강원대학교
AI 소프트웨어학과

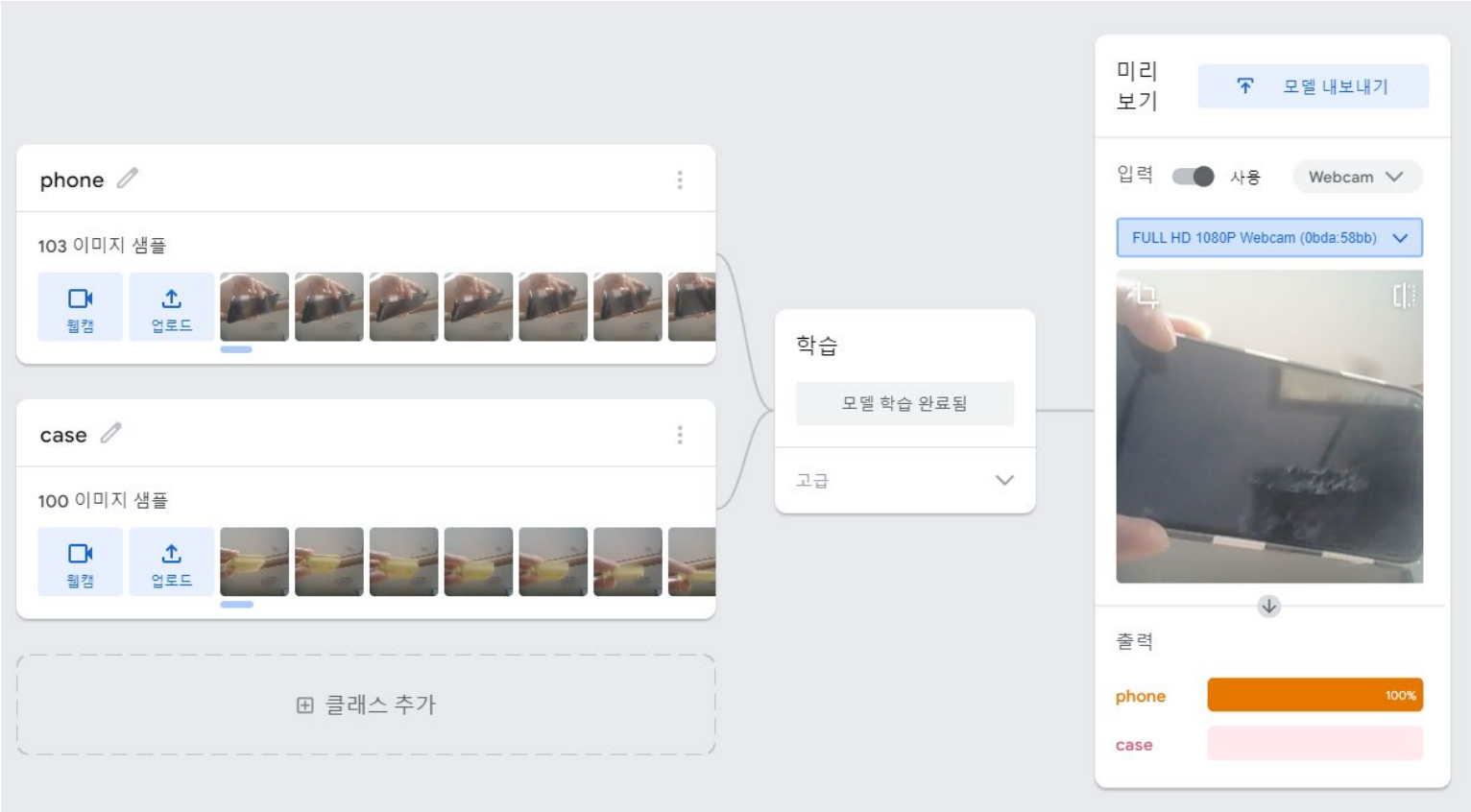
인공지능
– 이미지 분석 –

KNU
강원대학교

01 이미지 분석

이미지분석-티처블머신

- 티처블머신 : 구글에서 CNN기반의 모델을 Fine-tuning 할 수 있도록 하는 사이트
- https://teachablemachine.withgoogle.com/



