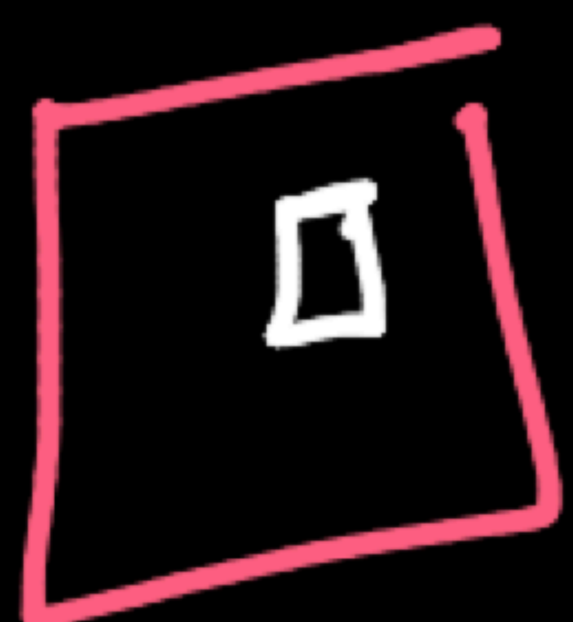
Topics:

- Efficient-Net



- object localization + Detection
  (basic Model)
- R-CNN
- Fast R-CNN
- Faster R-CNN

YOLO

ResNets $\longrightarrow$ $\boxed{\text{depth}}$

MobileNets

Conv-dw

1x1 conv

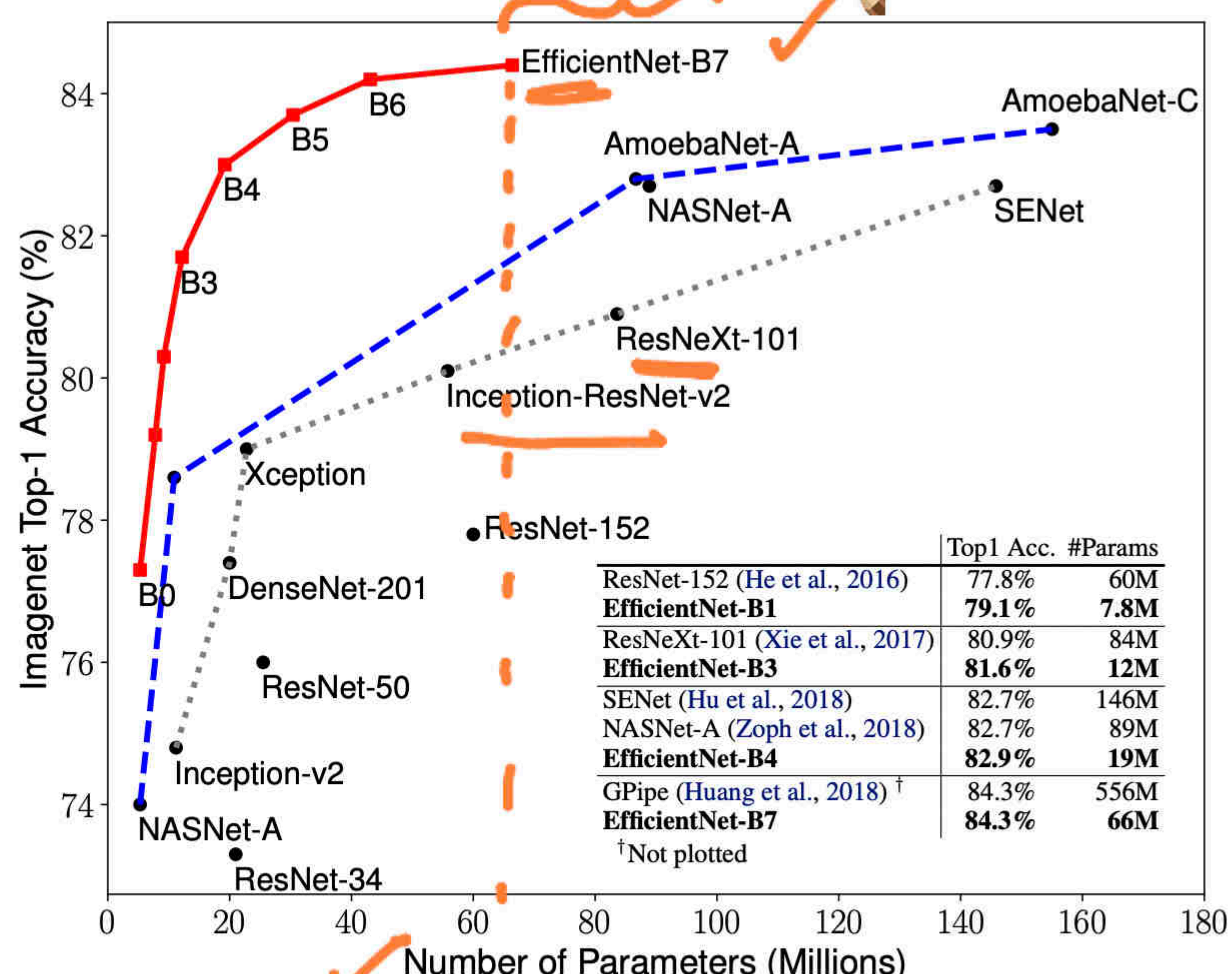

#channels
wider
deeper
layer_i
resolution HxW
higher resolution
deeper
higher resolution
(a) baseline
(b) width scaling
(c) depth scaling
(d) resolution scaling
(e) compound scaling

Mingxing Tan Quoc V. Le

Abstract

Neural Networks (ConvNets) are
eloped at a fixed resource budget,
d up for better accuracy if more
available. In this paper, we sys-
dy model scaling and identify that
cing network depth, width, and res-
ad to better performance. Based
ation, we propose a new scaling
niformly scales all dimensions of
solution using a simple yet highly
ound coefficient. We demonstrate
ss of this method on scaling up
d ResNet.

urther, we use neural architec-
design a new baseline network
o to obtain a family of



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

Figure 1. **Model Size vs. ImageNet Accuracy.** All numbers are
ntly out-

Figure 3. Scaling Up a Baseline Model with Different Network Width (w), Depth (d), and Resolution (r) Coefficients. Bigger networks with larger width, depth, or resolution tend to achieve higher accuracy, but the accuracy gain quickly saturate after reaching 80%, demonstrating the limitation of single dimension scaling. Baseline network is described in Table 1.

- EfficientNet models generally use an order of magnitude fewer parameters and FLOPS than other ConvNets with similar accuracy
- They Came up with eights models B0 to B7 (with increasing size)
- For example, **EfficientNet-B3 achieves higher accuracy than ResNet101 using 18x fewer FLOPS.**
- Full efficientnet code from scracth: https://github.com/qubvel/efficientnet/blob/f7f3e736c113b872caf53dae9fbbda996a8eb87d/efficientnet/model.py

Loading pretrained efficientnet:

https://keras.io/api/applications/efficientnet/

In fact, a few prior work (Zoph et al., 2018; Real et al., 2019) have already tried to arbitrarily balance network width and depth, but they all require tedious manual tuning.

In this paper, we propose a new **compound scaling method**, which use a compound coefficient ϕ to uniformly scales network width, depth, and resolution in a principled way:

$$
\begin{aligned}
\text{depth: } & d = \alpha^{\phi} \\
\text{width: } & w = \beta^{\phi} \\
\text{resolution: } & r = \gamma^{\phi} \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}
\tag{3}
$$

where α, β, γ are constants that can be determined by a small grid search. Intuitively, ϕ is a user-specified coeffi-cient that cont

for model scal

extra resources to network width, depth, and resolution re-

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Net, except our EfficientNet-B0 is slightly bigger due to the larger FLOPS target (our FLOPS target is 400M). Table 1 shows the architecture of EfficientNet-B0. Its main building block is mobile inverted bottleneck MBConv (San-

2019), to which we also add squeeze-and-excitation optimization (Hu et al., 2018).

