

small
object detection

$[40 \times 40]$

yolo

ssd

fast R-CNN

:

سؤال

چگونه می توان تصاویر کوچک را تشخیص داد؟

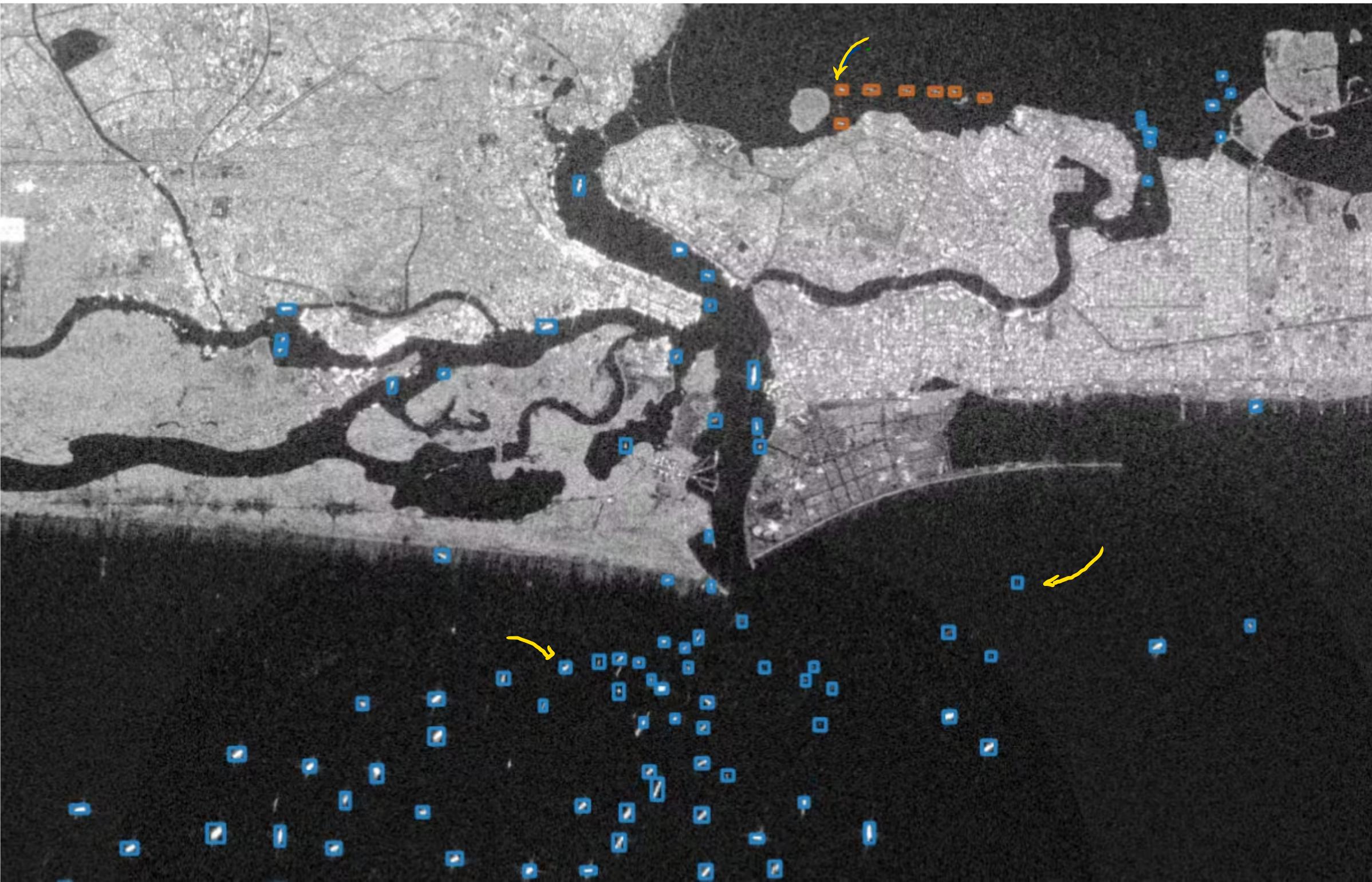
تشخیص اشیا

بازدهی مدل yolov8s-640

X



[نَجِيْرَةٌ]



بازدهی مدل های مختلف روی تصاویر کوچک



YOLOv4: Optimal Speed and Accuracy of Object Detection

Method	Backbone	Size	FPS	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
YOLOv4	CSPDarknet-53	416	96 (V)	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4	CSPDarknet-53	512	83 (V)	43.0%	64.9%	46.5%	24.3%	46.1%	55.2%
YOLOv4	CSPDarknet-53	608	62 (V)	43.5%	65.7%	47.3%	26.7%	46.7%	53.3%

EfficientDet: Scalable and Efficient Object Detection [77]

EfficientDet-D0	Efficient-B0	512	62.5 (V)	33.8%	52.2%	35.8%	12.0%	38.3%	51.2%
EfficientDet-D1	Efficient-B1	640	50.0 (V)	39.6%	58.6%	42.3%	17.9%	44.3%	56.0%
EfficientDet-D2	Efficient-B2	768	41.7 (V)	43.0%	62.3%	46.2%	22.5%	47.0%	58.4%
EfficientDet-D3	Efficient-B3	896	23.8 (V)	45.8%	65.0%	49.3%	26.6%	49.4%	59.8%

Learning Spatial Fusion for Single-Shot Object Detection [48]

YOLOv3 + ASFF*	Darknet-53	320	60 (V)	38.1%	57.4%	42.1%	16.1%	41.6%	53.6%
YOLOv3 + ASFF*	Darknet-53	416	54 (V)	40.6%	60.6%	45.1%	20.3%	44.2%	54.1%
YOLOv3 + ASFF*	Darknet-53	608 ×	45.5 (V)	42.4%	63.0%	47.4%	25.5%	45.7%	52.3%
YOLOv3 + ASFF*	Darknet-53	800 ×	29.4 (V)	43.9%	64.1%	49.2%	27.0%	46.6%	53.4%

good res - DS

bad Reg. Df



برای تشخیص اشیا کوچک باید چکار کنم؟

B → GAM → G

Transformers : 6f C x

morph. esp.

مکالمہ

Conv.
BN
Drop
max

A hand-drawn diagram illustrating a neural network architecture. It starts with a blue square representing an input image, which has a red dot in its center. A red arrow points from this image to the word "CNN" written in blue. From "CNN", a blue wavy arrow points to the word "maxp" written in blue above it. Another blue wavy arrow points from "maxp" to a handwritten-style output "iJUg".

روش های پیشنهادی دانشجویان

500

SR

300

GO

SRGAN

آرلیفٹر بیسیس نہن

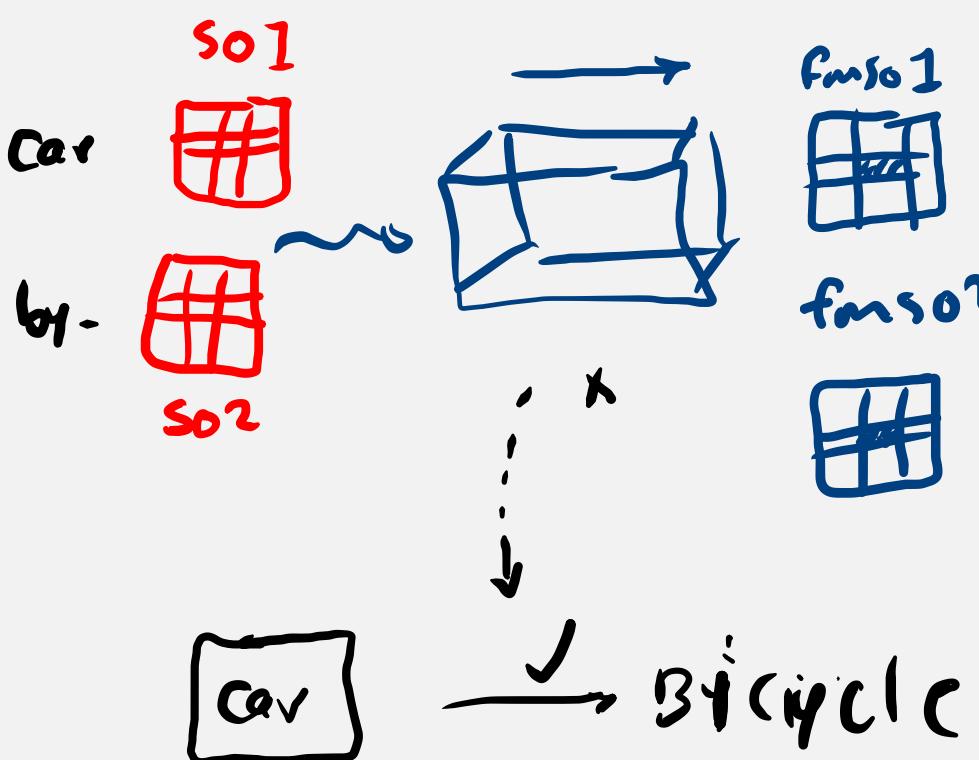
Faster R-CNN

+ FPN

DINO

برای تشخیص اشیا کوچک باید چه کار کنم؟

راه حل ها

~~سیستم پردازش تصویر - زبان~~
Small object detection system

yolo task=detect mode=predict model=yolov8n.pt imgs=1280

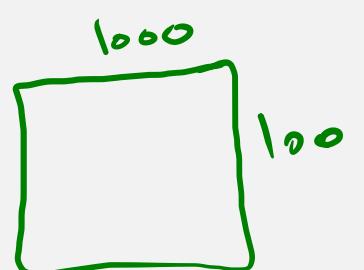
def. 640

YOLOv8n
YOLOv8x
new
old
x-large

مطمئن شویم که داده های کوچک هم برچسب خورده اند.