이미지분석-티처블머신

티처블머신

이미지분석-티처블머신

티처블머신

모델에서 사용할 코드 스니펫:

Keras | OpenCV Keras | Github에 참여

복사

```
from keras.models import load_model  # TensorFlow is required for Keras to work
import cv2  # Install opencv-python
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# CAMERA can be 0 or 1 based on default camera of your computer
camera = cv2.VideoCapture(0)

while True:
    # Grab the webcamera's image.
    ret, image = camera.read()

    # Resize the raw image into (224-height,224-width) pixels
    image = cv2.resize(image, (224, 224), interpolation=cv2.INTER_AREA)

    # Show the image in a window
    cv2.imshow("Webcam Image", image)
```

이미지분석-티처블머신

Python 3.10.9 버전 사용

```
pip install tensorflow==2.11.0
pip install keras-models
conda install -c conda-forge opencv
```

```
import tensorflow.keras
import numpy as np
import cv2
```

티처블 Code

```
np.set_printoptions(suppress=True)
model = load_model("keras_Model.h5", compile=False)
class_names = open("labels.txt", "r").readlines()
camera = cv2.VideoCapture(0)
```

수정 Code

```
model = tensorflow.keras.models.load_model('model.h5')
cap = cv2.VideoCapture(0)
size = (224, 224)
classes = ['phone', 'case']
```

이미지분석-티처블머신

티처블 Code

```
while True:
        ret, image = camera.read()

        image = cv2.resize(image, (224, 224), interpolation=cv2.INTER_AREA)
        cv2.imshow("Webcam Image", image)
        image = np.asarray(image, dtype=np.float32).reshape(1, 224, 224, 3)
        image = (image / 127.5) - 1
```

수정 Code

```
while cap.isOpened(): #비디오 캡처 장치가 열리도록 자동루프(계속 열려있어야하니깐)
        ret, img = cap.read() #비디오 캡처장치에서 프레임을 읽을때 프래임을 읽는 ret변수가 프레임을 성공적으로 가지고 왔는지 판단
        if not ret:
                break
        img = img[:, 200:200+img.shape[0]] #이미지를 자르고 싶은은 사이즈로 자른다.
        img = cv2.flip(img, 1)
```

이미지분석-티처블머신

티처블 Code

```
prediction = model.predict(image)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

print("Class:", class_name[2:], end="")
print("Confidence Score:", str(np.round(confidence_score * 100))[:-2], "%")
```

수정 Code

```
img_input = cv2.resize(img, size)
img_input = cv2.cvtColor(img_input, cv2.COLOR_BGR2RGB)
img_input = (img_input.astype(np.float32) / 127.0) - 1
img_input = np.expand_dims(img_input, axis=0)
prediction = model.predict(img_input)
idx = np.argmax(prediction)

cv2.putText(img, classes[idx], (10,45),cv2.FONT_HERSHEY_SIMPLEX,2 , (100, 255, 0), 2, cv2.LINE_AA)
```

이미지분석-티처블머신

티처블 Code

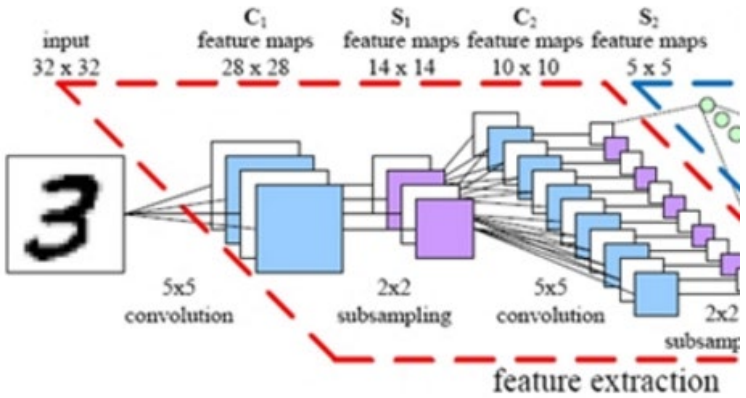
```
keyboard_input = cv2.waitKey(1)
    if keyboard_input == 27:
        break

camera.release()
cv2.destroyAllWindows()
```

