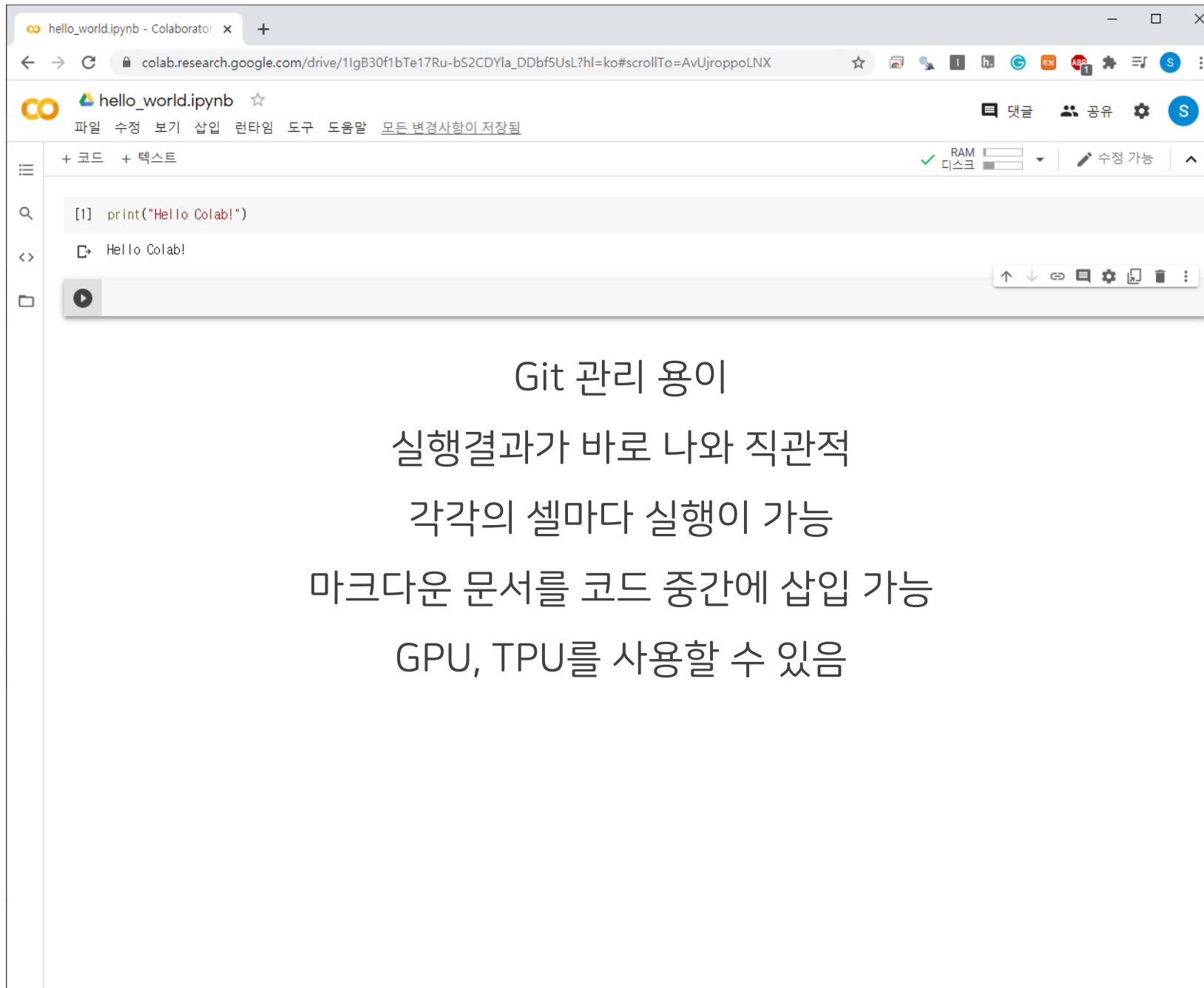
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [2]: `print("Hello Jupyter!")`

Hello Jupyter!

