



# OBJECT DETECTION: LEVERAGING PRE-TRAINED EFFICIENTDET MODEL FOR REAL-TIME DEPLOYMENT WITH GRADIO

Professor: WANG Jing

Member:

1. Somoeurn Virakden 2120246050
2. Uchita Hikaru 2120246057



# OBJECTIVE



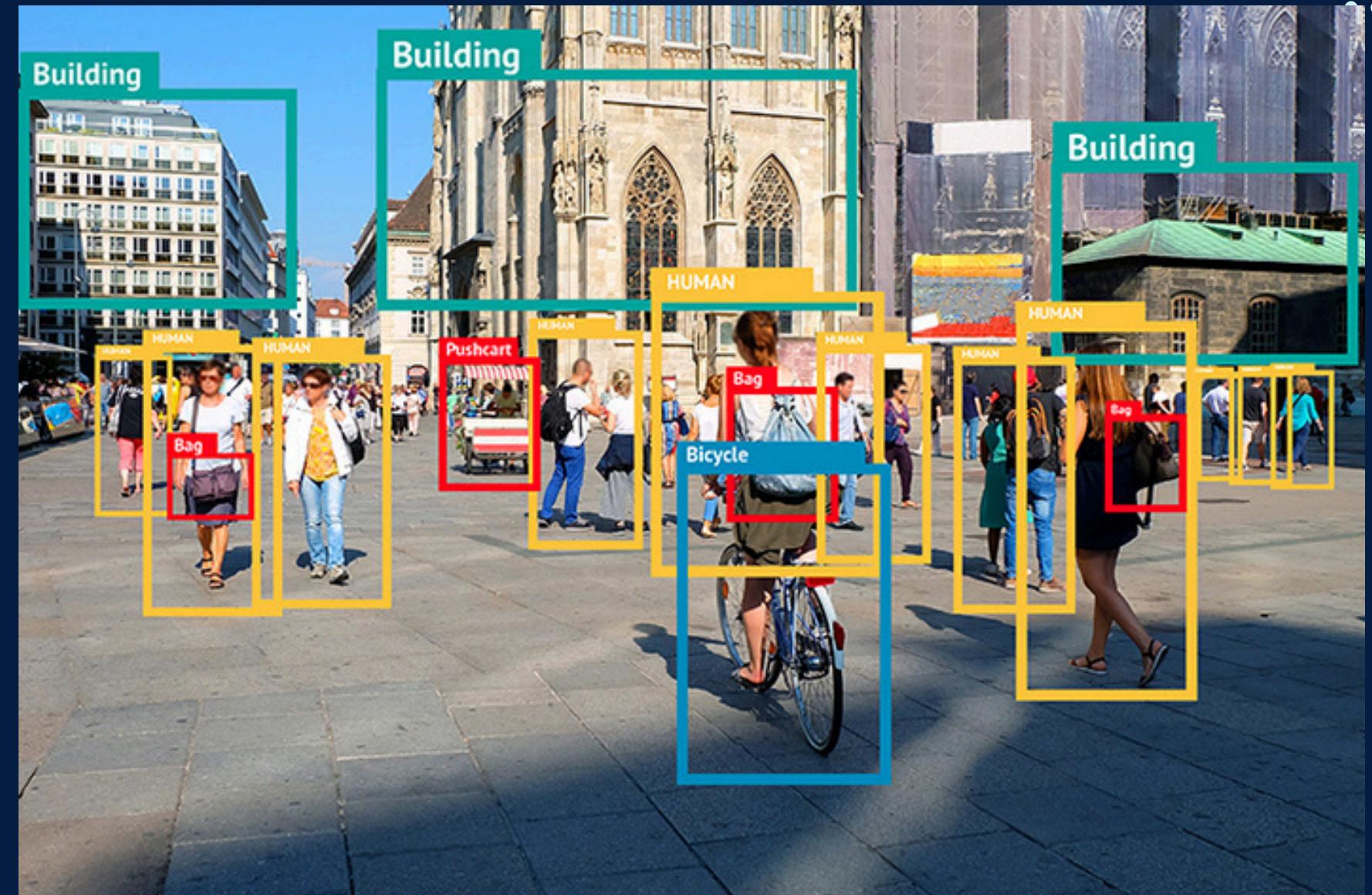
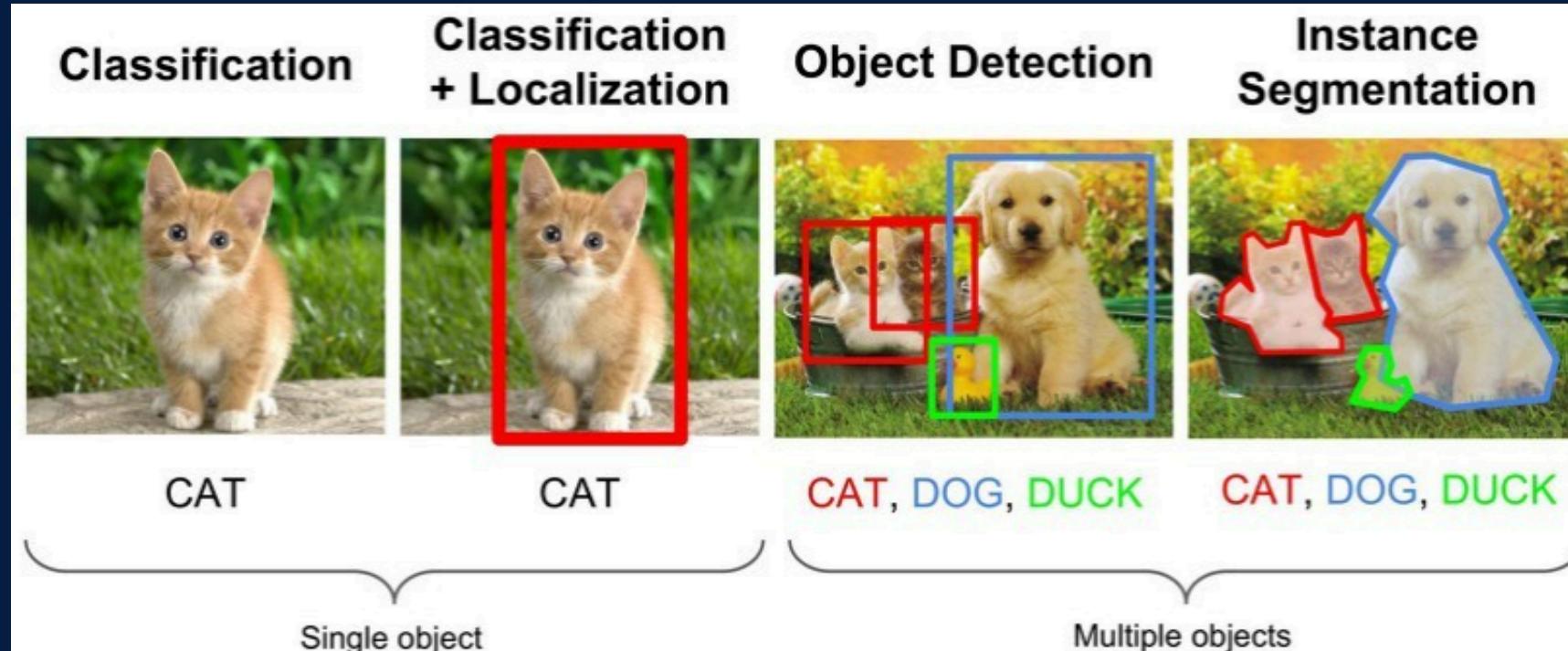
- ➡ Describe of Object Detection
- ➡ Explain EfficientDet models
- ➡ Explore with Model Usage
- ➡ Discuss Postprocessing
- ➡ Deploy Gradio
- ➡ Demo