수정 Code

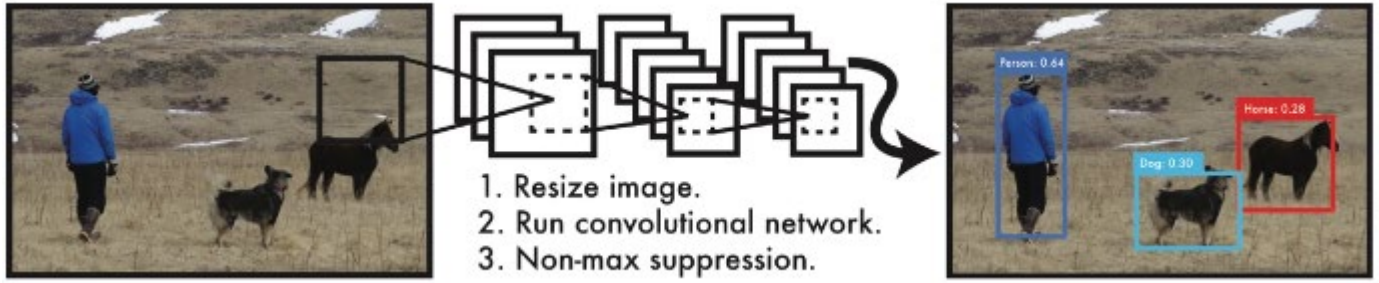
```
 cv2.imshow('result', img)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

이미지분석-YOLO

- YOLO(You Only Look Once) 약어로, 2015년 Joseph Redmon이 해당 모델의 논문으로 발표함
- YOLO가 등장하기 전 Faster R-CNN이라는 모델이 많이 사용 되었지만 실시간성이 부족해 사용되지 않다가 점점 발전하며 사용됨
- CNN기반 Detection방법은 다양한 특징들의 추출을 통해 최적의 box를 찾는 방법이라면 YOLO는 한번에 전체의 특징을 파악해 처리의 속도를 높임 → 주변의 배경들도 학습에 사용해 새로운 객체에 대해 잘 분류함