실행결과가 바로 나와 직관적
각각의 셀마다 실행이 가능
마크다운 문서를 코드 중간에 삽입 가능





The screenshot shows a Google Colaboratory notebook titled 'hello_world.ipynb'. The interface includes a top navigation bar with the Colab logo, file management options (파일, 수정, 보기, 삽입, 런타임, 도구, 도움말, 모든 변경사항이 저장됨), and a status bar showing RAM and disk usage. The main workspace contains a code cell with the following content:

```
[1] print("Hello Colab!")
```

Below the code cell, the output is displayed as 'Hello Colab!'. The interface also features a left sidebar with icons for file management, search, and execution. The bottom of the notebook shows a toolbar with various icons for editing and execution.

Git 관리 용이
실행결과가 바로 나와 직관적
각각의 셀마다 실행이 가능
마크다운 문서를 코드 중간에 삽입 가능
GPU, TPU를 사용할 수 있음



이제
Colab으로 가봅시다~ ٩(' 〇 ')و

링크: [HTTPS://COLAB.RESEARCH.GOOGLE.COM](https://colab.research.google.com)

오픈소스를 활용하여 비식별화 모델 만들기

선행 지식이 필요한 당신을 위해...

첫 번째 신경망 훈련하기: 기초적인 분류 문제

- Tensorflow에서 제작한 머신러닝 초보자를 위한 튜토리얼
- 이미지 분류 모델
- “구글 코랩에서 실행하기”를 눌러 직접 실행시켜보자.

조태호. (2020.02). 모두의 딥러닝 개정2판. 길벗

- 딥러닝에 관련된 쉬운 설명과 쉬운 코드
- 어떤 방식으로 딥러닝이 진행되는 지 감을 익히기 좋다.

비식별화란 무엇인가?

- 데이터 내 개인을 식별할 수 있는 정보가 있는 경우, 일련의 과정을 통해 특정 개인을 식별하기 어렵도록 하는 조치

정다은. (2020.09.08). 출입명부 쓴 뒤, 날아든 문자 "술 사줄게" "남친 있니?". SBSNEWS.

- 이름, 주민번호, 사진, 차량번호, 계좌번호 등

그 중 **사람 얼굴**과 **차량 번호판**을 비식별화 해보자!