Pr
Recall

$$\frac{FP}{FN}$$

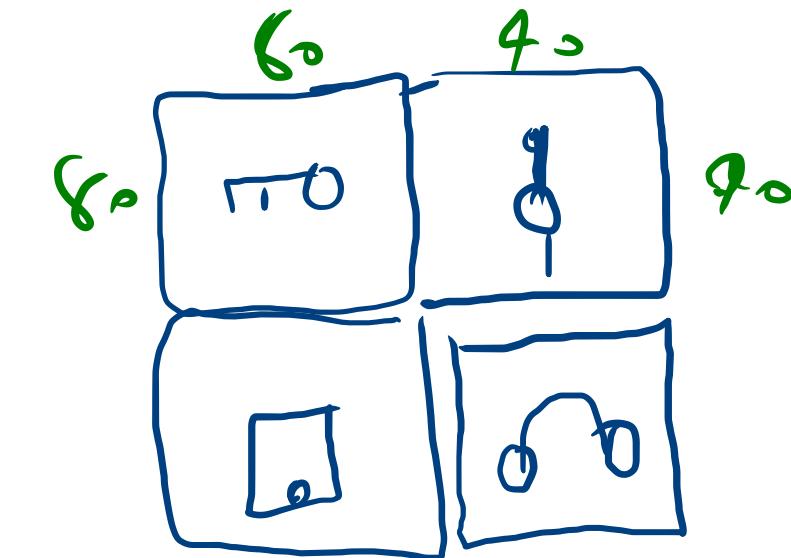
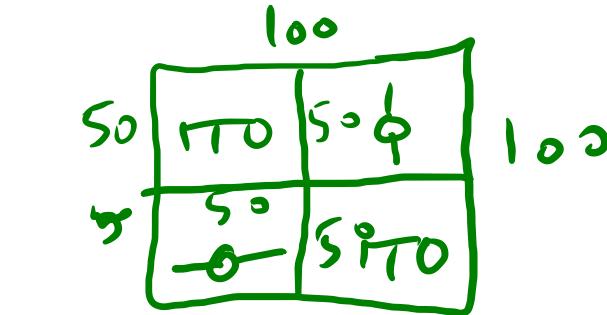
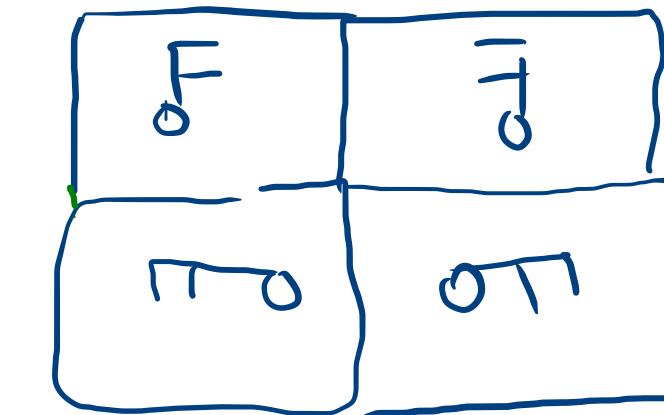
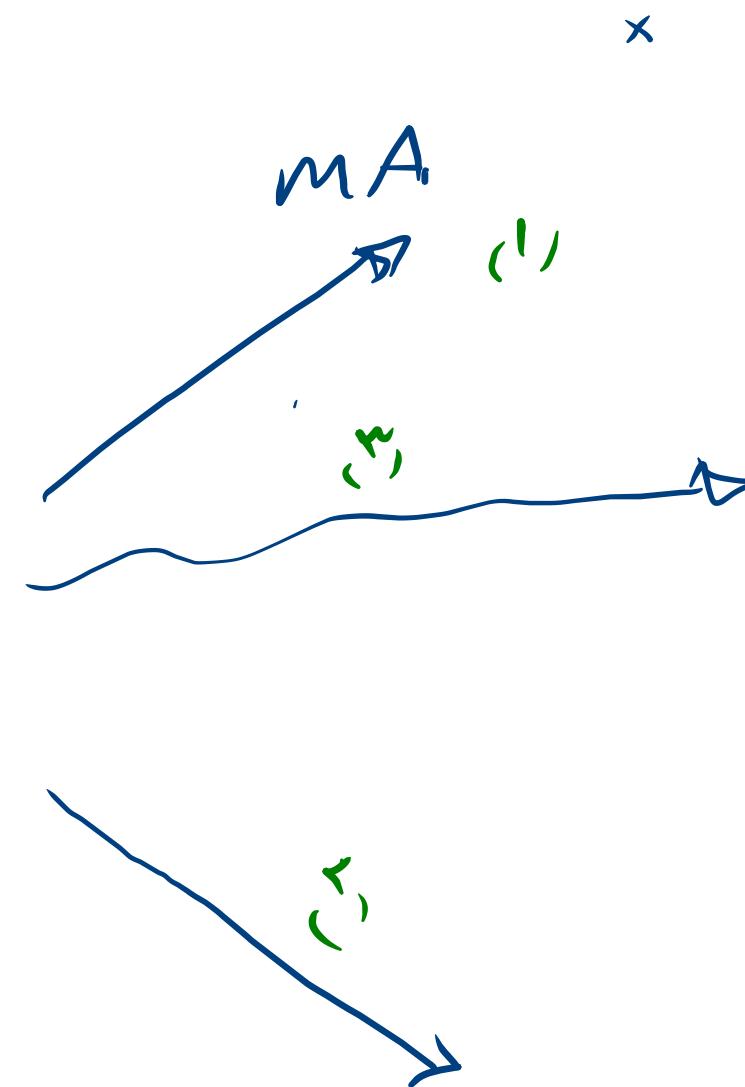
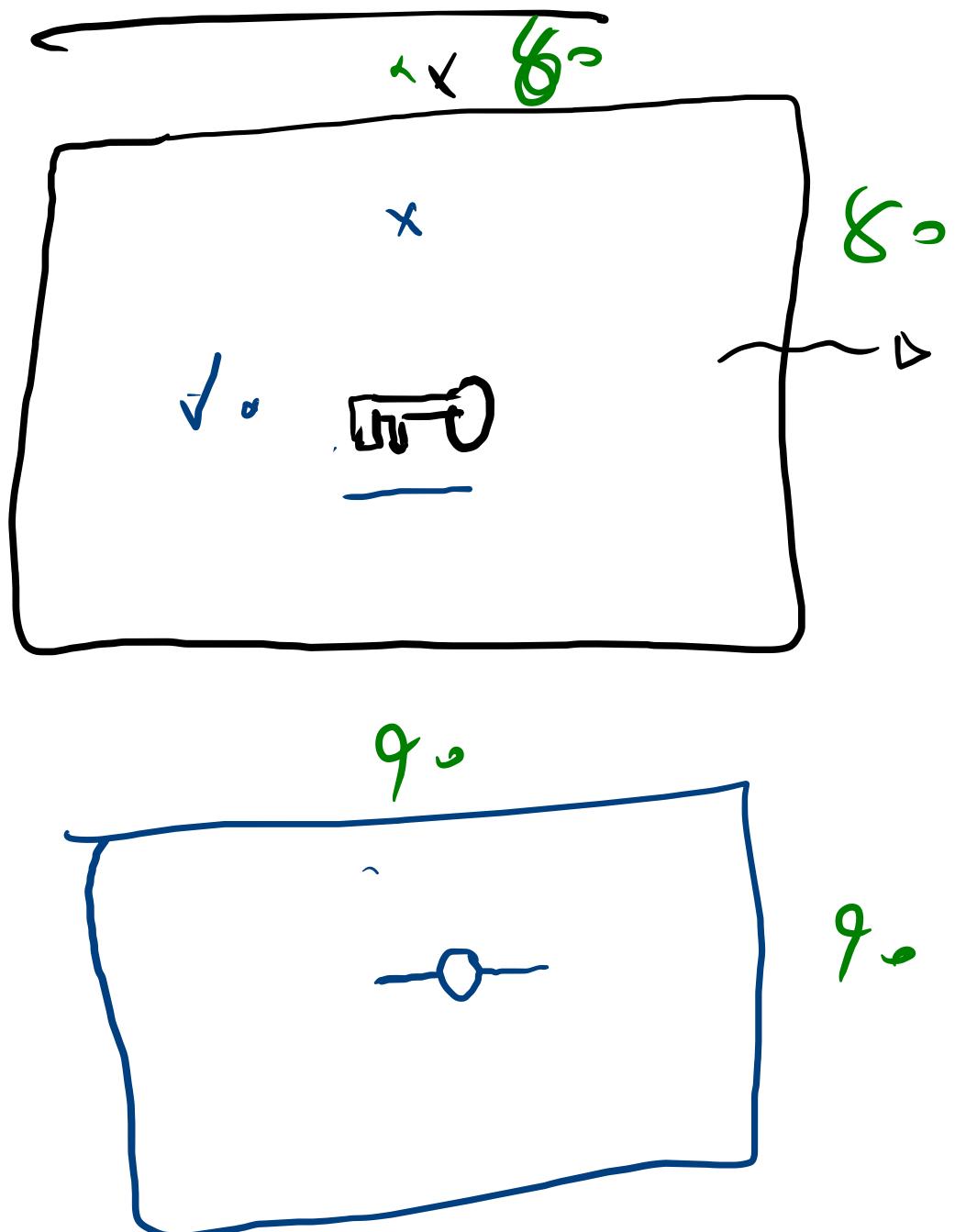