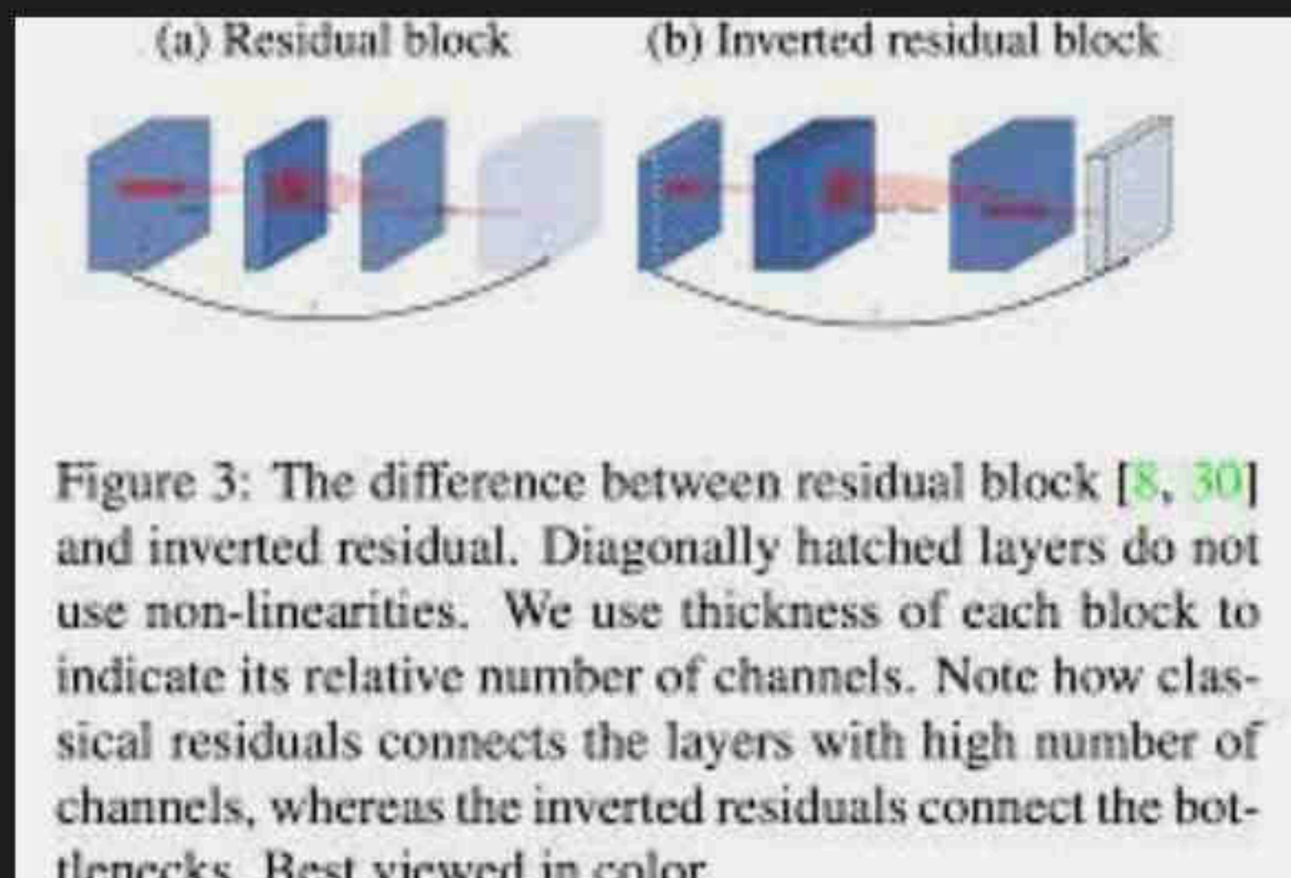
Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:45

L6_CNN_for_Medical_Diagnos | EfficientNet: Rethinking Model | L7: Object Detection with Two | mbconv block - Google Search

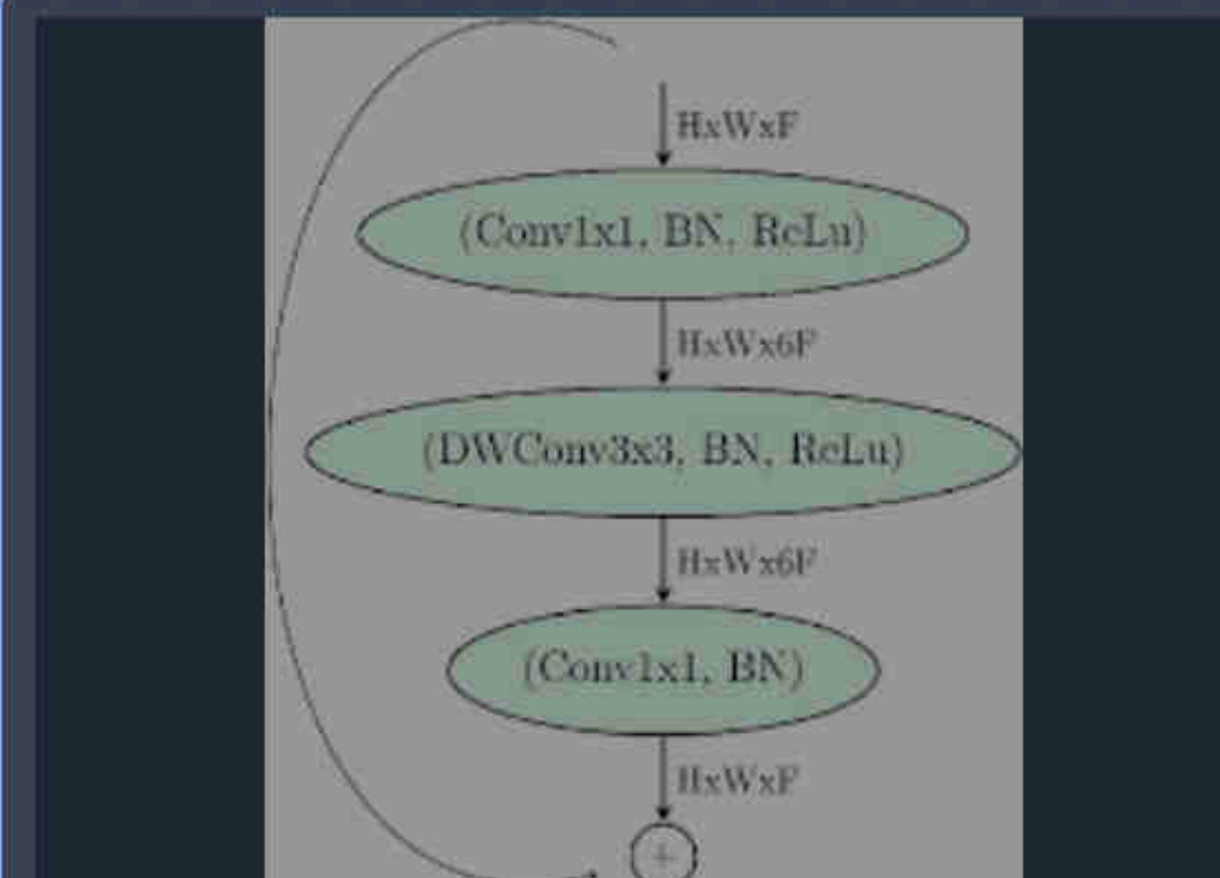
google.com/search?q=mbconv+block&rlz=1C5CHFA_enIN958IN958&source=lnms&tbm=isch&sa=X&ved=2ahUKEwihmsWhh9b6...