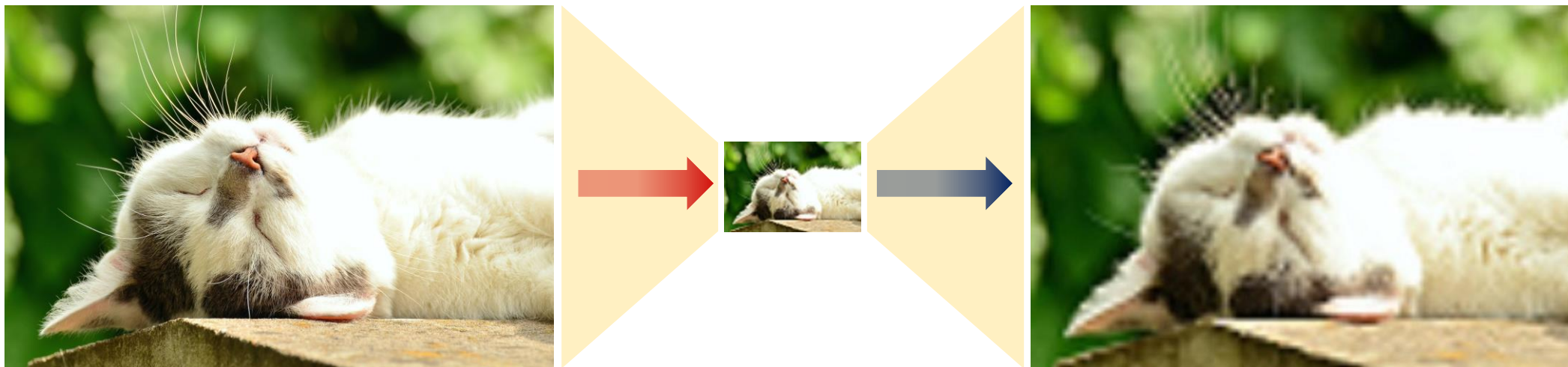
안면 탐지 및 비식별화



차량 번호판 탐지 및 비식별화

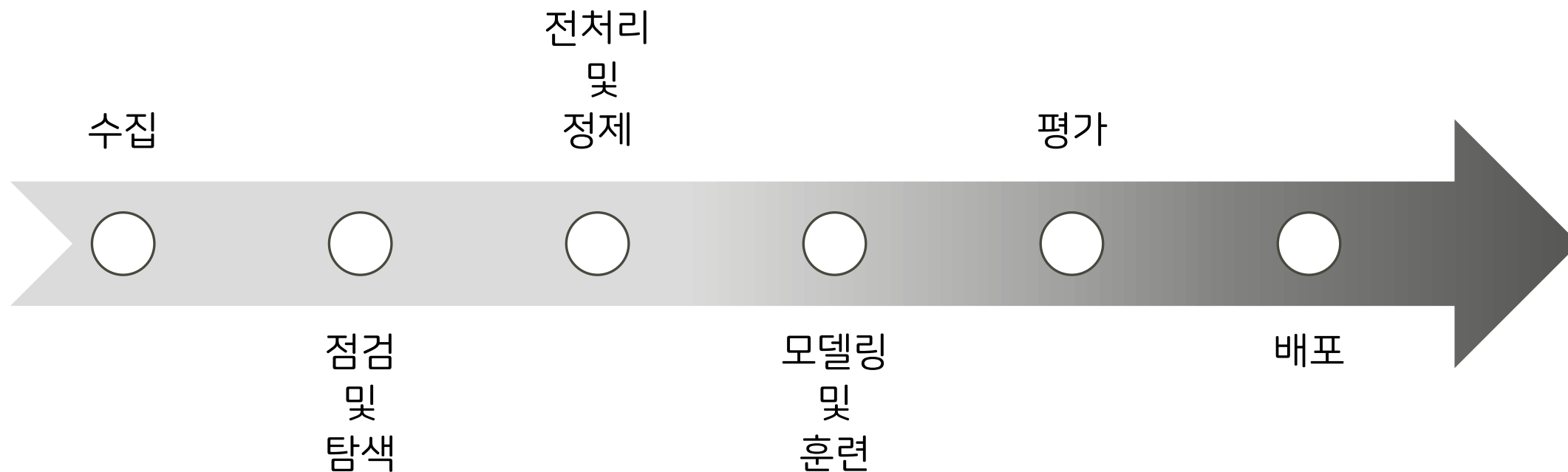


여기서 사용할 모자이크의 원리



1. 이미지를 축소했다가 확대하면 정보가 손실된다.
2. 원하는 부분만 극한으로 축소시키고 다시 확대하면 손실된 픽셀 값에 의해 계단 현상이 일어날 것이다.