# Object detection

# What is Object Detection?

2

1



# EfficientDet Model

# BackBone: EfficientNet

- BackBone extracting feature from input  $P_1$ to $P_7$  .

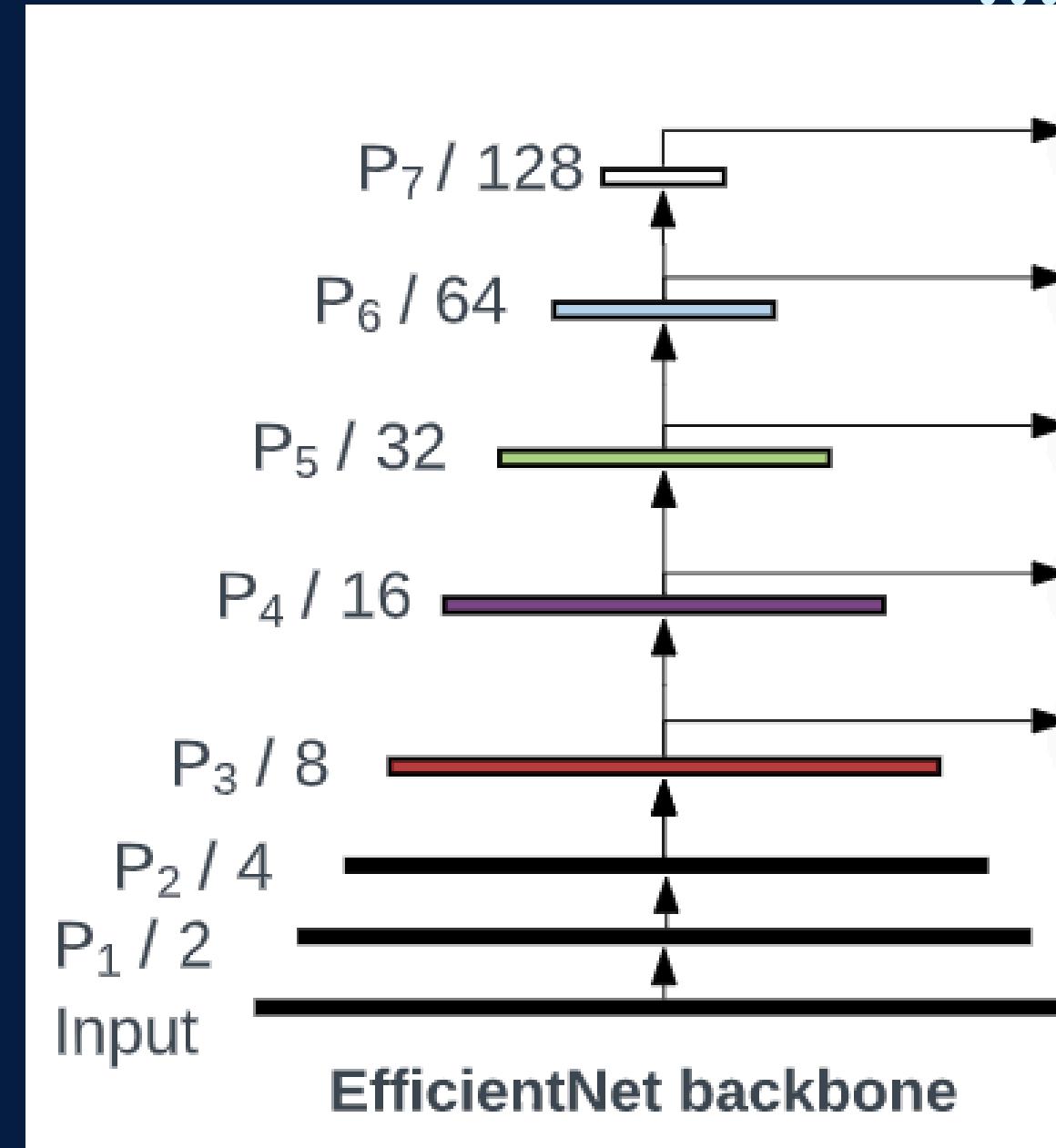
- With Resolution  $1/2^i$