Google

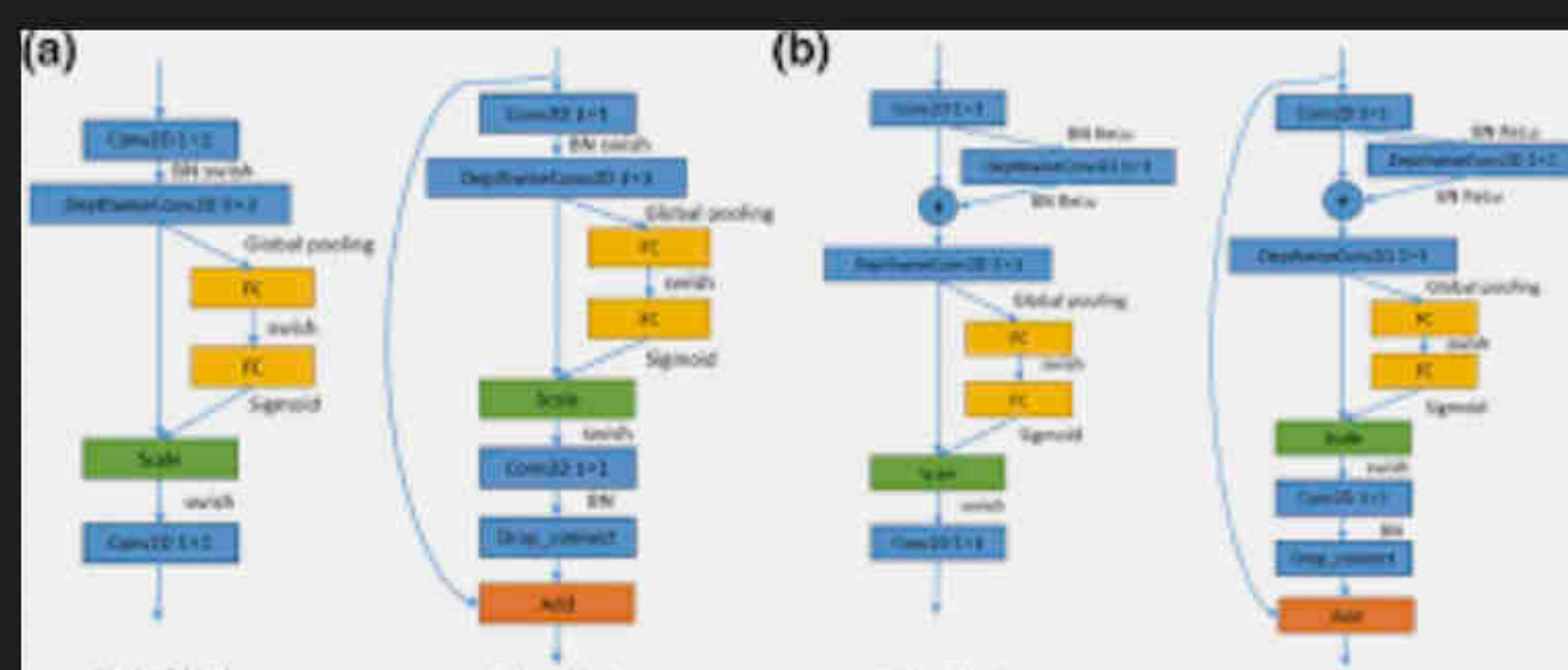
mbconv block



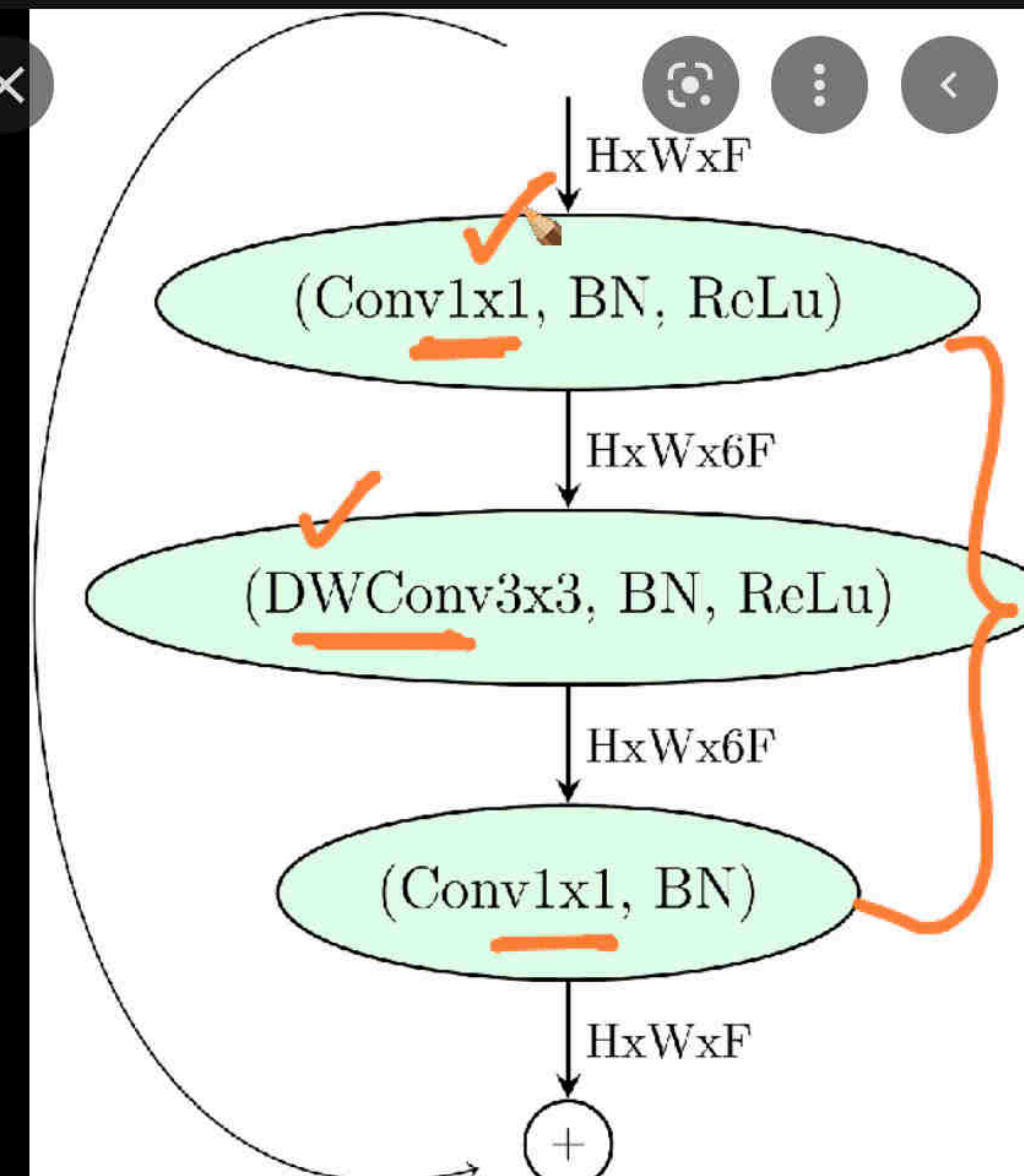
Inverted Residual Block Explained ...
paperswithcode.com



MBConv Block [16]. DWConv stands for ...
researchgate.net



Structure of MBConv module. a Basic ...
researchgate.net



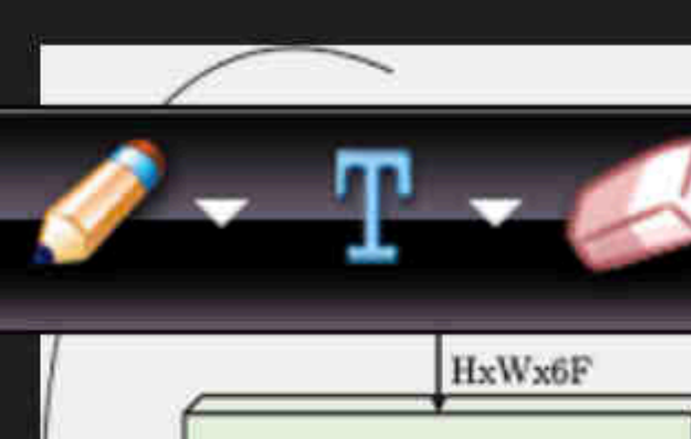
ResearchGate

Visit

...nds for depthwise
...load Scientific...

7 / 8

Images may be subject to copyright. **Learn More**




#channels
wider
deeper
deeper
layer_i
resolution HxW
higher resolution
higher resolution
(a) baseline
(b) width scaling
(c) depth scaling
(d) resolution scaling
(e) compound scaling

2022: CNNs

non-conv based

NLP (late 2018)

early 2021 → Vision Transformers

(later)

Chrome File Edit View History Bookmarks Profiles Tab Window Help 9:51

11 / 11

Summary of pretrained models:

Model Name	Number of params	Top 1 Acc	Top 5 Acc
EfficientnetB0	5.3M	77.3	93.5
MobileNet	2.3M	71.0	90.5
ResNet50	25.6M	83.2	96.5
Inception	22.9M	79.0	94.5
VGG16	138M	74.4	91.9
AlexNet	62M	63.3	84.6

How do you decide which model you should chose on a new dataset?

- If you want to roll your own layers, start with VggNet. It's the simplest model that will perform well.
- For edge devices, you typically want to optimize for models that can be downloaded fast, occupy very little space on the device, and don't incur high latencies during prediction. For a small model that runs fast on low-power devices, con- sider MobileNetV2.
- If you don' ... itoscaling cloud

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:00

(Q) classfn / regrsn:

✓ dataset → small (few thousand)

↳ Imagenet like

↓

✓ Transfer learning → AlexNet ✗

ResNet ✗

Efficient Net B7 ✓ ← MobileNet ✗

12 / 14

Q

EN-BO → 5.3 MM params → 2.5 Sec

MobileNet → 2.3 MM params → 1 sec

Qualcomm

Q

Small dataset (few 1000's)

↓

[data set ≠ ImageNet

→ ① MobileNet all-layers trainable

||

→ ② Some of the last layers trainable

other layers: ImageNet

May / Maynot

Siamese n/w
&
(later)
Contrastive learning

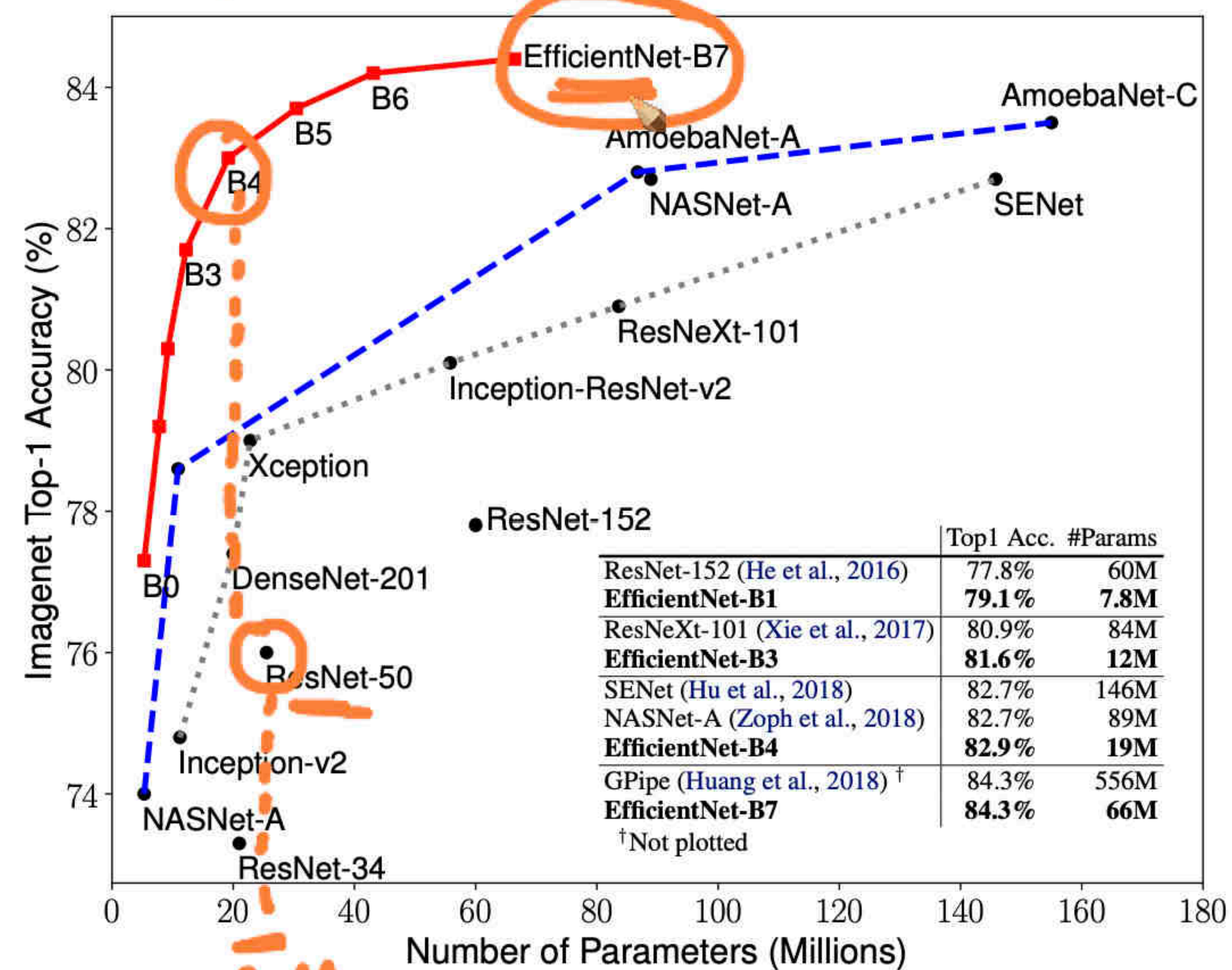
Mingxing Tan [1] **Quoc V. Le** [1]