일반적인 머신 러닝 과정



오늘의 작업 과정

영상 입력

관심 영역 탐지

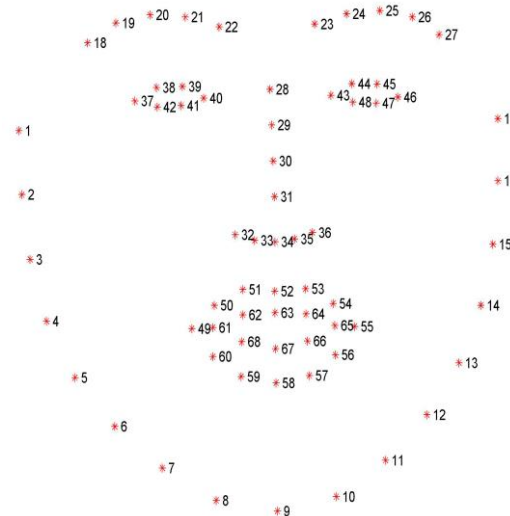
기 존재하는 모델
또는 가중치 적용



Dlib-models



- Dlib 내 안면인식에 관련한 기능 존재
- 그 중 사전 학습된 face landmark 모델 사용



초기 세팅 및 과정들 - Person de-identification

- 구글 드라이브 연동
- 필요한 자료 GitHub에서 클론
- 모델 압축 풀기
- 사용할 라이브러리 import
- 사용할 이미지 입력 및 확인
- 사용 모델 선언



```
import cv2 # 이미지 처리
import dlib # 이미지 처리 및 얼굴 인식
from google.colab.patches import cv2_imshow # 이미지 확인

# 이미지 파일 입력 및 확인
filename = "people3.jpg"
image = "/content/drive/My Drive/Colab Notebooks/★Dataset★/" + filename
frame = cv2.imread(image)
cv2_imshow(frame)

# point_num에는 5개 또는 68개의 랜드마크 데이터 지정
point_num = 5 # 5 or 68

predictor_path = f"/content/drive/My Drive/Colab Notebooks/dlib-models/shape_predictor_{point_num}_face_landmarks.dat"
```

dlib에 있는 정면 얼굴 검출 메소드

```
detector = dlib.get_frontal_face_detector()
```

인식된 얼굴에서 랜드마크를 찾기 위한 클래스 생성

```
predictor = dlib.shape_predictor(predictor_path)
```

입력 사진에서 얼굴을 검출하여 **faces**로 반환

```
faces = detector(frame)
```

```
check_img = frame.copy()
```

```
# 얼굴로 예상되는 부분을 바운딩 박스로 그림
```

```
for face in faces:
```

```
    x1, y1, x2, y2 = face.left(), face.top(), face.right(), face.bottom()
```

```
# 왼쪽 상단의 좌표와 오른쪽 하단의 좌표로 얼굴을 감싸는 사각형  
# 을 그림
```

```
    cv2.rectangle(check_img, (x1, y1), (x2, y2), (0,0,255), thicknes  
s=2)
```

```
cv2_imshow(check_img)
```

```
# cv2.rectangle(img, start, end, color, thickness)
```

```
# img -> 입력 영상
```

```
# start -> 시작 좌표
```

```
# end -> 종료 좌표
```

```
# color -> BGR형태의 컬러
```

```
# thickness -> 선의 두께
```



```
check_img = frame.copy()

# 얼굴 내 랜드마크로 예상되는 부분에 point_num개의 점을 찍음

if len(faces):

    for face in faces:

        landmarks = predictor(check_img, face)

        for n in range(0, point_num):

            # 각 랜드마크의 x, y 좌표를 찾아냄

            x = landmarks.part(n).x

            y = landmarks.part(n).y

            # 원을 그림

            cv2.circle(check_img, (x, y), 5, (255,0,0), -1)

            # 텍스트를 입력

            cv2.putText(check_img, str(n), (x,y), cv2.FONT_HERSHEY_SIMPLEX, 1,
(255, 255, 255), 1, cv2.LINE_AA)

cv2.imshow(check_img)
```

원을 그림

cv2.circle(img, center, radian, color, thickness)

img -> 입력 영상

center -> 원의 중심 좌표(x,y)

radian -> 반지름

color -> BGR 형태의 컬러

thickness -> 선의 두께, -1 이면 원 안쪽을 채움

```
check_img = frame.copy()

# 얼굴로 예상되는 부분을 모자이크

if len(faces):
    for face in faces:
        x1, y1, x2, y2 = face.left(), face.top(), face.right(), face.bottom()

        ROI = check_img[y1:y2, x1:x2]

        shrink = cv2.resize(ROI, None, fx=0.05, fy=0.05)

        zoom = cv2.resize(shrink, (x2-x1, y2-
y1), interpolation=cv2.INTER_AREA)

        check_img[y1:y2, x1:x2] = zoom

cv2_imshow(check_img)
```