**Look like**  $P_1 = 1/2^1$

$$P_2 = 1/2^2$$

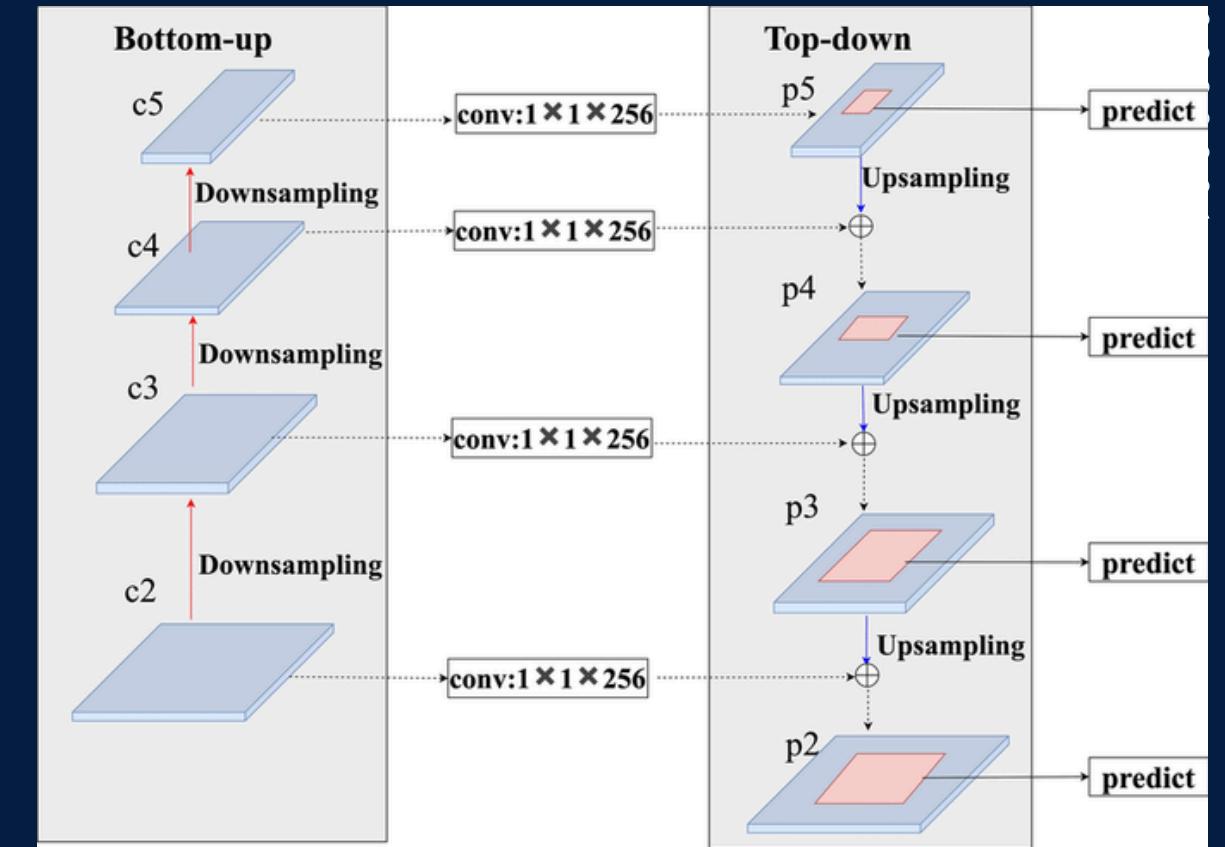
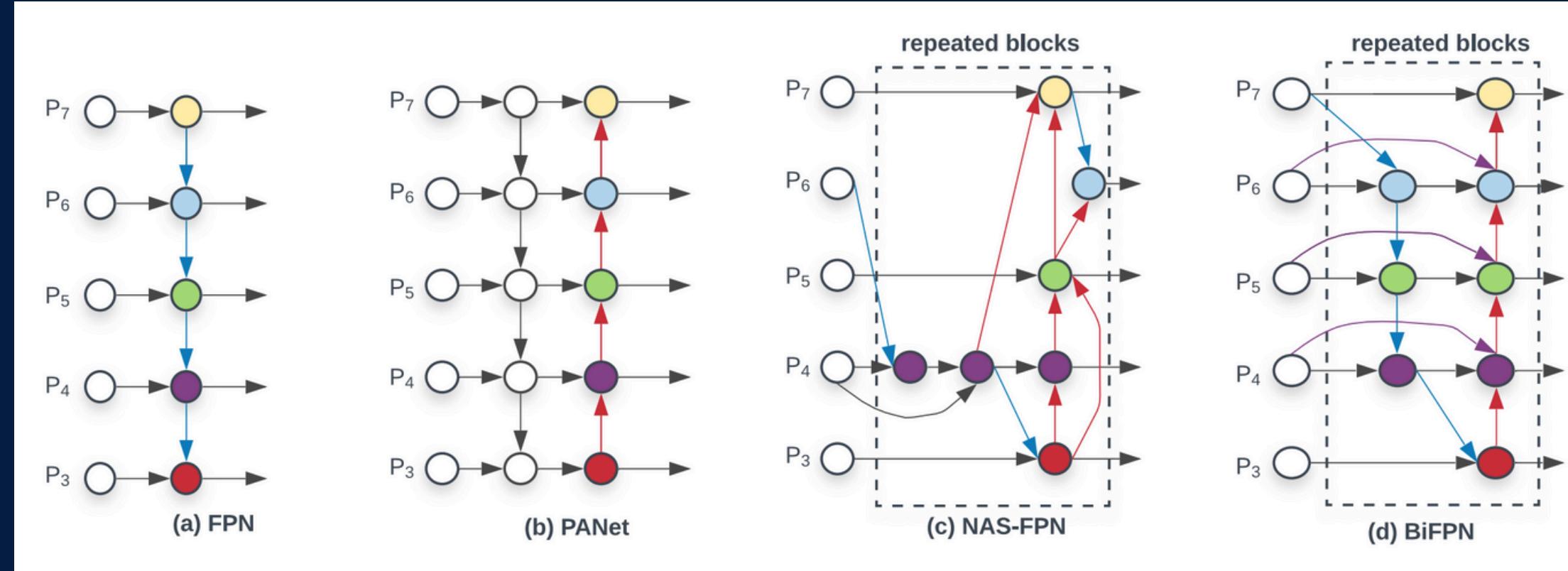
⋮

$$P_7 = 1/2^7$$



# BiFPN

- BiFPN accept feature input  $P^{in} = (P_3^{in}, \dots, P_7^{in})$  from backbone.