راه حل ها

def. 640

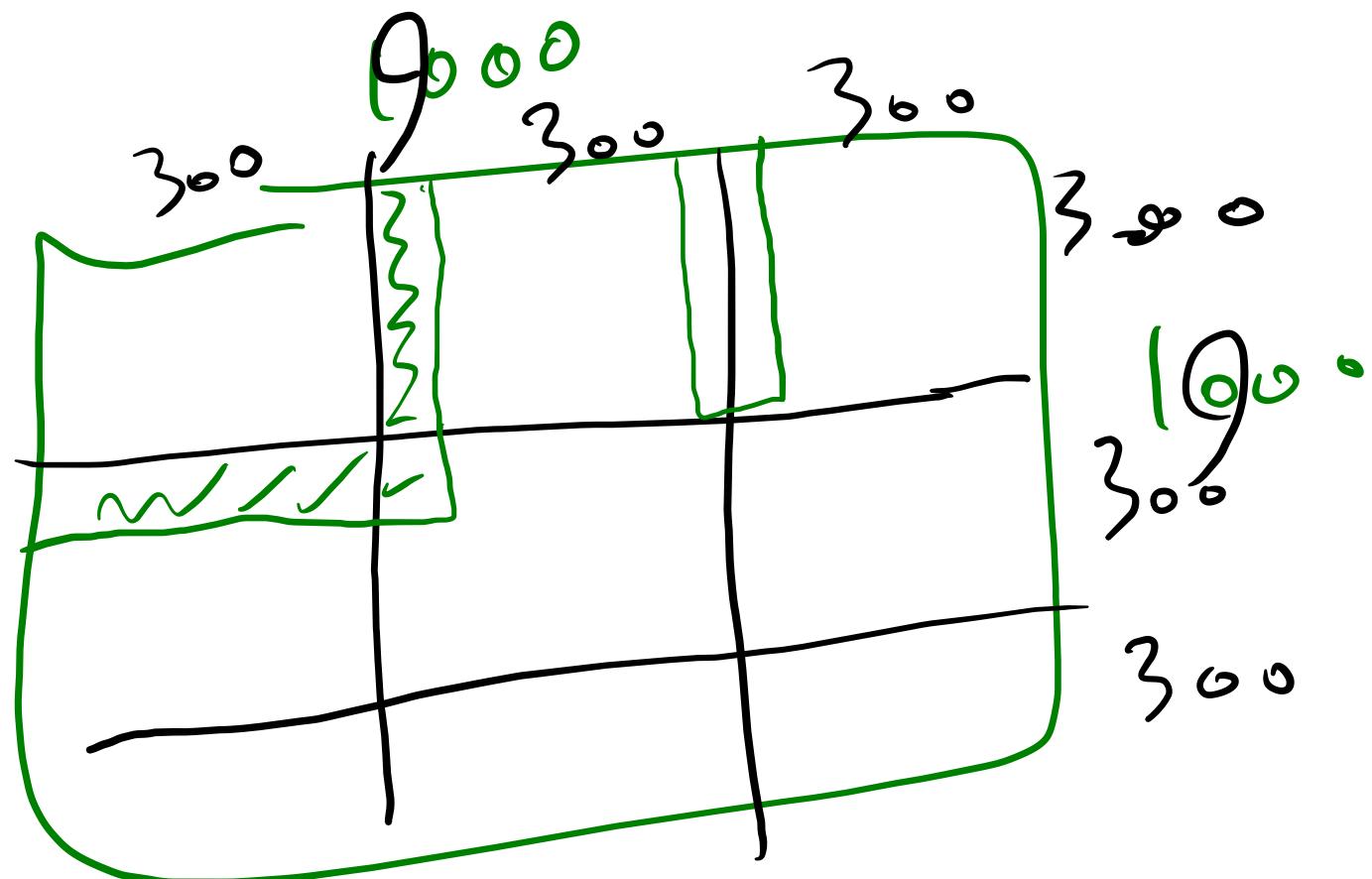
- ✓ افزایش رزولوشن ورودی تصاویر
- ✓ استفاده از مدل های دقیق تر و بزرگتر
- ✓ آموزش با داده های مناسب
- ✓ Mosaic Augmentation
- ✓ Feature Pyramid Networks (FPN)
- ✓ Sliding Window
- ✓ Slicing Aided Hyper Inference(SAHI)

Magic Aug.

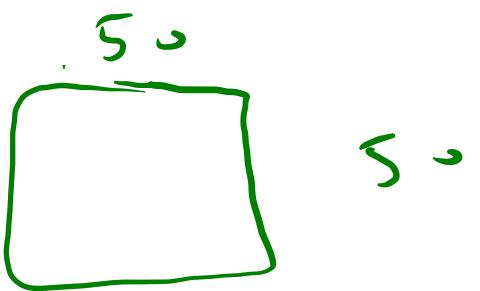
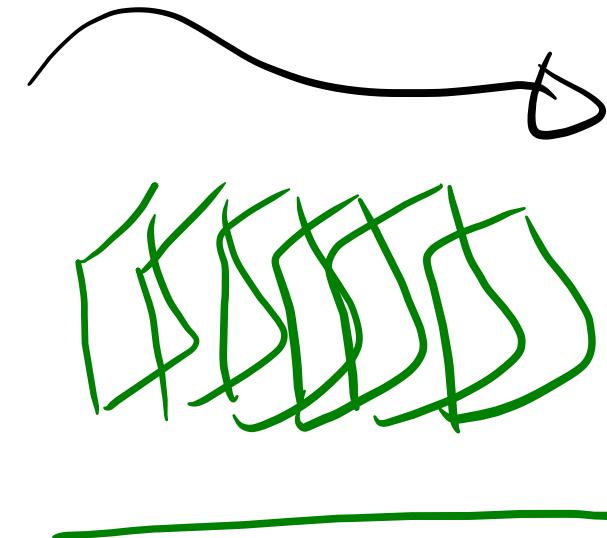


2

Sliding window

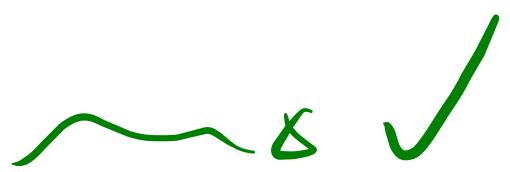


350×350

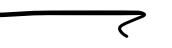


$$\frac{30^\circ}{\text{---}} \quad \frac{80^\circ}{\text{---}} \\ \frac{30^\circ}{8}$$

OD model.



