**### CNN 실습 (최소메모리요구사항: 2,560MB) ###**

**01. Anaconda 가상환경 생성**

# conda create -y --name **cnn** python=3.9

# conda env list

**02. Anaconda 가상환경 실행**

# conda activate **cnn**

# pwd

# mkdir CNN

# cd CNN

**03. Python Package 설치**

# vi requirements.txt

=========================================================

flask==2.2.2

opencv-python==4.6.0.66

tensorflow==2.11.0

numpy==1.23.5

=========================================================

>>> requirements.txt 생성

# pip install -r requirements.txt

**04. PyCharm 실행** (**GUI에서 실행**)

>>> **Open** (선택) >>> **/roor/CNN** (OK) >>> **Trust Project** (선택) >>> **Cancel** (선택)

**05. Anaconda cnn 가상환경 지정**

**File** >>> **Settings…** (선택)

Settings >>> Project:**CNN** (선택)

>>> **Project Settings** >>> Python Interpreter (선택)

>>> **Add Interpreter** (선택) >>> Add Local Interpreter… (선택)

Add Python Interpreter >>> Conda Environment (선택)

>>> **Use existing environment**: (리스트에서 **‘cnn’** 선택) >>> **OK** (선택)

>>> **Apply** (선택) >>> **OK** (선택)

**06. CNN\_Model 파일 디렉토리**

>>> **model** 디렉토리 생성 (CNN ~/CNN에서 마우스 오른쪽버튼 클릭 후 >>> New >>> Directory)

>>> **Model\_animals.h5** 파일 **model** 디렉토리로 이동

**07. WEB\_HTML 파일 디렉토리 생성**

>>> **templates** 디렉토리 생성

**08. ‘templates’ 디렉토리에 base.html 및 index.html 생성**

>>> **base.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CNN Classification</title>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Nanum+Gothic&family=Noto+Sans+KR:wght@300&display=swap" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

</head>

<body>

<nav class="navbar bg-light border-bottom">

<div class="container-fluid">

<a class="navbar-brand bg-" href="/">Animals Classification</a>

</div>

</nav>

<div class="container">

{% block content %}{% endblock %}

</div>

<footer class="text-center">

{% block footer %}

<small> CNN Test </small>

{% endblock %}

</footer>

</body>

</html>

>>> **index.html**

{% extends 'base.html' %}

{% block content %}

<div class="row justify-content-center">

<form class="col-4 p-md-5 mt-4 border rounded-3 bg-light" enctype=multipart/form-data action="{{ url\_for('upload') }}" method="post">

<div class="mb-3">

<label for="upImage" class="form-label">사진&영상</label>

<input type=file id=upImage name=file>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

{% if image is not none %}

<div class="row justify-content-center">

<img class="col-6 m-4" id="image" src="data:image/jpeg;base64,{{ image }}" width="416" height="416" style="background-color: grey">

<h3 class="col-6">

<small class="text-muted">예측 결과 : </small>

{{predict}}

</h3>

</div>

{% endif %}

{% if filename is not none %}

<div class="row justify-content-center">

{{filename}}

<video controls="" autoplay="" name="media">

<source src="{{ url\_for('static', filename=filename) }}" type="video/mp4">

</video>

</div>

{% endif %}

{% endblock %}

**09. animalcnn.py 파일 생성 (CNN ~/CNN 경로에서 작업)**

import numpy as np

import tensorflow as tf

from tensorflow import keras

import cv2

import numpy as np

model = keras.models.load\_model("./model/Model\_animals.h5")

model.predict(np.zeros((1,150,150,3)))

declass = {0:'butterfly', 1:'cat', 2:'chicken', 3:'cow', 4:'dog', 5:'elephant', 6:'horse', 7:'sheep', 8: 'spider', 9:'squirre'}

def predict\_image(image):

im = cv2.resize(image, (150, 150), interpolation=cv2.INTER\_AREA)

print(im.shape)

inp = np.asarray(im).reshape((1,150,150,3)) / 255.0

out = model.predict(inp)

print(out)

ind = np.argmax(out)

print(ind)

if out[0][ind] < 0.4:

return '판별 불가'

return declass[ind]

def predict\_batch(images):

inp = np.asarray(images) / 255.0

out = model.predict(inp)

print(out)

inds = np.argmax(out, axis=1)

print(inds)

classes = []

for i, ind in enumerate(inds):

if out[i][ind] < 0.4:

classes.append('none')

else:

classes.append(declass[ind])

print(classes)

return classes

**10. app.py 파일 생성 (CNN ~/CNN 경로에서 작업)**

from flask import Flask, request, make\_response, render\_template, Response, session, redirect, url\_for, send\_file

import cv2

import numpy as np

import datetime

import os

import sys

from pathlib import Path

import tempfile

**from animalcnn import predict\_image, predict\_batch**

import time

os.environ['CUDA\_VISIBLE\_DEVICES'] = '0'