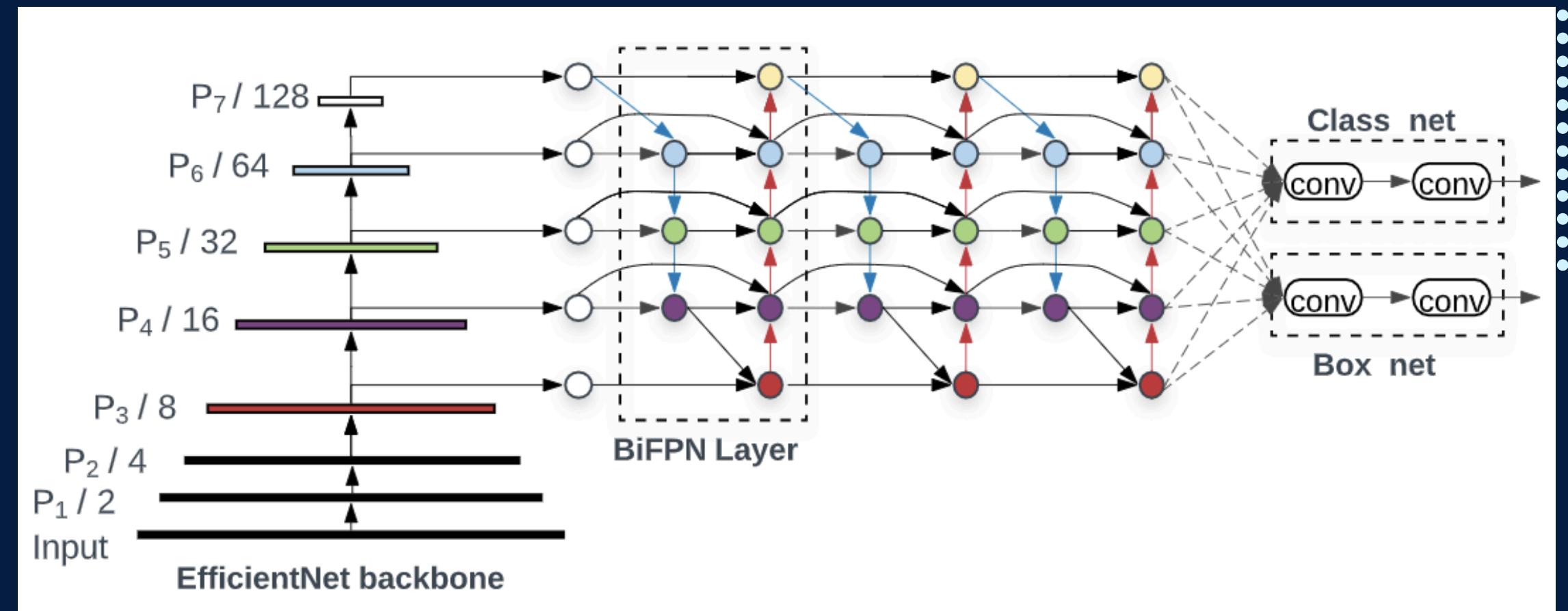
For instance: If input resolution 640x640,  
then  $P_3^{in}$  represents feature level 3 ( $640/2^3 = 80$ )

It use with resolution 80x80

$$\begin{aligned}
 P_7^{out} &= Conv(P_7^{in}) \\
 P_6^{out} &= Conv(P_6^{in} + Resize(P_7^{out})) \\
 &\dots \\
 P_3^{out} &= Conv(P_3^{in} + Resize(P_4^{out}))
 \end{aligned}$$

# Prediction Network

- Two lightweight sub-networks:
- Class Prediction Network
  - Box Prediction Network



EfficientNet variants, ranging from D0 to D7 (e.g., EfficientDet-D0, EfficientDet-D1, ..., EfficientDet-D7)