Case 1

Abstract

Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. In this paper, we systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, we propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective *compound coefficient*. We demonstrate the effectiveness of this method on scaling up MobileNets and ResNet.

To go even further, we use neural architecture search to design a new baseline network and scale ... called *E*... better accuracy and efficiency than previous



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

Figure 1. **Mo... Size vs. ImageNet Accuracy.** All numbers are ... ur EfficientNets significantly out- ... icular, EfficientNet-B7 achieves new state-of-the-art 84.3% top-1 accuracy but being 8.4x smaller and 6.1x faster than GPipe. EfficientNet-B1 is 7.6x smaller and

(Q)

Train

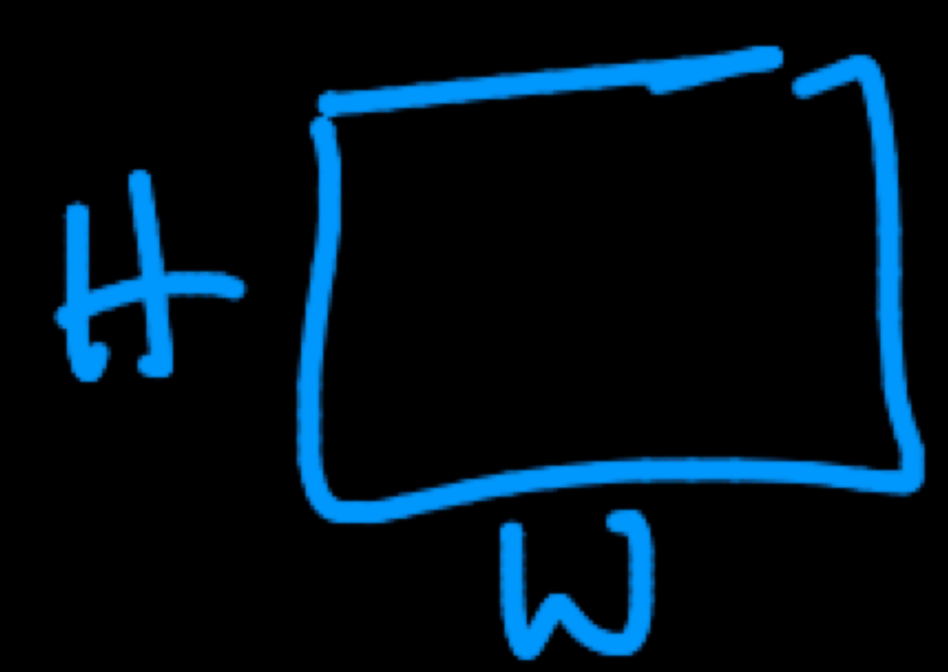
{ color: RGB $\boxed{M}$ → trained

[Query: → grayscale → RGB

OpenCV

M_{RGB}

M_{Gr}



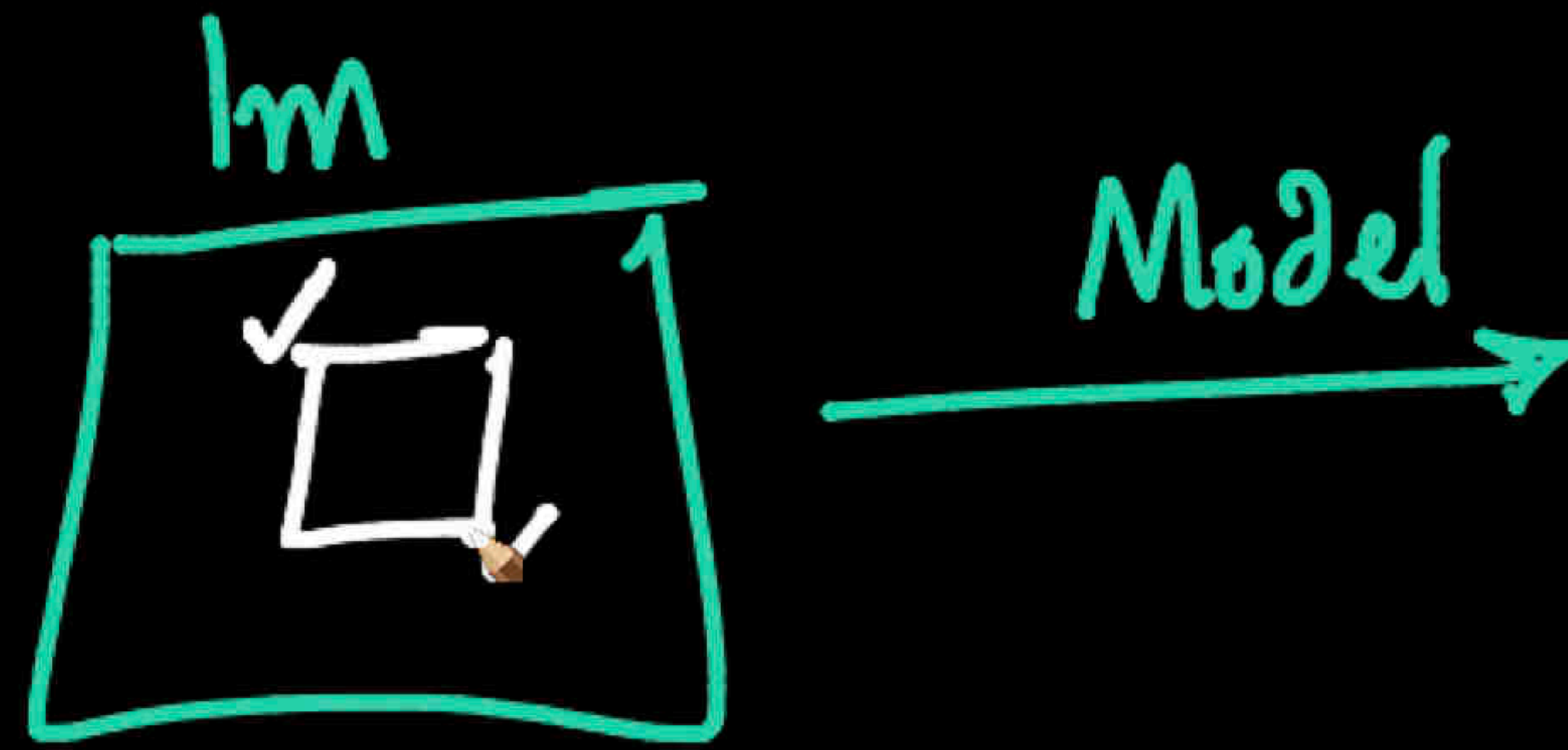
to this dataset as well.

- New dataset is large and similar to the original dataset.
  - Since we have more data, we can have more confidence that we won't overfit if we were to try to fine-tune through the full network.
- New dataset is large and very different from the original dataset. Imagenet
  - Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch. EN B7
  - However, in practice it is very often still beneficial to initialize with weights from a pretrained model.