# use image upload

import base64

app = Flask(\_\_name\_\_, static\_folder='static')

app.config['SECRET\_KEY'] = 'aiot'

@app.before\_request

def before\_request():

session.permanent = True

app.permanent\_session\_lifetime = datetime.timedelta(minutes=5)

session.modified = True

def send\_file\_data(data, mimetype='image/jpeg', filename='output.jpg'):

response = make\_response(data)

response.headers.set('Content-Type', mimetype)

response.headers.set('Content-Disposition', 'attachment', filename=filename)

return response

def gen\_frames(cap):

# 프레임하나씩 예측 (스트리밍 방식으로 사용)

w = round(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = round(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

fps = cap.get(cv2.CAP\_PROP\_FPS)

fourcc = cv2.VideoWriter\_fourcc(\*'mp4v')

filename = 'pred\_video.mp4'

out = cv2.VideoWriter('./static/'+filename, fourcc, 30, (w, h))

cnt = 0

while True:

success, frame = cap.read() # read the camera frame

if not success:

break

cnt+=1

if cnt % 3 ==0:

continue

pred = predict\_image(frame)

font = cv2.FONT\_HERSHEY\_SIMPLEX

org = (300, 100)

fontScale = 1

color = (255, 0, 0)

thickness = 2

frame = cv2.putText(frame, pred, org, font,

fontScale, color, thickness, cv2.LINE\_AA)

out.write(frame)

cap.release()

out.release()

return filename

def batch\_frames(cap):

# 여러프레임 한번에 예측 (영상화 시킬때 사용)

w = round(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = round(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

fps = cap.get(cv2.CAP\_PROP\_FPS)

fourcc = cv2.VideoWriter\_fourcc(\*'mp4v')

filename = 'pred\_video.mp4'

out = cv2.VideoWriter('./static/'+filename, fourcc, 30, (w, h))

cnt = 0

origins = []

images = []

while True:

success, frame = cap.read() # read the camera frame

if not success:

# del session['temp\_filename']

break

cnt+=1

if cnt % 3 ==0:

continue

im = cv2.resize(frame, (150, 150), interpolation=cv2.INTER\_AREA)

origins.append(frame.copy())

images.append(im)

if len(images) > 10:

preds = predict\_batch(images)

for ori, pred in zip(origins, preds):

font = cv2.FONT\_HERSHEY\_SIMPLEX

org = (150, 150)

fontScale = 1

color = (255, 0, 0)

thickness = 2

frame = cv2.putText(ori, pred, org, font, fontScale, color, thickness, cv2.LINE\_AA)

out.write(frame)

images = []

origins = []

cap.release()

out.release()

return filename

@app.route("/", methods=['POST', 'GET'])

def index():

return render\_template('index.html',image = None, filename= None)

@app.route('/upload', methods=['GET', 'POST'])

def upload():

if request.method == 'POST':

if 'file' not in request.files:

# flash('No file part')

return 'You forgot Snap!'

fs = request.files.get('file')

# print(fs.read(), file=sys.stderr)

if 'jpg' in fs.filename:

img = cv2.imdecode(np.frombuffer(fs.read(), np.uint8), cv2.IMREAD\_UNCHANGED)

pred = predict\_image(img)

ret, buf = cv2.imencode('.jpeg', img)

b64\_img = base64.b64encode(buf).decode('utf-8')

return render\_template('index.html', image = b64\_img, predict = pred, filename = None)

elif 'mp4' in fs.filename:

with tempfile.TemporaryDirectory() as td:

temp\_filename = Path(td) / 'uploaded\_video'

fs.save(temp\_filename)

vidcap = cv2.VideoCapture(str(temp\_filename))

filename = batch\_frames(vidcap)

return send\_file('./static/{}'.format(filename), download\_name = filename, as\_attachment = True)

# return render\_template('index.html', image = None, filename = filename)

else:

return 'You forgot Snap!'

return 'Hello World!'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, port=5000)

**11. PyCharm 에서 실행버튼 클릭 후**

>>> \* Running on <http://127.0.0.1:5000> (클릭)

>>> 동물 이미지(**확장자: jpg**) 다운로드 후 동작 확인

* **수고하셨습니다 -**