cv2.putText(img, text, org, font, fontSacle, color, thickness, lineType)

img -> 입력 영상

text -> 표시할 문자열

org -> 문자열이 표시될 위치

font -> 폰트 종류

fontSacle -> 폰트 크기

color -> 폰트 색상

thickness -> 글자 두께

lineType -> 선 종류

cv2.resize(img, dsize, fx, fy, interpolation)

img -> 입력 영상

dsize -> 절대 크기. 가로, 세로 형태의 튜플

fx -> 가로 사이즈의 배수

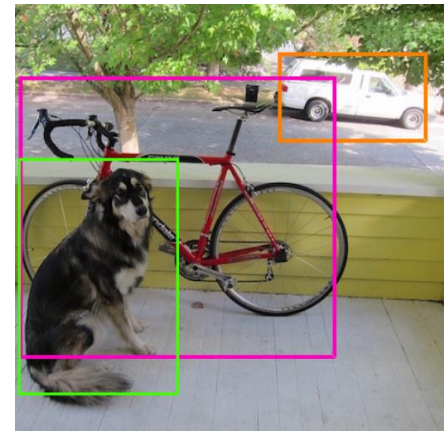
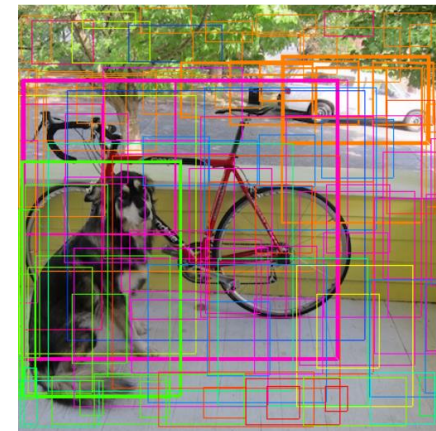
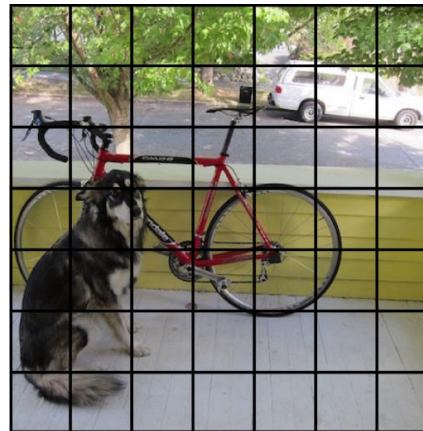
fy -> 세로 사이즈의 배수

interpolation -> 보간법

License-plate Detection by YOLO



- 이란 차량 번호판 검출 목적
- YOLO v.3를 사용
- 이미지/비디오 모두 사용할 수 있는 코드



초기 세팅 및 과정들 – Plate de-identification

- 구글 드라이브 연동
- 필요한 자료 GitHub에서 클론
- model.weight 다운로드 및 드라이브에 적재
- 사용할 라이브러리 import
- 사용할 이미지 입력 및 확인
- 딥러닝을 위해 사용할 파라미터값 선언
- 사용 딥러닝 모델 선언



```
import cv2 # 이미지 처리 패키지

import numpy as np # 수학 연산 패키지

from google.colab.patches import cv2_imshow # 이미지 확인

# 이미지 파일 입력 및 확인

filename = "car2.jpg"

image = "/content/drive/My Drive/Colab Notebooks/★Dataset★/" + filename

test = cv2.imread(image, cv2.IMREAD_COLOR)

cv2_imshow(test)
```