Supplementary material

Other models to follow :-

Q



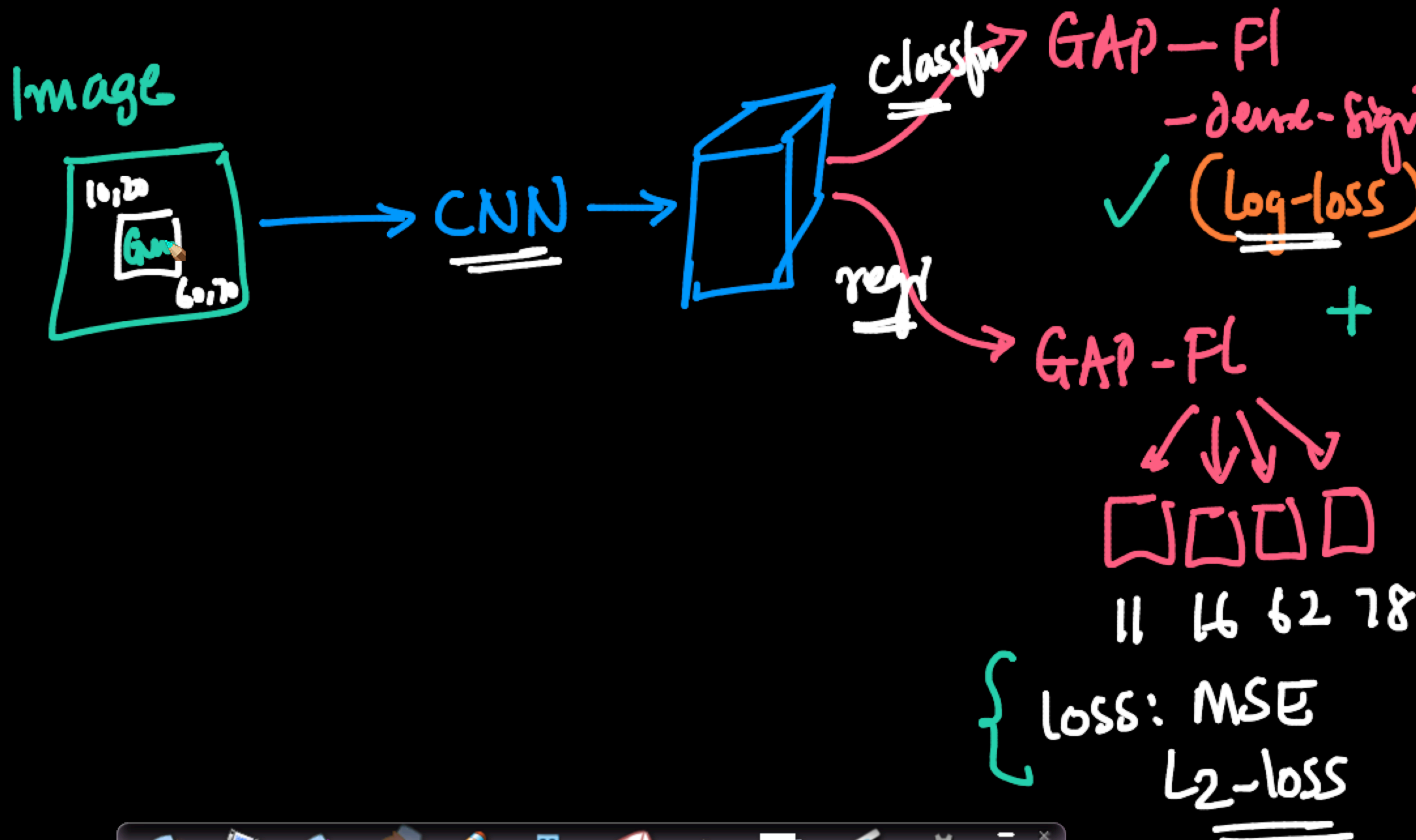
Is Gun?

class fn

Co-ordinates of BBox

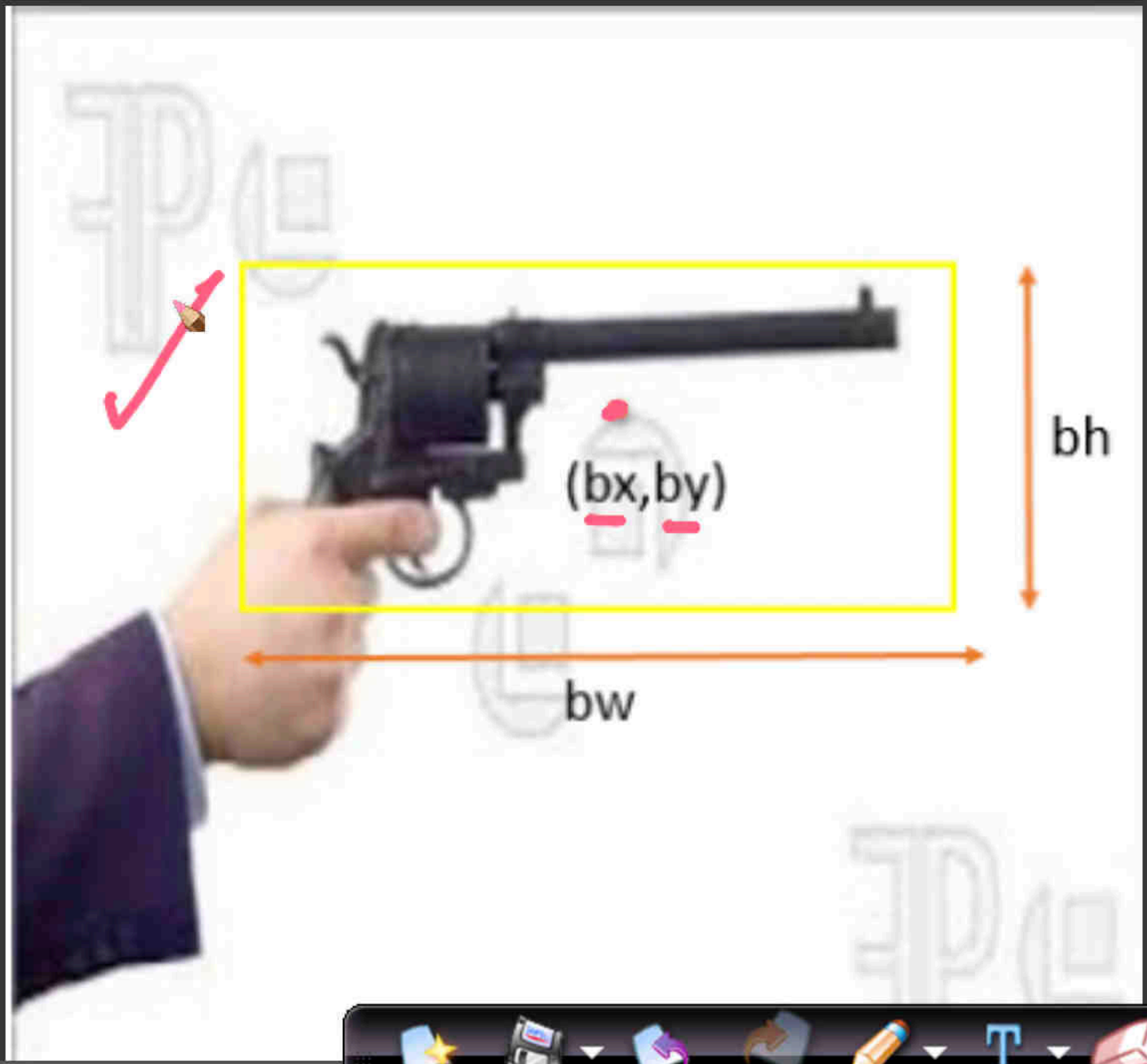
Regr

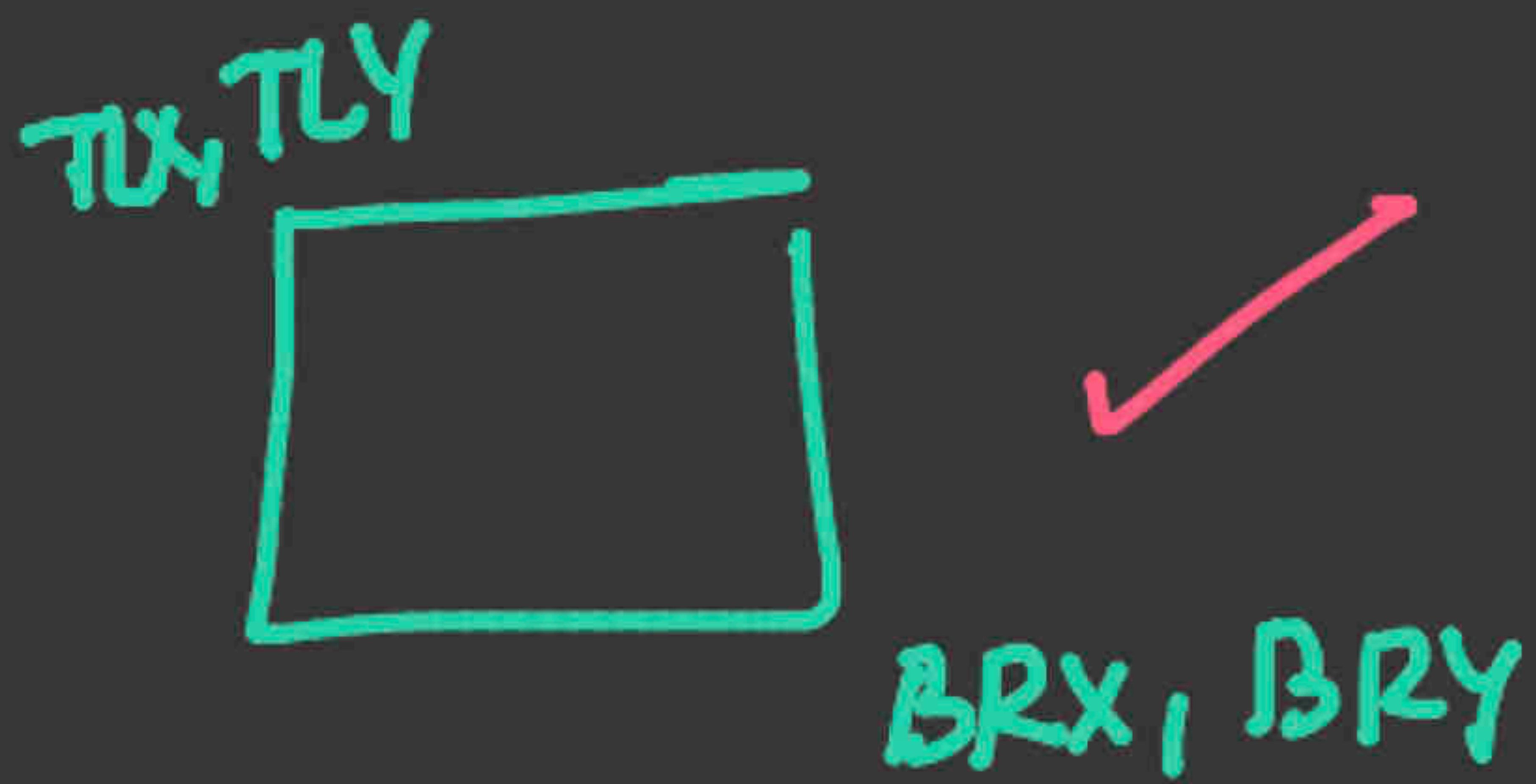
Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:23