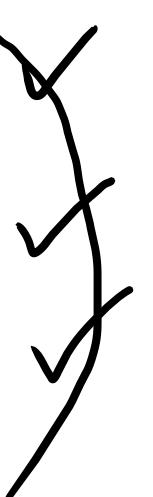
(YOLO، Faster R-CNN و ...)



• محدودیت در Receptive Field

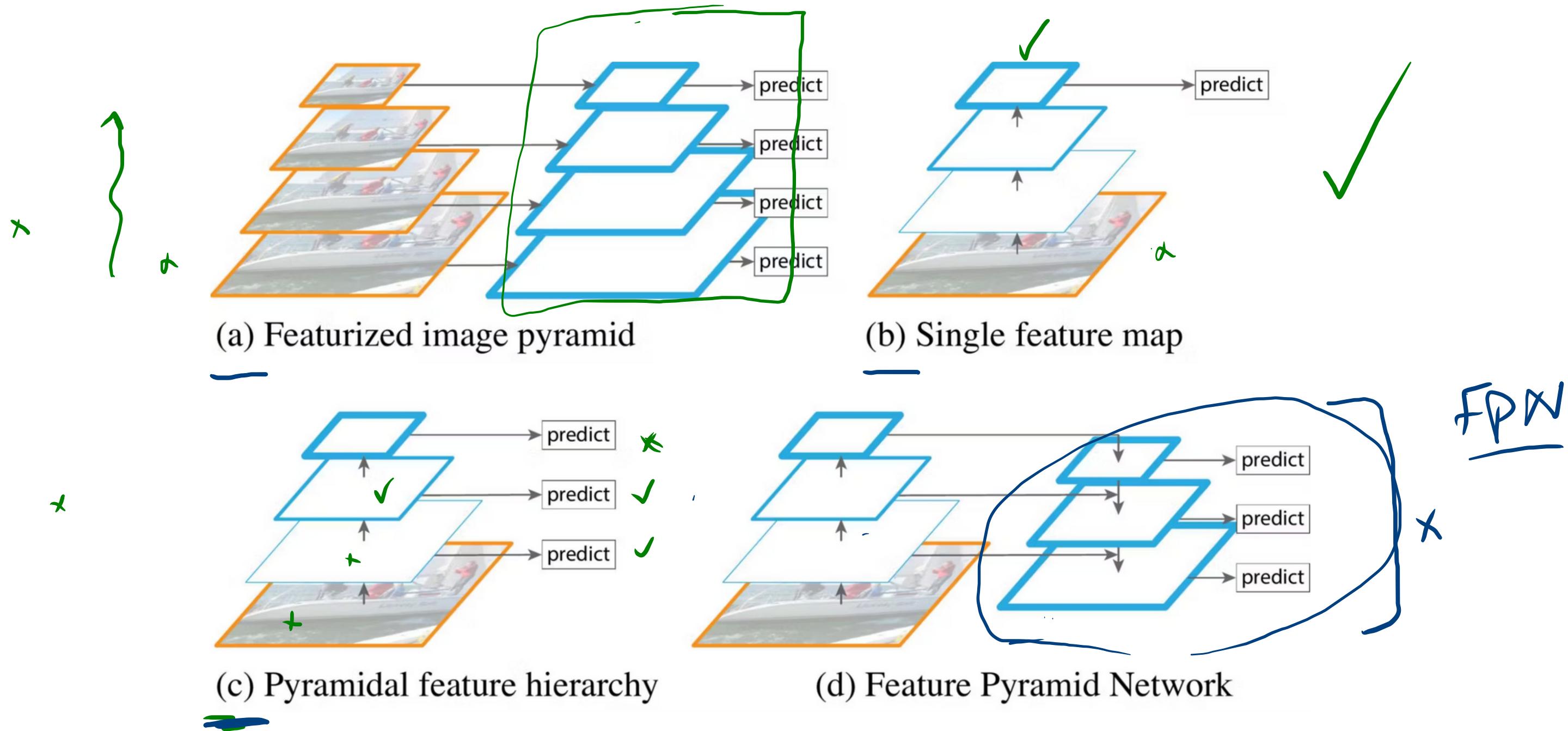
• رزولوشن خیلی پایین

• دیتا ست و کلاس های نامتعادل



FIP

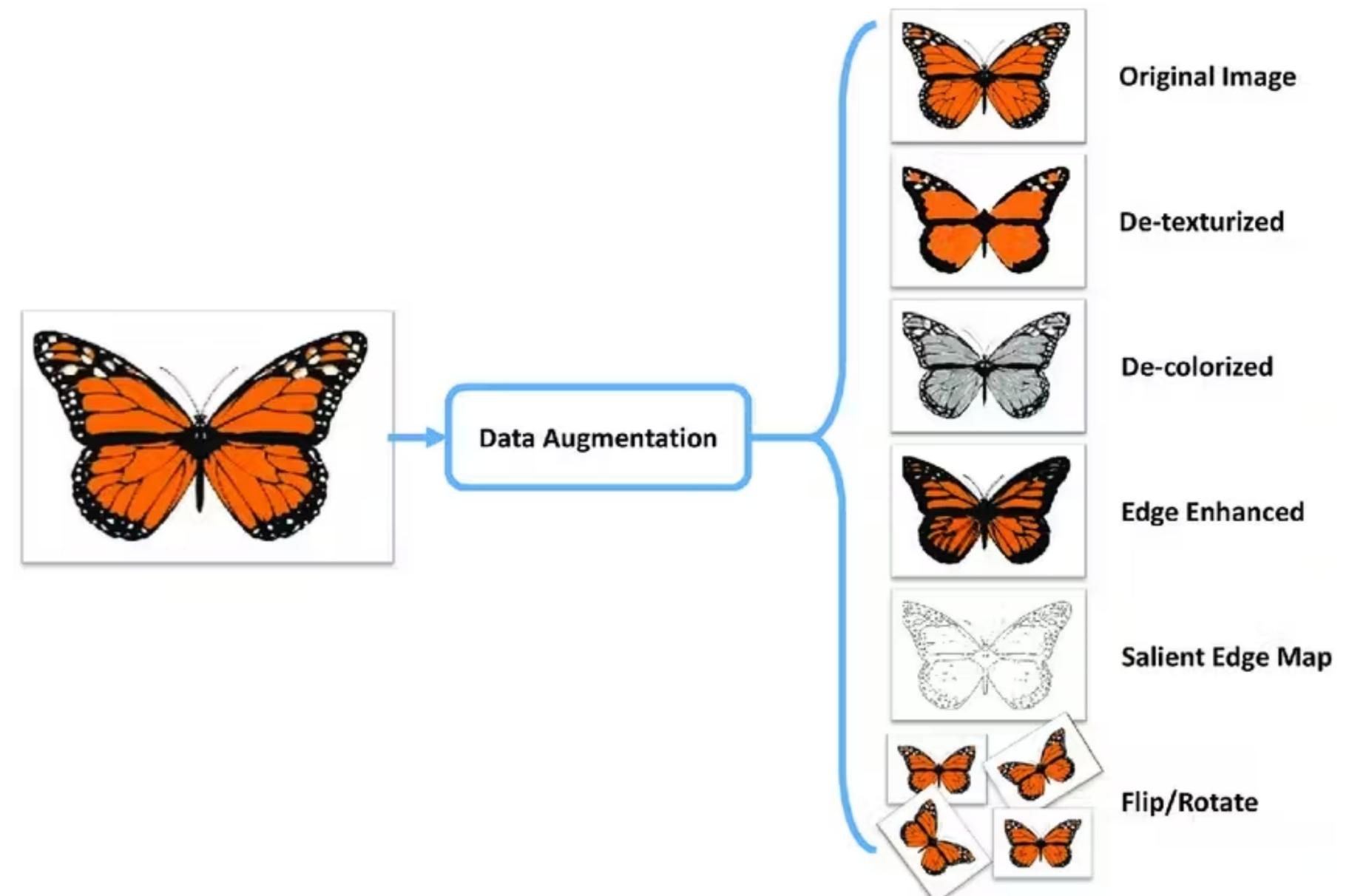
Feature Pyramid Networks (FPN) .