하이퍼 파라미터 초기값

하이퍼 파라미터는 모델링할 때 사용자가 직접 세팅해주는 값

confThreshold = 0.5 # 신뢰도 임계치 Confidence threshold

너무 낮게 설정하면(0.1) 겹치는 객체를 감지하지 못하고

너무 높게 설정하면(1.0) 같은 객체의 중복 바운딩 박스를 얻음

nmsThreshold = 0.4 # 비최대값 억제 임계치 Non-
maximum suppression threshold

inpWidth = 416 # 네트워크 입력 너비 # 320(빠른 결과), 416(디폴트), 608(정확한 결과) 가능

inpHeight = 416 # 네트워크 입력 높이 # 320(빠른 결과), 416(디폴트), 608(정확한 결과) 가능

훈련된 가중치가 저장된 model 파일과 네트워크 구조를 표현하는 config 파일 불러오기

```
modelWeights = "/content/drive/My Drive/Colab Notebooks/yolo-license-plate-detection/model.weights"
```

```
modelConfiguration = "/content/drive/My Drive/Colab Notebooks/yolo-license-plate-detection/darknet-yolov3.cfg"
```

opencv에서 사용할 수 있는 dnn(deep neural network) 중 하나인 Darknet을 사용

readNetFromDarknet() 함수를 사용하여 사용자가 직접 호출할 수도 있으나,

OpenCV 4.0.0버전부터는 readNet() 대표함수를 사용하는 것을 권장

```
net = cv2.dnn.readNet(model=modelWeights, config=modelConfiguration, framework="darknet")
```

```
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV) # 선호하는 백엔드를 지정
```

```
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU) # 선호하는 타겟 디바이스를 지정
```

readNet(String&model, String&config = "", String& framework = "")

String&model -> model 파일

String&config -> 네트워크 구조를 표현하는 config 파일

framework -> 명시적인 딥러닝 프레임워크 이름

OpenCV의 forward 함수에는 네트워크에서 어디까지 실행할지를 알려주는 끝 레이어의 정보가 필요

출력 레이어의 이름 가져오기

```
def getOutputsNames(net):
```

네트워크에 있는 모든 레이어의 이름 가져오기

```
    layersNames = net.getLayerNames()
```

연결되어 있지 않은 출력 레이어(네트워크의 마지막 레이어)의 이름을 가져옴

```
    return [layersNames[i[0] -  
1] for i in net.getUnconnectedOutLayers()]
```

자동차 번호판으로 예상되는 부분을 모자이크 처리

```
def blur_out(classId, conf, left, top, right, bottom):
```

```
# print('left= ', left, 'top= ', top, 'right= ', right, 'bottom= ', bottom)
```

```
car_img = frame[top:bottom, left:right]
```

```
shrink = cv2.resize(car_img, None, fx=0.05, fy=0.05)
```

```
zoom = cv2.resize(shrink, (right-left, bottom-top), interpolation=cv2.INTER_AREA)
```

```
frame[top:bottom, left:right] = zoom
```

후처리 단계

비최대값 억제 방법(non-maxima suppression)을 이용하여 낮은 신뢰도를 가진 바운딩 박스를 제거하기

```
def postprocess(frame, outs):
```

```
    frameHeight = frame.shape[0]
```

```
    frameWidth = frame.shape[1]
```

네트워크를 통해 나오는 모든 바운딩 박스를 훑어보고 높은 점수를 가진 바운딩 박스만 남김.

그리고 가장 신뢰도 점수가 높은 클래스로 바운딩 박스의 클래스값이 할당

```
    classIds = []
```

```
    confidences = []
```

```
    boxes = []
```

```
for out in outs:
```

```
    print("out.shape :", out.shape)
```

```
for detection in out:
```

```
    scores = detection[5:]
```

```
    classId = np.argmax(scores)
```

```
    confidence = scores[classId]
```

```
if detection[4] > confThreshold:
```

```
    print(detection[4], " - ", scores[classId],
```

```
          " - th :", confThreshold)
```

```
    print(detection)
```

```
if confidence > confThreshold:
```