19 / 19

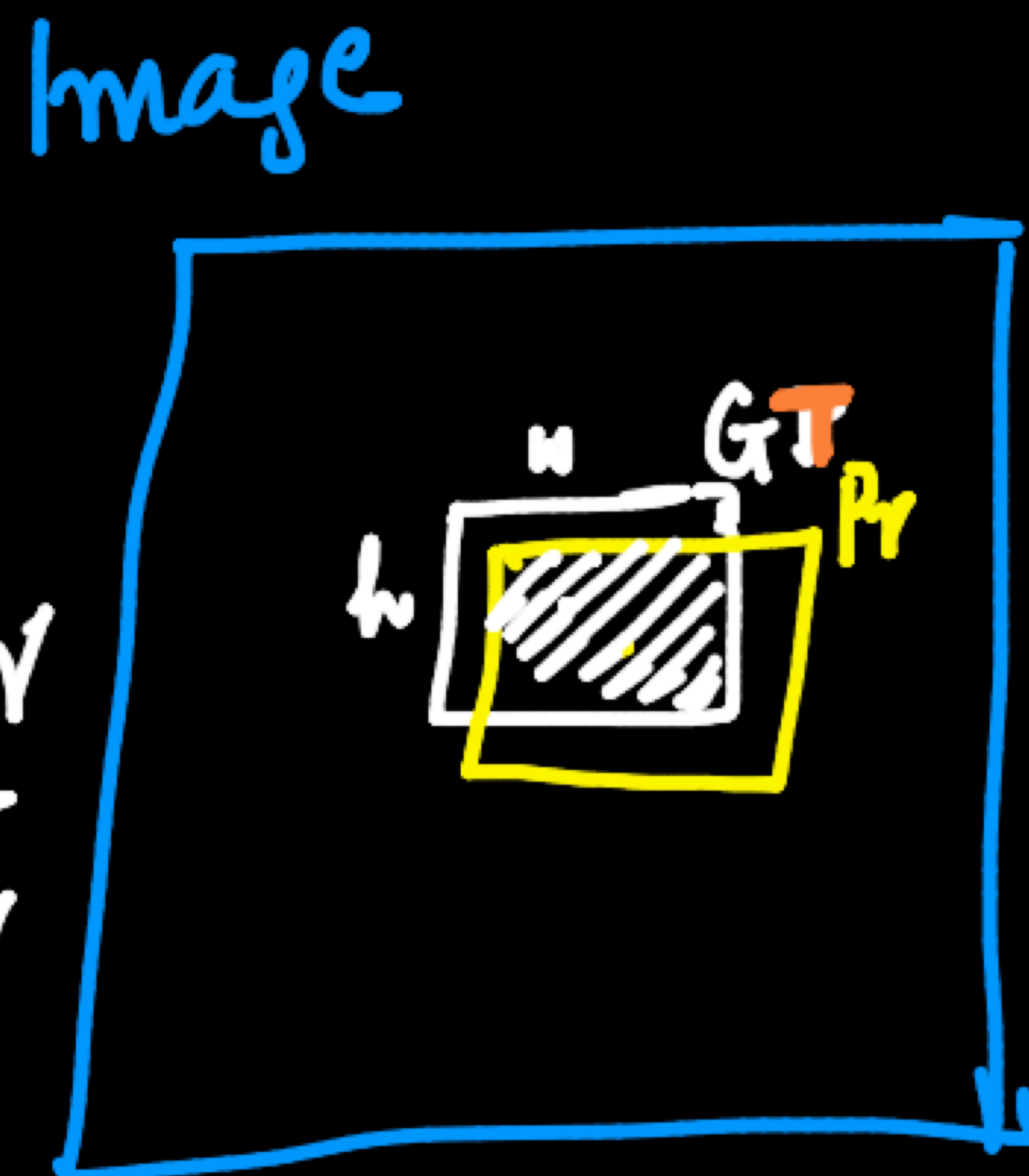
[1 0.5061538461538462 0.4229559748427673 0.3553846153846154 0.3050314465408805]





Q

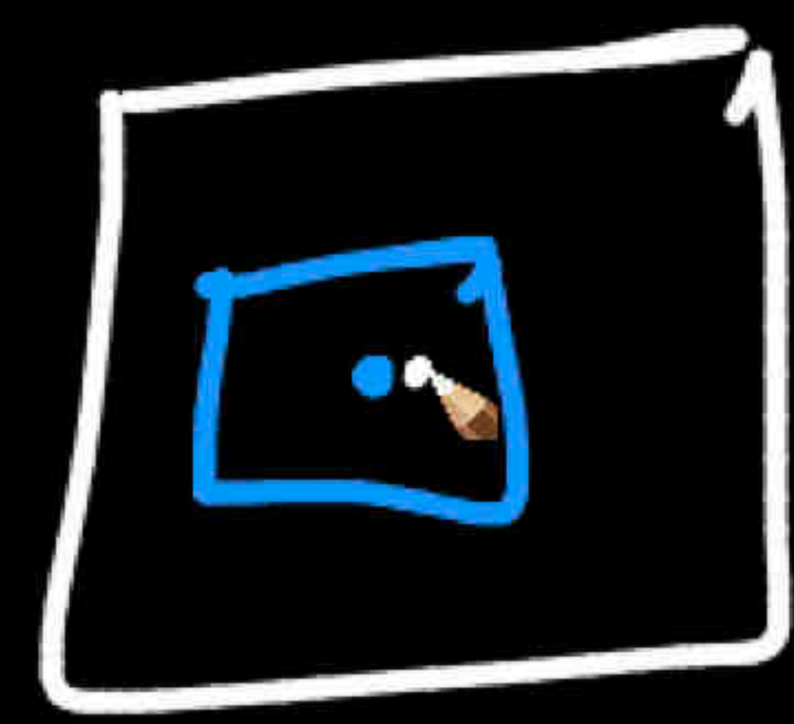
$$IoU = \frac{GT \cap Pr}{GT \cup Pr}$$



measure:

fully overlapping = high → 1

less overlapping = low → 0



Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:38

L6_CNN_for_Medical_Diagnos | EfficientNet: Rethinking Model | L7: Object Detection with Two | mbconv block - Google Search

colab.research.google.com/drive/1MTvGvZPDhYaHhf3fkCyfgyFpaHqqAQO1#scrollTo=oTuVuGKjT455

+ Code + Text Connect

Locate Multiple Objects

5 guns → 5x4

2 guns → 2x4

7 guns → 7x4

23 / 23

Chrome File Edit View History Bookmarks Profiles Tab Window Help

L6_CNN_for_Medical_Diagnos | EfficientNet: Rethinking Model | L7: Object Detection with Two | mbconv block - Google Search

colab.research.google.com/drive/1MTvGvZPDhYaHhf3fkCyfgyFpaHqqAQO1#scrollTo=oTuVuGKjT455