Sliding Window •

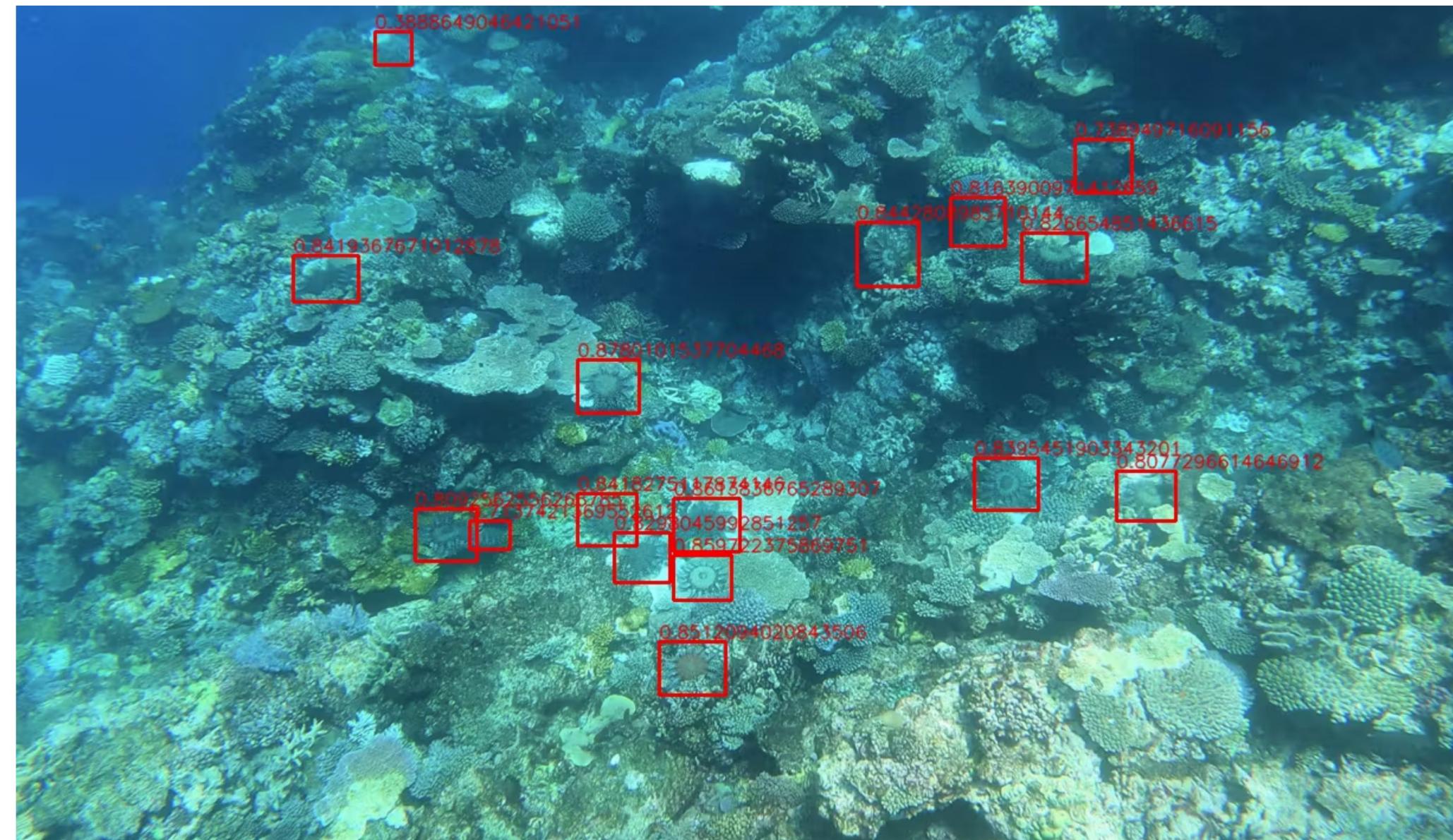


Data Augmentation •



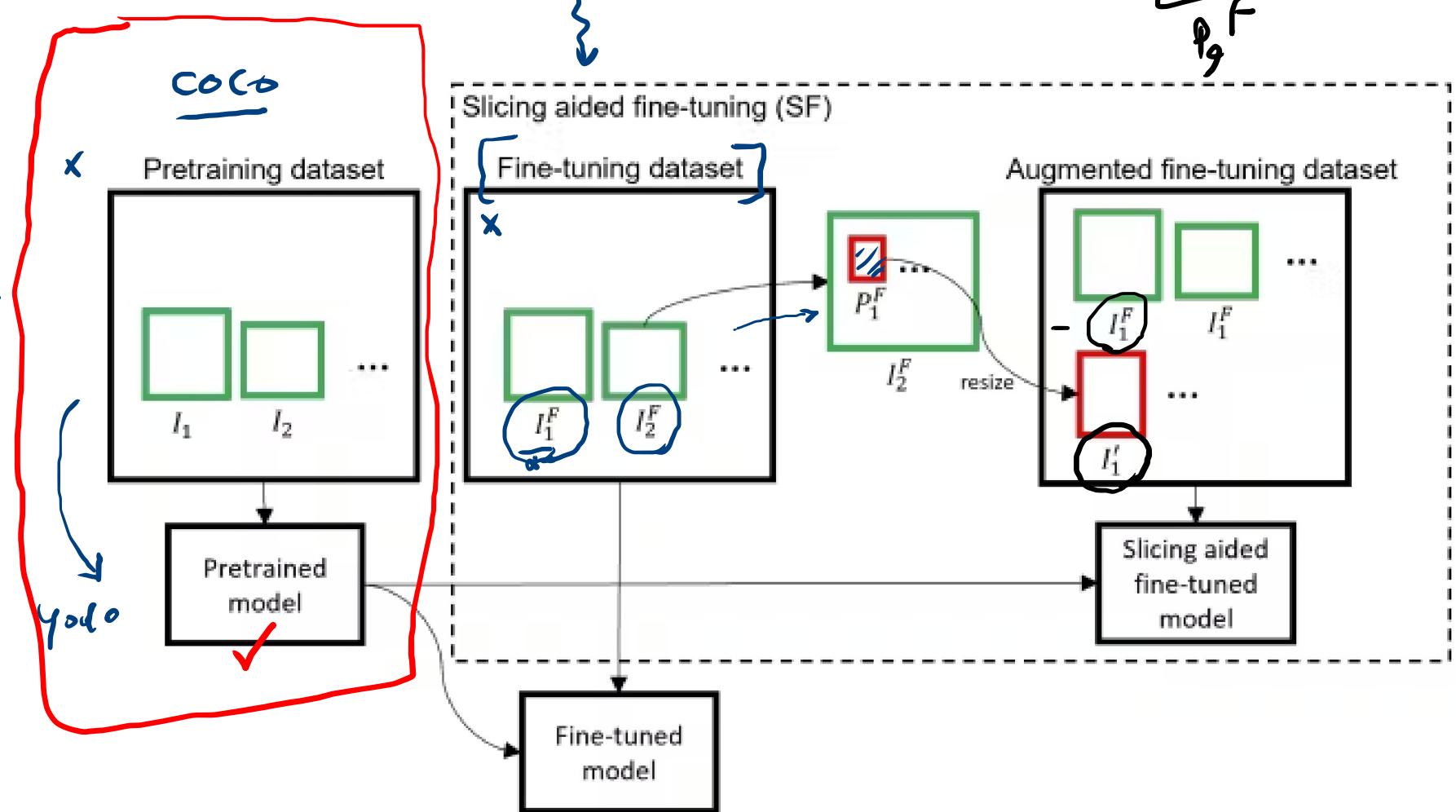
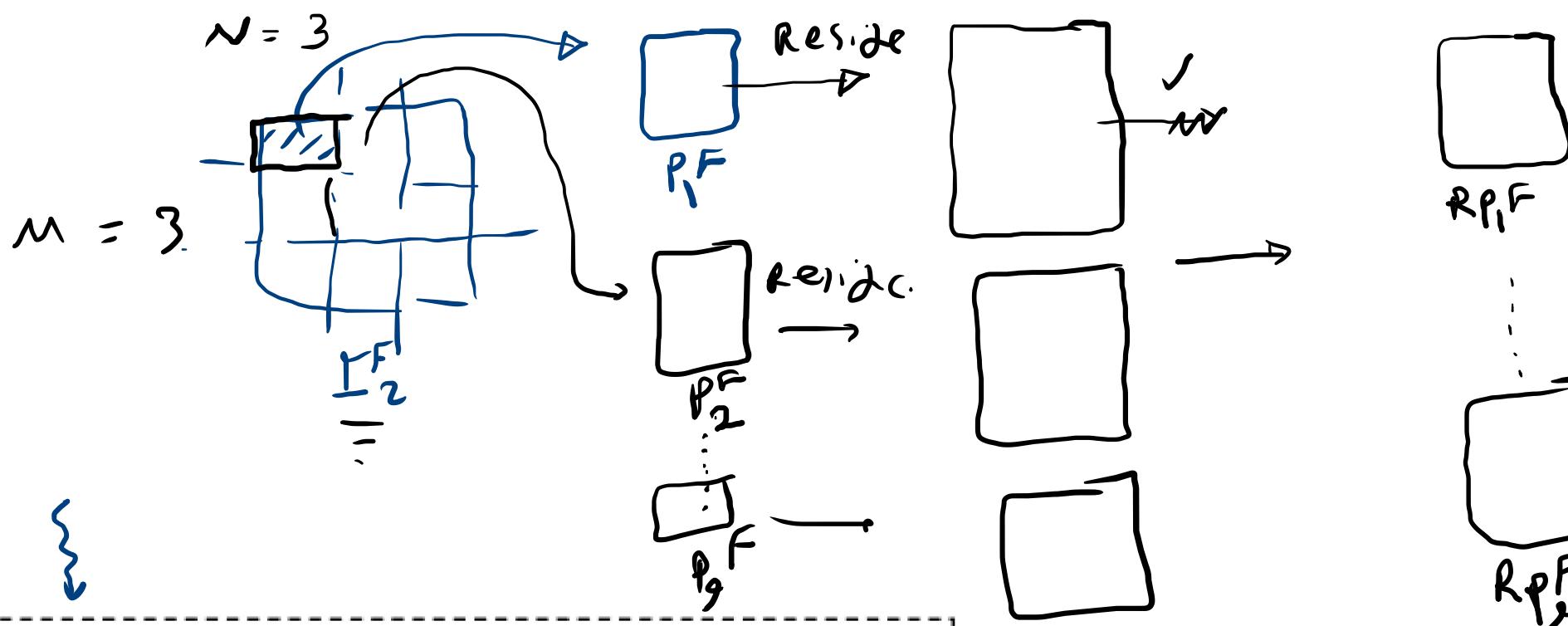
yolo + SAHI