# MODEL USAGE

# Import Model with TensorFlow Hub



Tensorflow-Hub is a repository of pre-trained models that can be easily integrated into projects. It offers a wide range of models for various tasks, including object detection.

```
● ● ●  
1 import tensorflow_hub as hub  
2  
3 model_url = EfficientDet['EfficientDet D4 1024x1024']  
4 model = hub.load(model_url)
```

```
● ● ●  
1 EfficientDet = {'EfficientDet D0 512x512' : 'https://tfhub.dev/tensorflow/efficientdet/d0/1',  
2 'EfficientDet D1 640x640' : 'https://tfhub.dev/tensorflow/efficientdet/d1/1',  
3 'EfficientDet D2 768x768' : 'https://tfhub.dev/tensorflow/efficientdet/d2/1',  
4 'EfficientDet D3 896x896' : 'https://tfhub.dev/tensorflow/efficientdet/d3/1',  
5 'EfficientDet D4 1024x1024' : 'https://tfhub.dev/tensorflow/efficientdet/d4/1',  
6 'EfficientDet D5 1280x1280' : 'https://tfhub.dev/tensorflow/efficientdet/d5/1',  
7 'EfficientDet D6 1280x1280' : 'https://tfhub.dev/tensorflow/efficientdet/d6/1',  
8 'EfficientDet D7 1536x1536' : 'https://tfhub.dev/tensorflow/efficientdet/d7/1'  
9 }
```

# Model Usage



## Input

- A three-channel image of variable size, has shape [1, height, width, 3] with values in [0, 255].

## Output

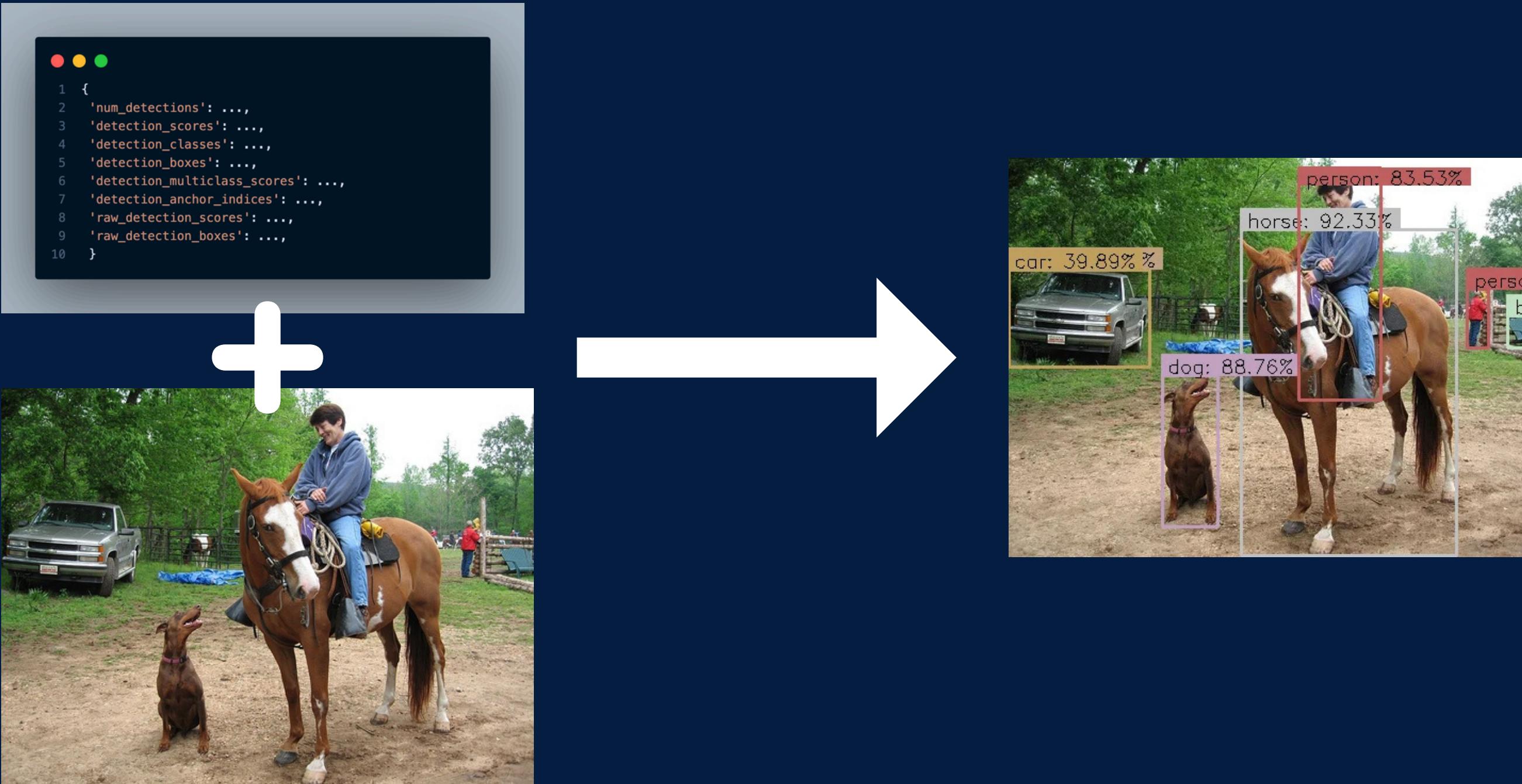
The output dictionary contains:

- num\_detections
- detection\_classes
- detection\_boxes
- detection\_scores
- detection\_multiclass\_scores
- detection\_anchor\_indices
- raw\_detection\_boxes
- raw\_detection\_scores

# POST-PROCESSING

# Post-processing

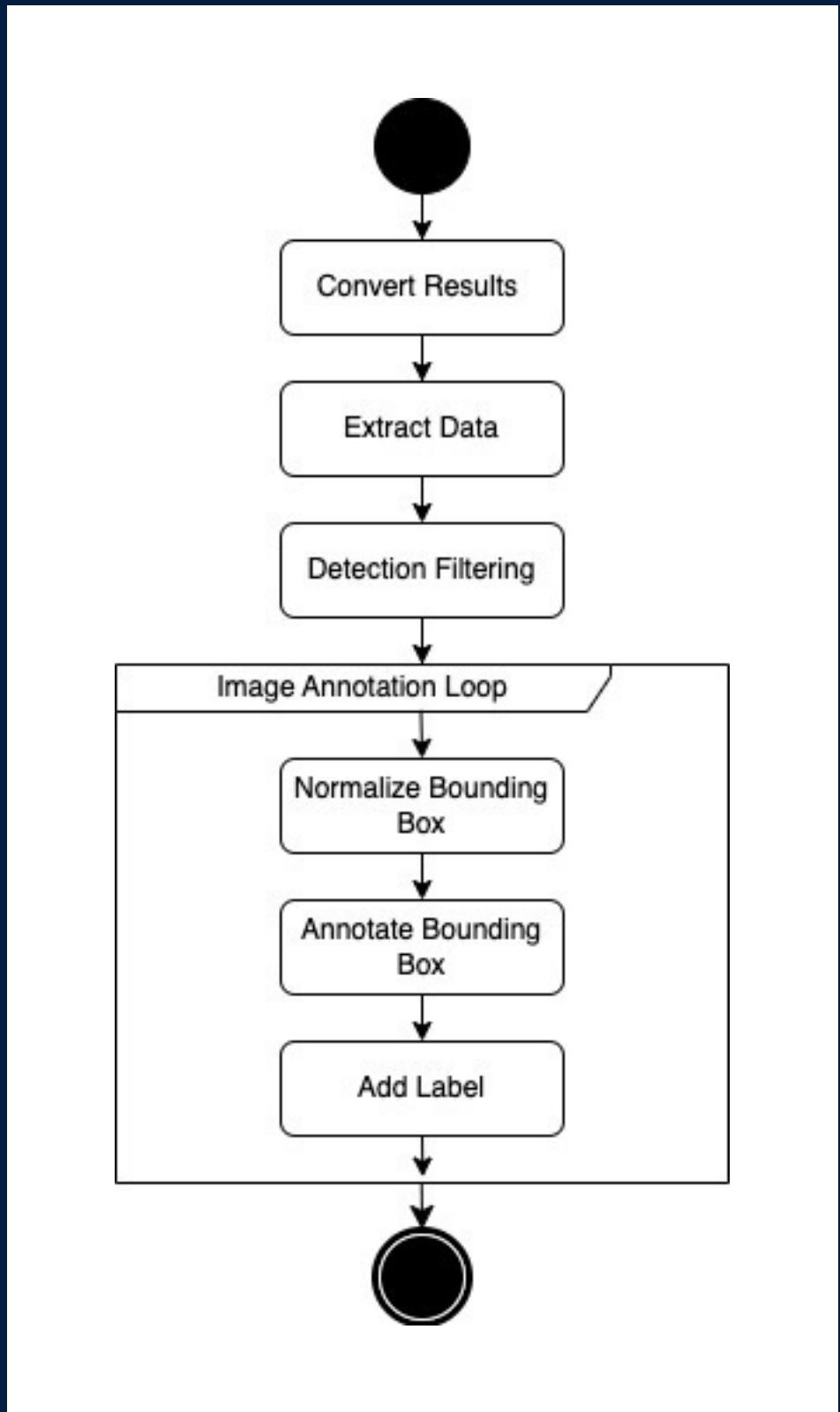
The post-processing function is used for interpret the detection data and annotate the input image for finalize image output.