+ Code + Text Connect

24 / 24

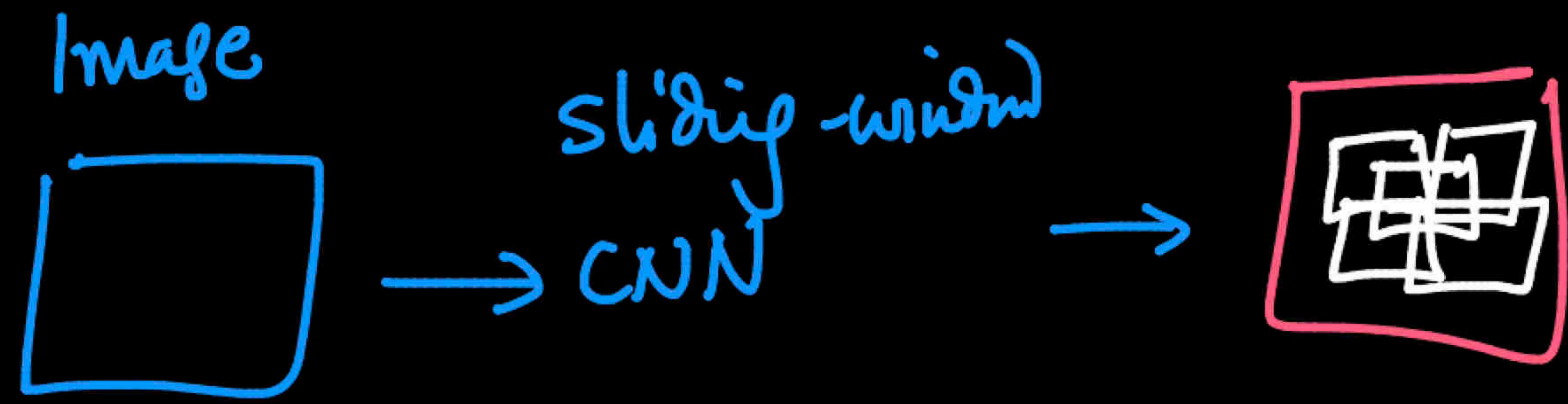
Sliding windows + run CNN multiple

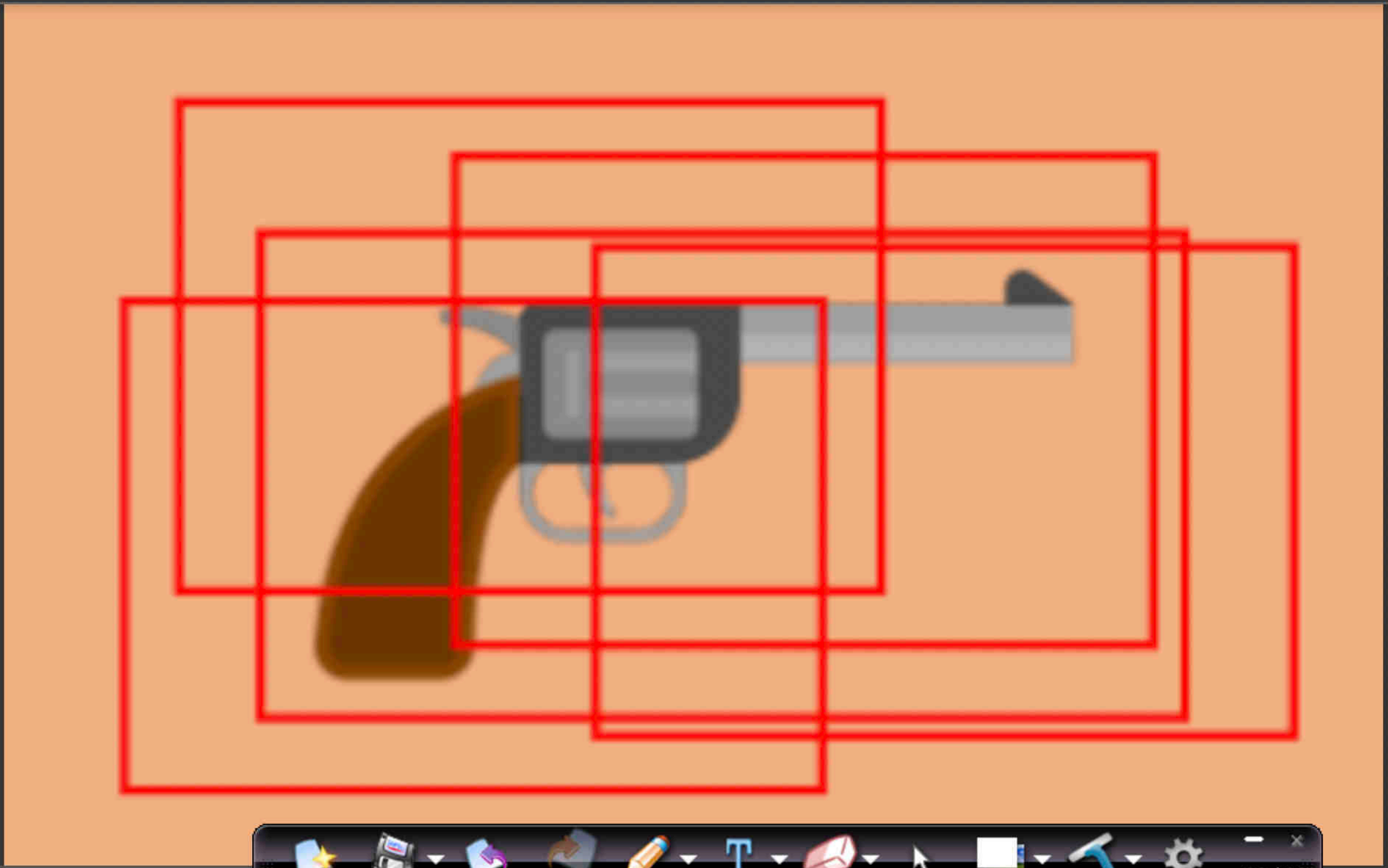
CNN


Chrome File Edit View History Bookmarks Profiles Tab Window Help
10:42
L6_CNN_for_Medical_Diagnos
EfficientNet: Rethinking Model
L7: Object Detection with Two
mbconv block - Google Search
colab.research.google.com/drive/1MTvGvZPDhYaHhf3fkCyfgyFpaHqqAQO1#scrollTo=oTuVuGKjT455
+ Code
+ Text
Connect
(?)
27 / 27

Q

runtime





10 rect

p(gun)
score


Chrome File Edit View History Bookmarks Profiles Tab Window Help
10:47
L6_CNN_for_Medical_Diagnos
EfficientNet: Rethinking Model
L7: Object Detection with Two
mbconv block - Google Search
colab.research.google.com/drive/1MTvGvZPDhYaHhf3fkCyfgyFpaHqqAQO1#scrollTo=oTuVuGKjT455
+ Code
+ Text
Connect
0.9
R_1
0.7
$\tau = 0.5$
0.3
10 BBOX
30 / 30

Chrome File Edit View History Bookmarks Profiles Tab Window Help 10:56

3

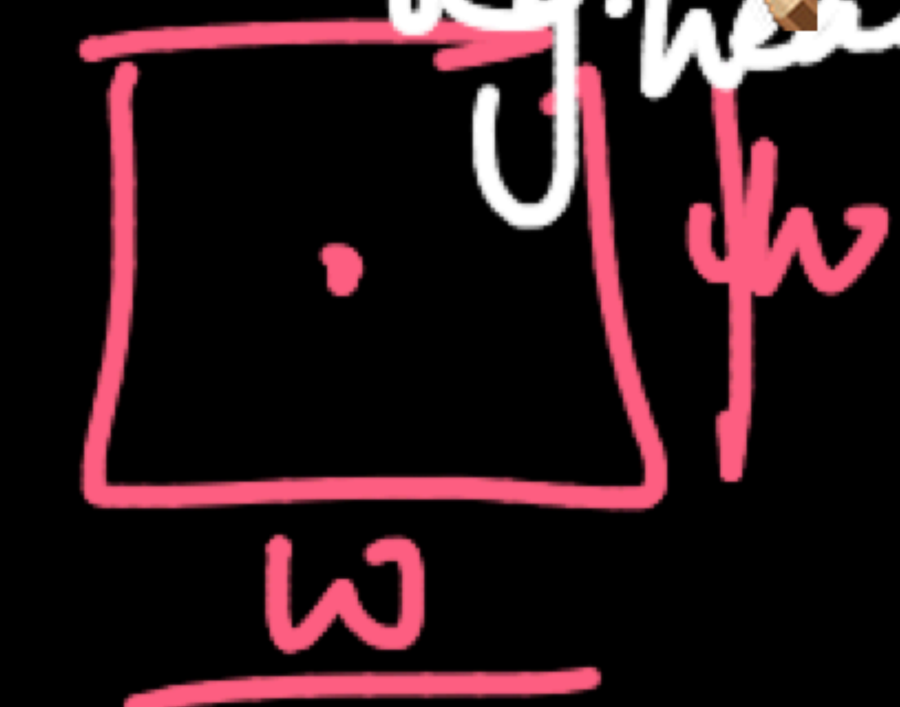
1. BBOX (reg) + object-det (classfn)

2. sliding-window

3. NMS ✓

IoU: metric for BBOX

31 / 33



$$\mathcal{L}oss = \alpha \, \underline{\log \text{loss}} + \beta \, \underline{L_2\text{-loss}}$$