Slicing Aided Hyper Inference (SAHI)



Slicing Aided Hyper Inference (SAHI)

SAHI: Slicing Aided Fine Tuning



تصویر اصلی I به تکه های کوچکی با ابعاد $M \times N$ تقسیم می شود.

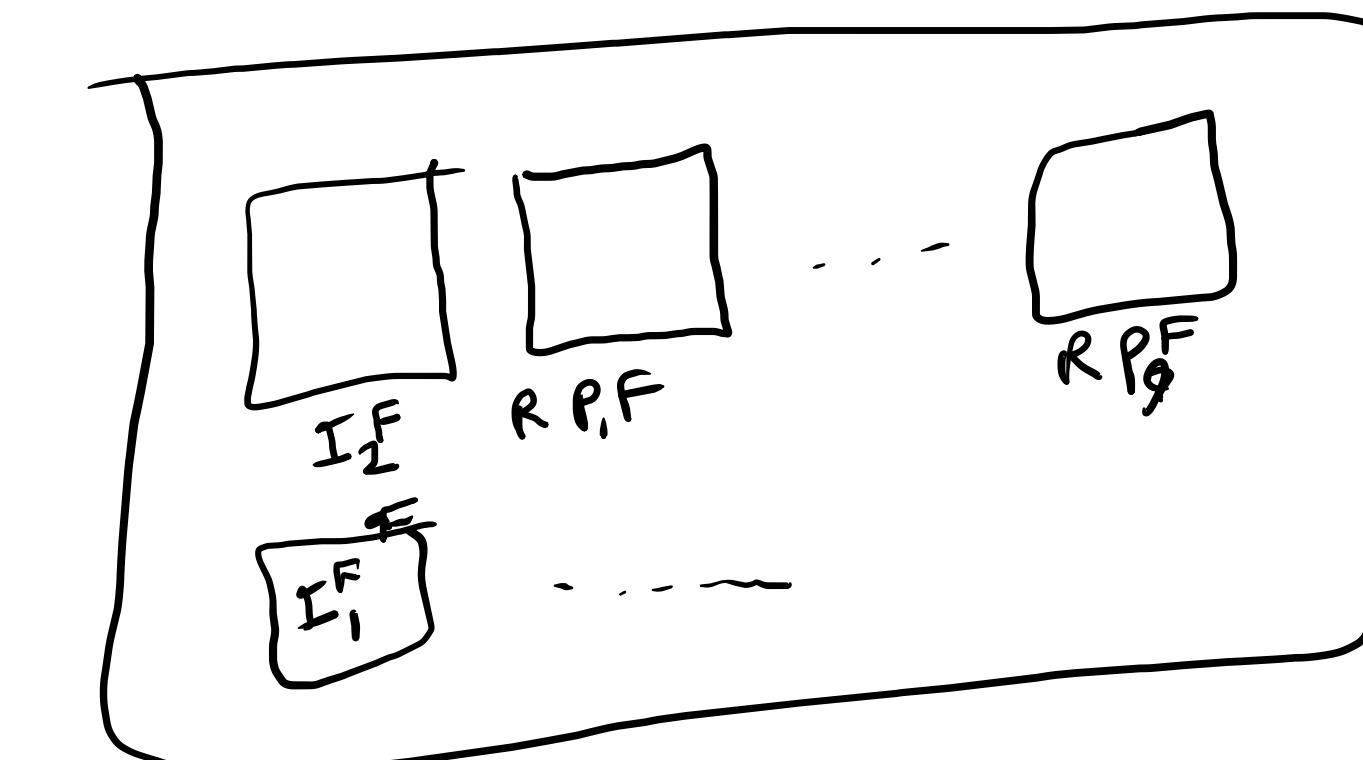
این تکه ها ممکن است همپوشانی داشته باشند تا هیچ بخش مهمی از تصویر از دست نرود.

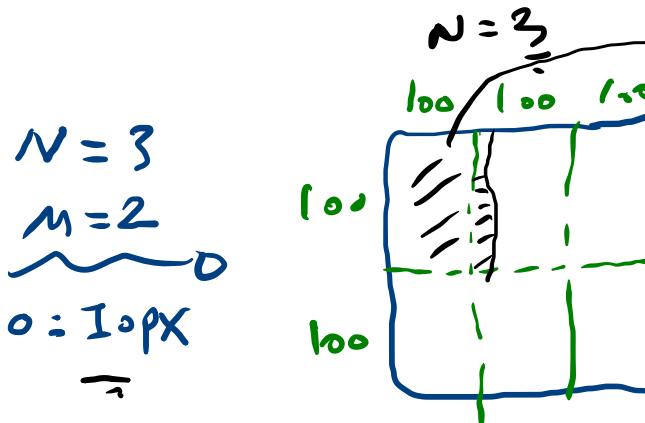
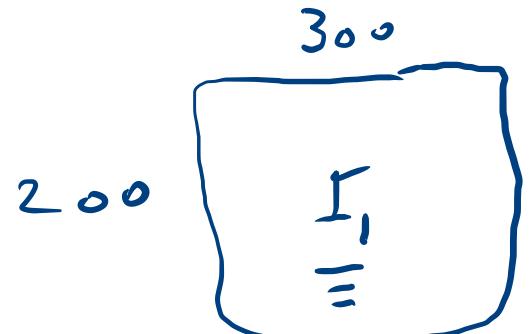
۱

- تغییر اندازه بخش ها در طول

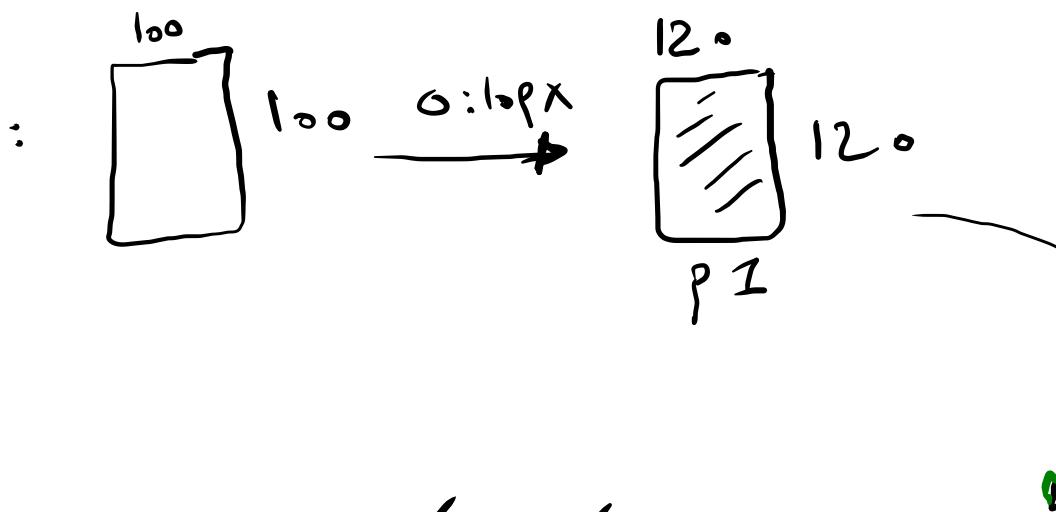
در طول فرآیند fine-tuning، هر تکه P_i به ابعاد جدید $M_p \times N_p$ تغییر اندازه می یابد، اما نسبت ابعاد حفظ می شود.