- 복잡한 특징을 가지는 물체를 잘 판단하지 못함



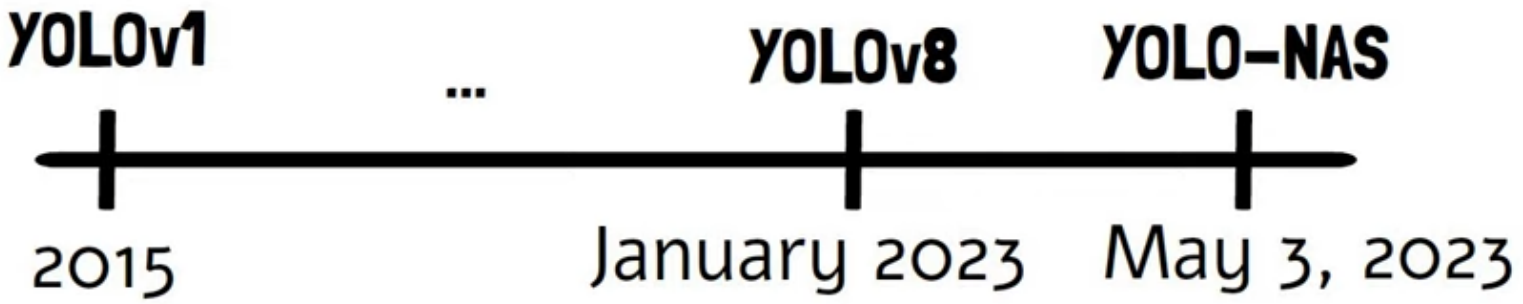
CNN기반 모델



YOLO모델

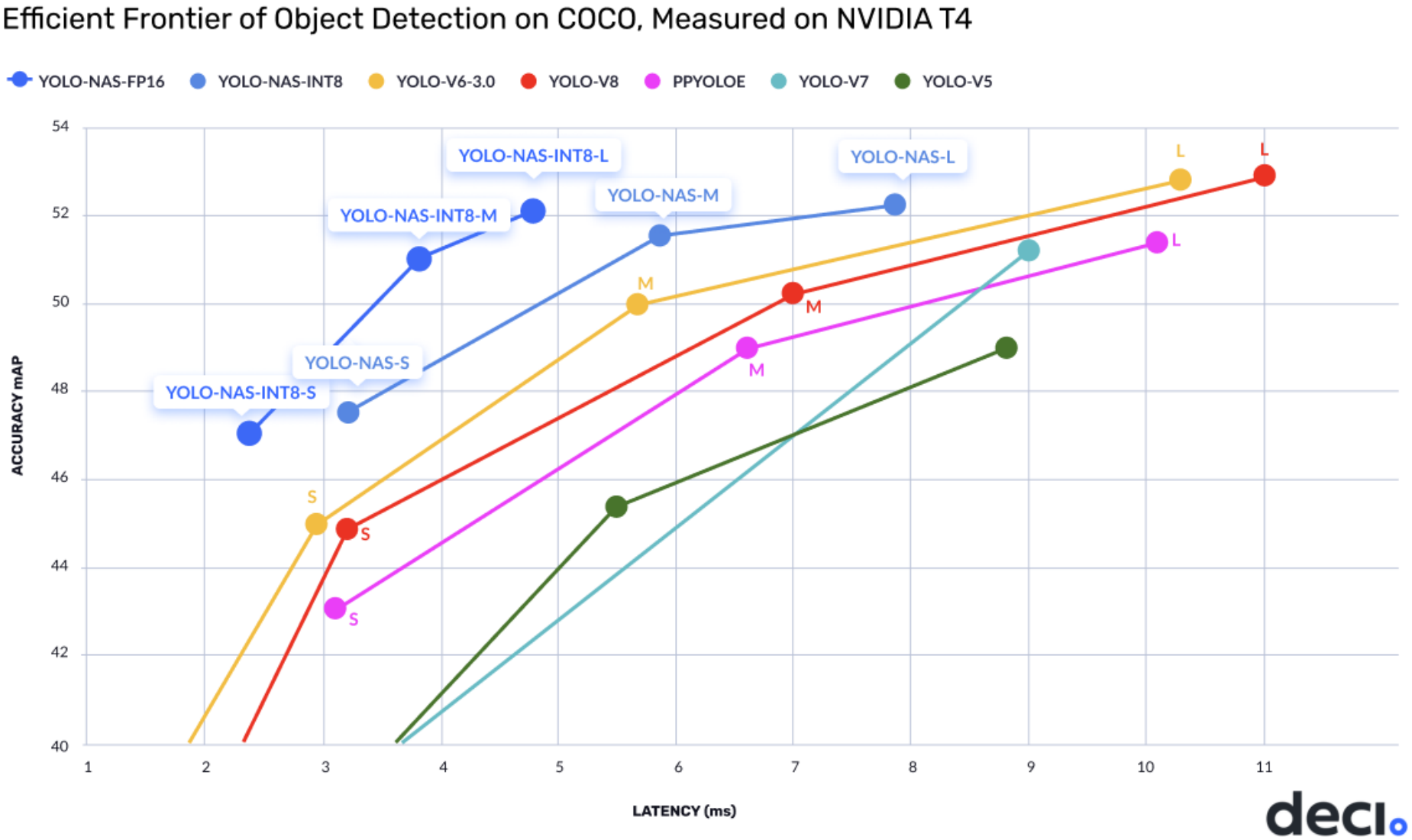
이미지분석-YOLO

- 이미지 디텍팅과 세그멘테이션을 활용해서 구현할 수 있음
- 현재 YOLOv8 버전까지 나와있는 상황



01 이미지 분석

이미지분석-YOLO



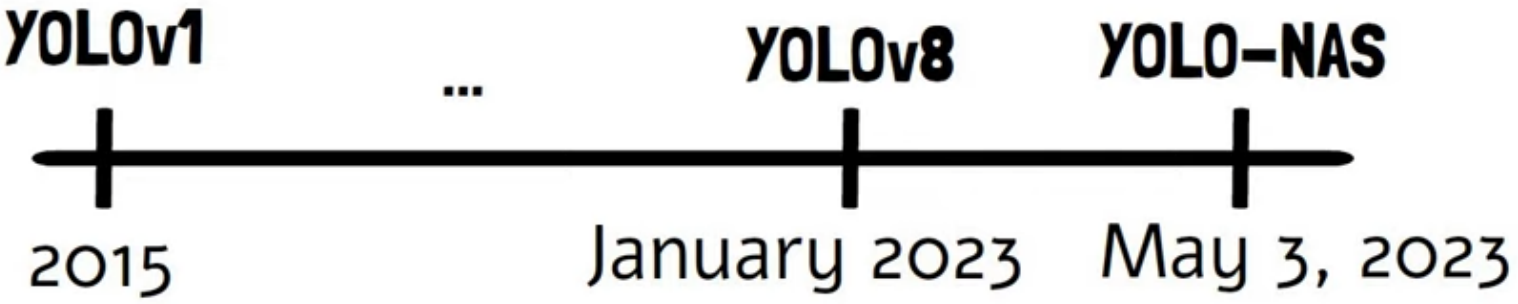
01 이미지 분석

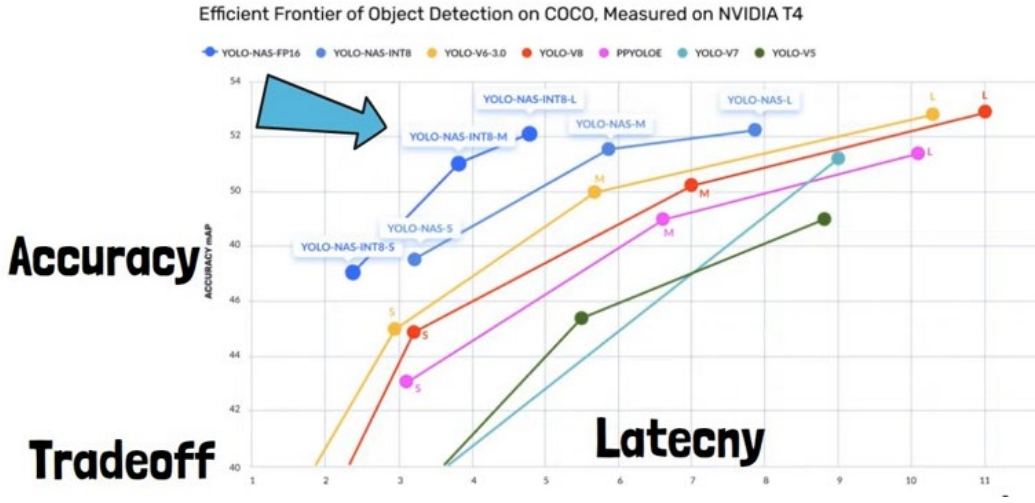
이미지분석-YOLO