# Post-processing

The main steps in the post processing function:

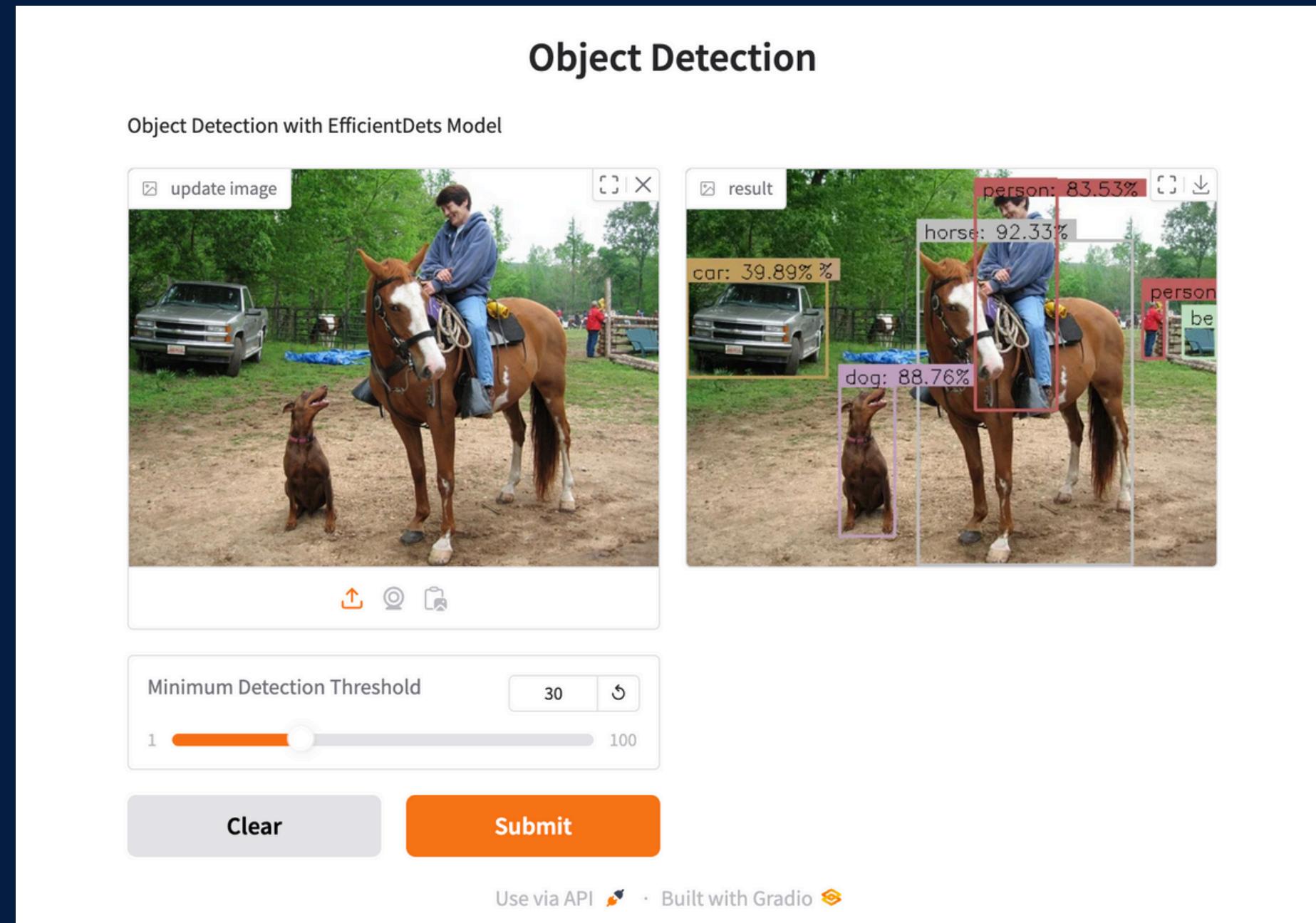
- Convert Result
- Extract Data
- Detection Filtering
- Image Annotation Loop
- Finalization



# GRADIO

# Gradio

Gradio is an open-source Python library for building user-friendly web interfaces for machine learning models, data workflows, and applications.



Gradio web interface for our project

# DEMO

# Thank You