پس از تغییر اندازه تکه ها، این بخش ها به عنوان تصاویر augmentation مورد استفاده قرار می گیرند. این تصاویر به مدل کمک می کنند تا اشیاء کوچک را بزرگ تر و واضح تر در تصویر ببینند.





$N = 3$
 $M = 2$
 $O = I_0PX$



$$\frac{M}{N} = \frac{M'}{N'}$$

$$\frac{M}{N} = \frac{2}{3} \Rightarrow$$

$$\frac{N}{M} = 1,5$$

$$\frac{N'}{M'} = 1,5$$

$w: 12^o$
 ~~$l = 12^o \times 1,5$~~

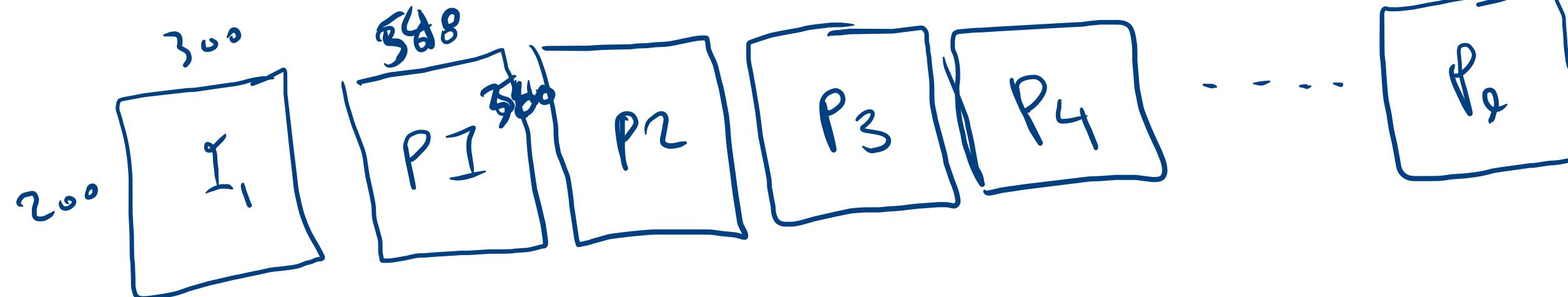
36^o

56^o

$\times 3$

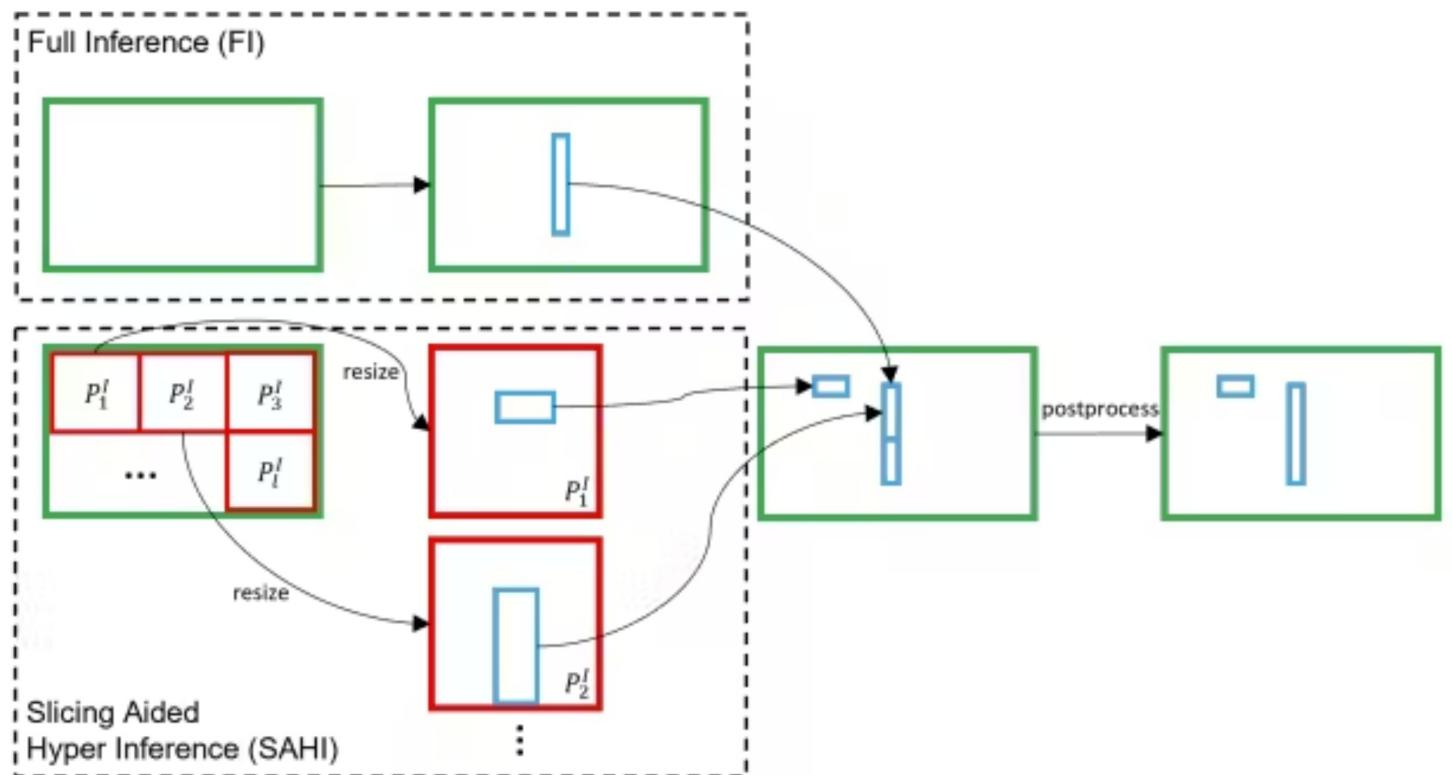
24^o
 $36^o \leftarrow \times 2$

$$\begin{cases} w = 12^o \\ l = 18^o \end{cases}$$



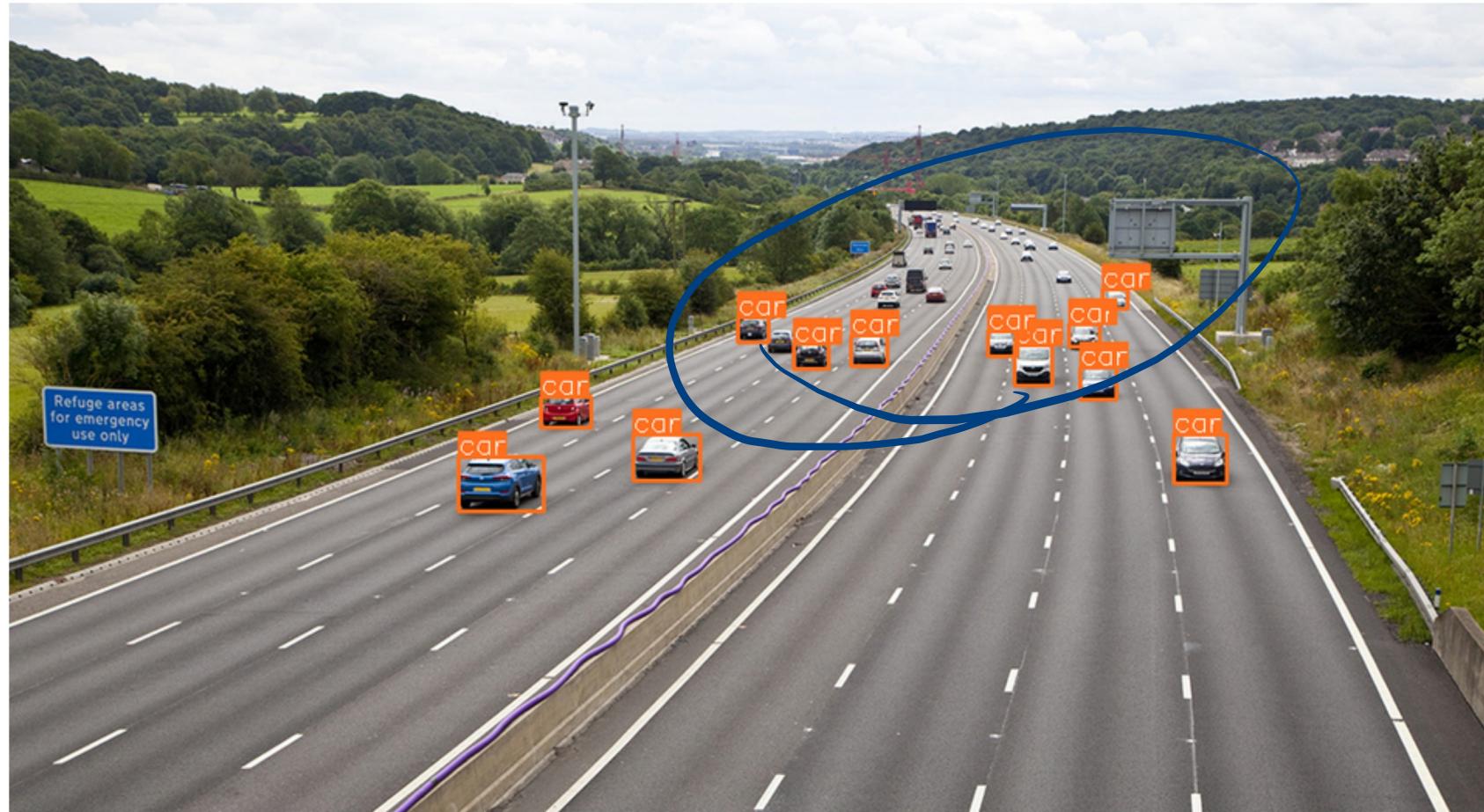
SAHI: Slicing Aided Hyper Inference

- تصویر اصلی I به بخش‌های کوچکتر با ابعاد $M \times N$ تقسیم می‌شود. این بخش‌ها ممکن است همپوشانی داشته باشند تا هیچ بخش مهمی از تصویر از دست نرود.
- هر یک از بخش‌های تقسیم شده P_i به طور جداگانه ریزایز شده و به مدل ارسال می‌شود.
- مدل برای هر تکه پیش‌بینی‌های خود را انجام می‌دهد و به شناسایی اشیاء موجود در آن بخش می‌پردازد.
- پس از پیش‌بینی برای هر بخش، تکنیک NMS (حذف پیش‌بینی‌های تکراری) برای کاهش تعداد پیش‌بینی‌های مشابه و اشتباه اعمال می‌شود.