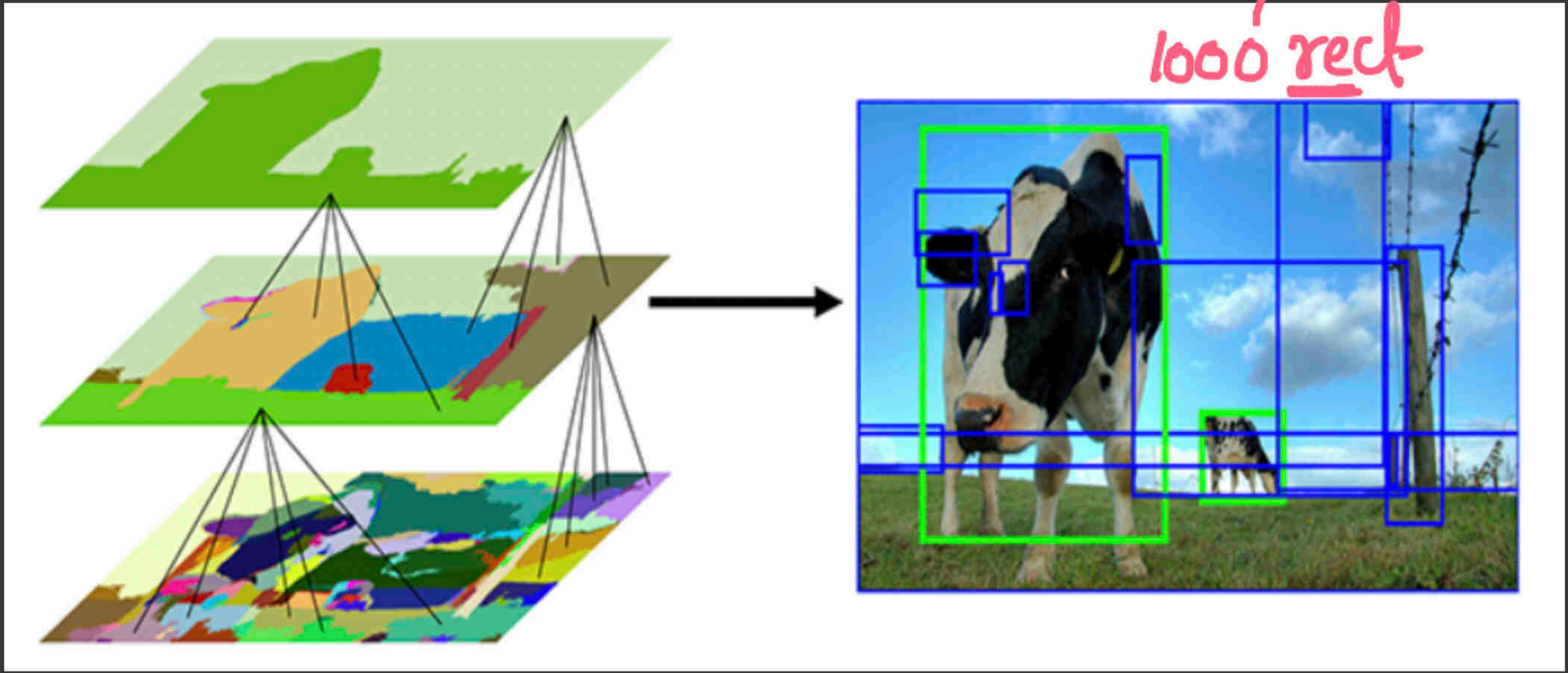

Chrome
File
Edit
View
History
Bookmarks
Profiles
Tab
Window
Help
10:56
33 / 33

2. Then at second stage, the Regions proposed by first stage are classified into different classes.

algo

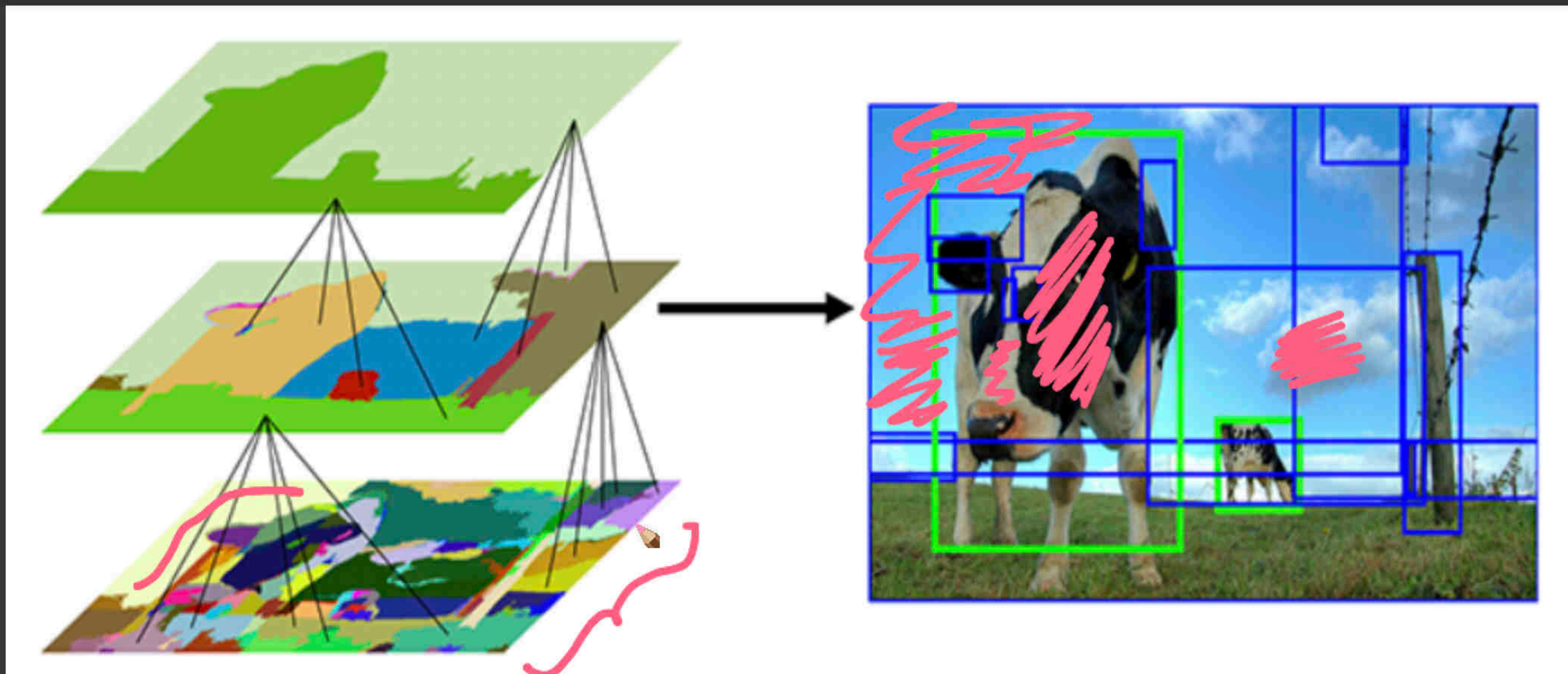
https://learnopencv.com/selective-search-for-object-detection-cpp-python/

1000 rect → CNN --

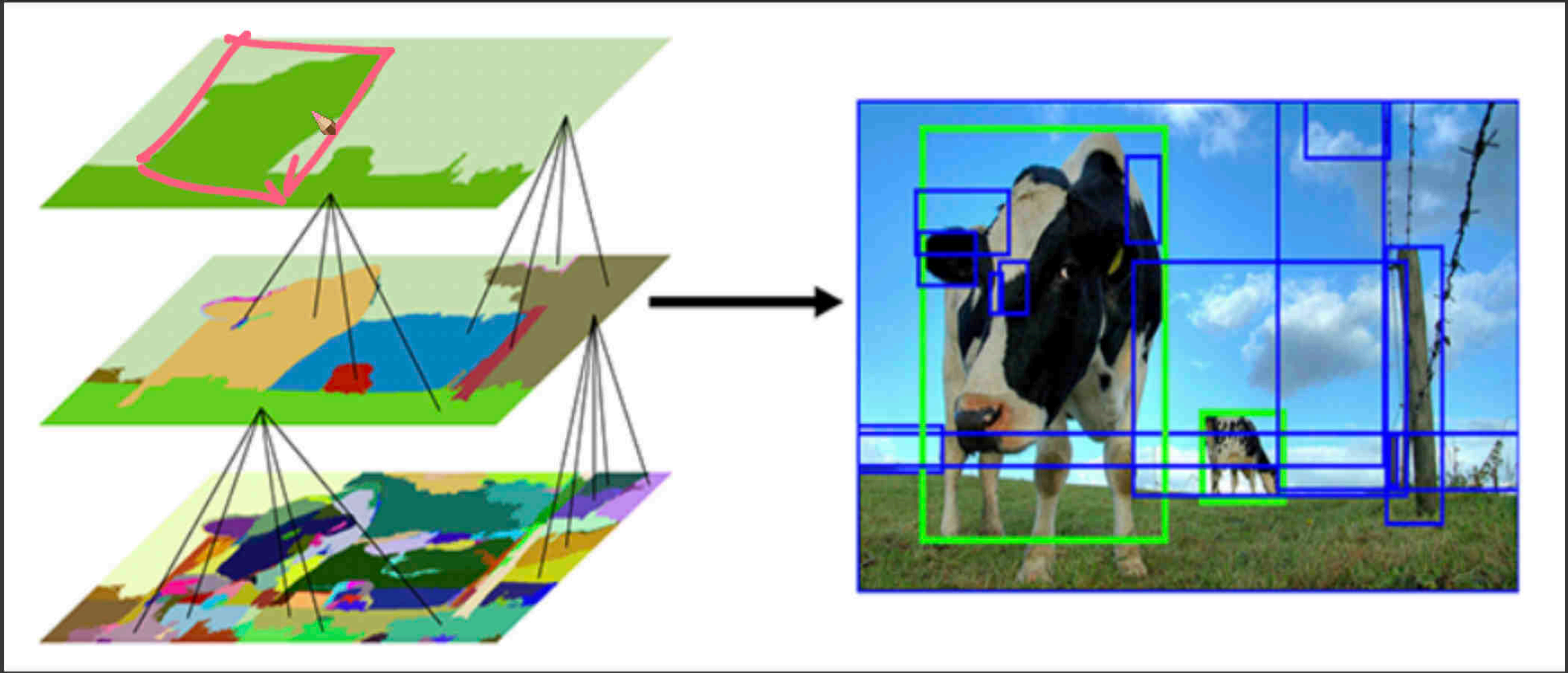



Input Image
Extract Region Proposals (~2k)
Wrapped Region
Compute CNN Features
Classification on Each ROI
Car ? not
Person ? not
Mobile ? not
CNN
Selective Search


Input Image
Extract Region Proposals (~2k)
Wrapped Region
Compute CNN Features
Classification on Each ROI
Car ? not
Person ? not
Mobile ? not
CNN
Selective Search



Is it possible to obtain end-to-end deep learning-based object detection?



Is it possible to obtain end-to-end deep learning-based object detection?

RCNN: Rich feature hierarchies for accurate object detection

https://arxiv.org/abs/1311.2524