- 이미지 디텍팅과 세그멘테이션을 활용해서 구현할 수 있음

이미지분석-YOLO

- YOLO-NAS(Neural Architecture Search)
- Neural Architecture Search : 최적의 네트워크 구조를 디자인은 하는 일을 자동화 해주는 방법



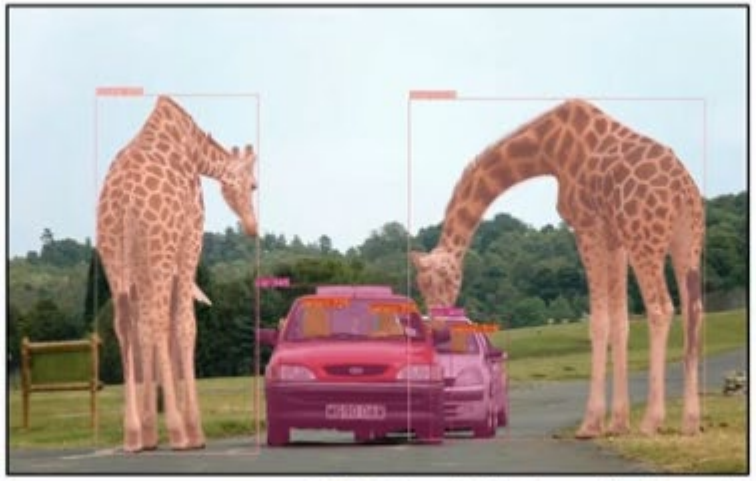


01 이미지 분석

이미지분석-YOLO

이미지 디텍팅과 세그멘테이션, Joint 디텍션을 활용해서 구현할 수 있음

출처: https://github.com/ultralytics/ultralytics



출처: https://github.com/ibaiGorordo/ONNX-YOLOv8-Instance-Segmentation