- چون بخش‌ها همپوشانی دارند، در این مرحله باید نتایج از همپوشانی‌های مختلف تنظیم شوند تا به خروجی دقیق‌تری برسیم.



مُكَبَّرُ
مازِستوا
patch

YOLOv5

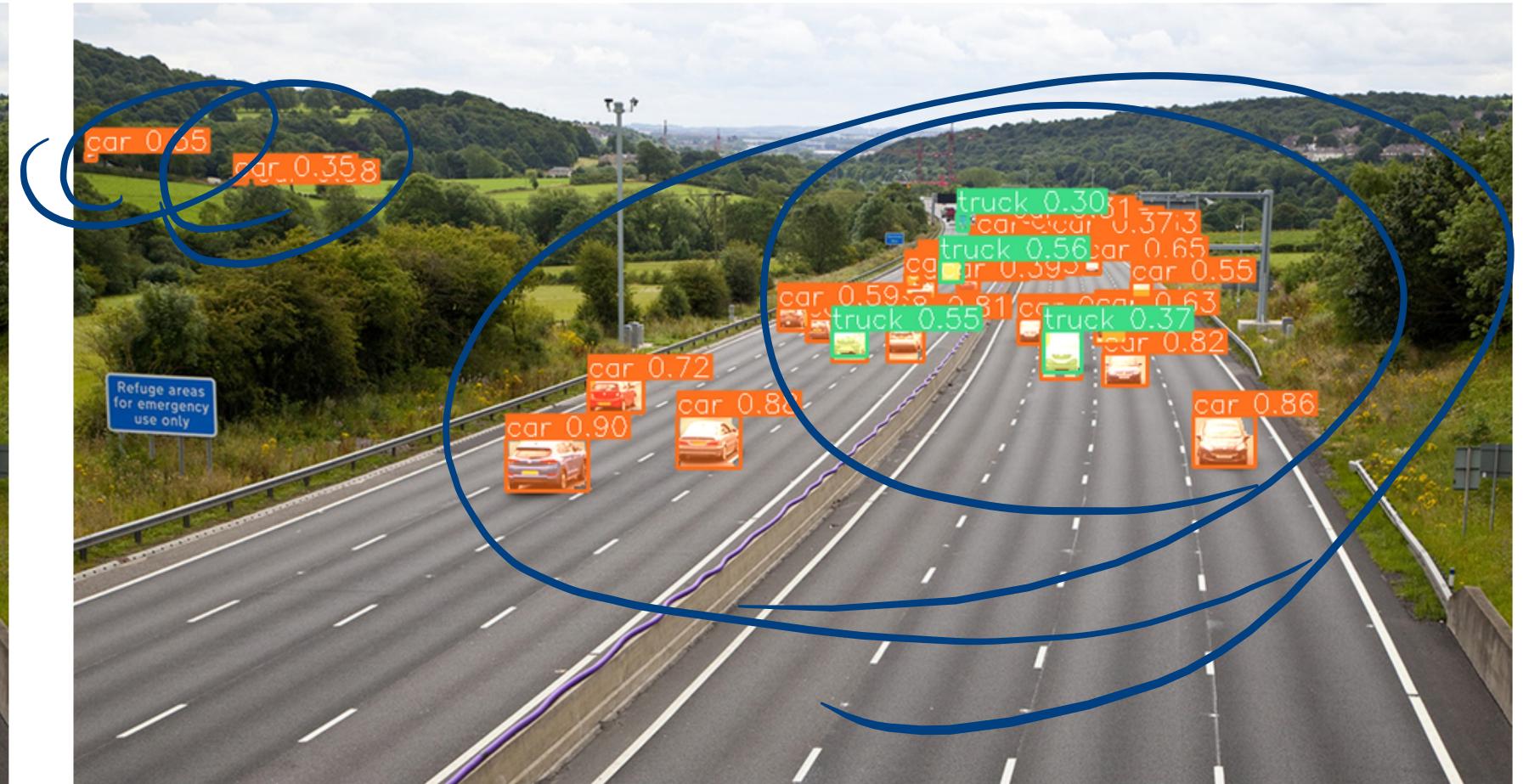


YOLOv5



<https://github.com/obss/sahi?tab=readme-ov-file>

<https://huggingface.co/spaces/fcakyon/sahi-yolox>



YOLOv5 + SAHI

وقتی SAHI استفاده می کنیم.

Setup	AP ₅₀	AP _{50s}	AP _{50m}	AP _{50l}
FCOS+FI ✓	2.20	0.10	1.80	7.30
FCOS+SF+SAHI ✓	15.8	11.9	18.4	11.0
FCOS+SF+SAHI+PO	17.1	<u>12.2</u>	20.2	12.8
FCOS+SF+SAHI+FI	15.7	11.9	18.4	14.3
FCOS+SF+SAHI+FI+PO	17.0	<u>12.2</u>	20.2	15.8
VFNet+FI	2.10	0.50	1.80	6.80
VFNet+SF+SAHI	16.0	11.9	17.6	13.1
VFNet+SF+SAHI+PO	17.7	<u>13.7</u>	19.7	15.4
VFNet+SF+SAHI+FI	15.8	<u>11.9</u>	17.5	15.2
VFNet+SF+SAHI+FI+PO	17.5	<u>13.7</u>	19.6	17.6
TOOD+FI	2.10	0.10	2.00	5.20
TOOD+SF+SAHI	19.4	14.6	22.5	14.2
TOOD+SF+SAHI+PO	20.6	<u>14.9</u>	23.6	17.0
TOOD+SF+SAHI+FI	19.2	14.6	22.3	14.7
TOOD+SF+SAHI+FI+PO	20.4	<u>14.9</u>	23.5	17.6

بریم سراغ کدش!

SAHI