네트워크 출력 경계 상자는 각각 클래스 수 + 5개 요소로 구성된 벡터로 표시

5개 요소: center_x, center_y, width, height 그리고 confidences

```
center_x = int(detection[0] * frameWidth)
center_y = int(detection[1] * frameHeight)
width = int(detection[2] * frameWidth)
height = int(detection[3] * frameHeight)
left = int(center_x - width / 2)
top = int(center_y - height / 2)
classIds.append(classId)
confidences.append(float(confidence))
boxes.append([left, top, width, height])
```

비최대값 억제 방법(non-maxima suppression)을 통해 낮은 신뢰도를 가진 불필요한 중복 바운딩 박스를 제거하기

```
indices = cv2.dnn.NMSBoxes(bboxes = boxes, scores = confidences, score_threshold = confThreshold, nms_threshold = nmsThreshold)
```

```
for i in indices:
```

```
    i = i[0]
```

```
    box = boxes[i]
```

```
    left = box[0]
```

```
    top = box[1]
```

```
    width = box[2]
```

```
    height = box[3]
```

```
    blur_out(classIds[i], confidences[i], left, top, left + width, top + height)
```

```
# NMSBoxes(bboxes, scores, score_threshold, nms_threshold[, eta[, top_k]])
```

```
# bboxes -> NMS를 적용시킬 바운딩 박스
```

```
# score -> 네트워크를 통해 나오는 점수
```

```
# score_threshold -> 신뢰도 임계치 Confidence threshold
```

```
# nmsThreshold -> 비최대값 억제 임계치
```


이미지 읽기

```
frame = cv2.imread(image)
```

네트워크 입력으로 Mat 타입의 2차원 영상을 그대로 입력하는 것이 아니라 블롭(blob) 형식으로 변경해야 함.

블롭이란 영상 등의 데이터를 포함할 수 있는 다차원 데이터 표현 방식

OpenCV에서 블롭은 Mat타입의 4차원 행렬로 표현됨

각 차원은 N(영상개수), C(채널개수), H(영상의 세로 크기), W(영상의 가로 크기)를 의미함

OpenCV에서는 blobFromImage() 함수를 이용하여 Mat 영상으로부터 블롭을 생성함.

네트워크 입력을 위해 데이터 형식을 바꾸기

```
blob = cv2.dnn.blobFromImage(frame, scalefactor = 1/255, size = (inpWidth, inpHeight), mean = [0, 0, 0], swapRB = False, crop=False)
```

블롭 객체를 네트워크 입력으로 설정하기

```
# setInput(blob)
```

blob -> 블롭 객체.

```
net.setInput(blob)
```

위의 setInput() 함수로 설정한 입력 블롭을 이용하여 네트워크를 실행하고, outputName에 해당하는 레이어에서의 결과를 Mat 객체로 반환

```
# forward(outputName)
```

outputName -> 출력 레이어 이름

```
outs = net.forward(getOutputsNames(net))
```

Remove the bounding boxes with low confidence

```
postprocess(frame, outs)
```

```
cv2_imshow(frame.astype(np.uint8))
```

blobFromImage(image, scalefactor=1.0, Size&size=Size(), Scalar&mean=Scalar(), swapRB=false, crop=false)

image -> 입력 영상. 1 또는 3 또는 4채널

scalefactor -> 입력 영상 픽셀 값에 곱할 값 e.g. scalefactor = 1/255 는 모든 픽셀 값을 0에서 1로 만들겠다는 것

Size&size -> 출력 영상의 크기

Scalar&mean -> 입력 영상 각 채널에서 뺄 평균값. 만약 image가 BGR 채널 순서이고 swapRB가 true이면 (R 평균, G 평균, B 평균) 순서로 값을 지정

swapRB -> 첫 번째 채널과 세 번째 채널을 서로 바꿀 것인지를 결정하는 플래그. 이 값이 true이면 컬러 입력 영상의 채널 순서를 BGR에서 RGB로 변경

crop -> 입력 영상의 크기를 변경한 후, 크롭(crop)을 수행할 것인지를 결정하는 플래그

감사합니다.

데이터 및 코드 -> ★☆☆여기☆☆★