

4-Jan-2025

lenet \rightarrow

Alexnet \rightarrow complex

Vggnet

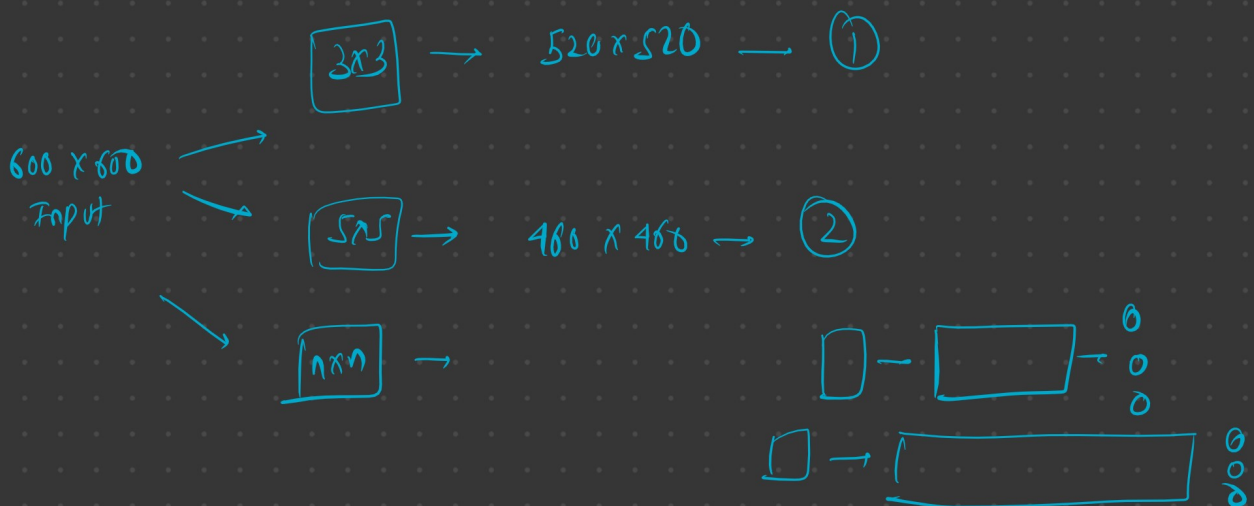
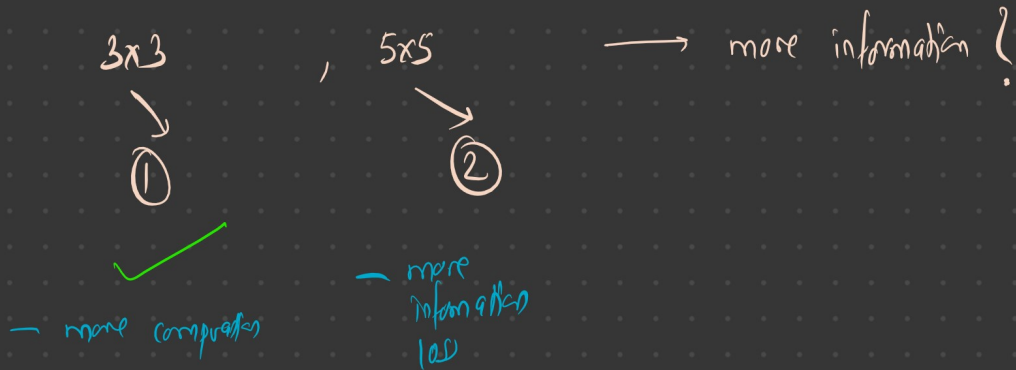
Resnet

Inception Block

— Relu \rightarrow To solve vanishing gradient problem

— less complex

— derivative simpler to compute



Vgg — 3x3 \rightarrow stack \rightarrow Deep Network

\rightarrow 64 128 256 512

1x1 Convolution \rightarrow

\rightarrow Dimensionality reduction | Expansion

\rightarrow Used to mix information across different channels \rightarrow Inception block



640x840



3x3 x 32

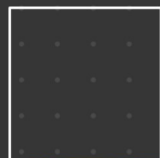


540x540 x 32



150x150 x 256

x 1

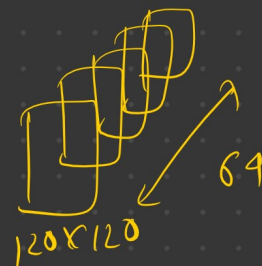


3x3 x 256, 64



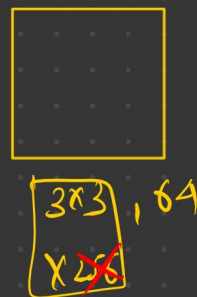
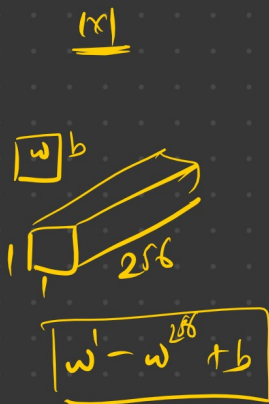
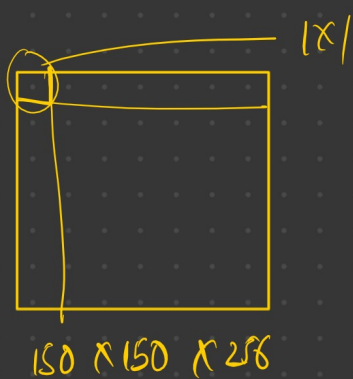
120x120 x 64

3x3 x 1, 64

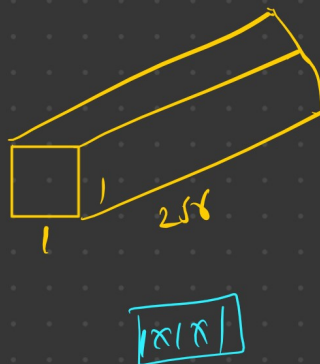
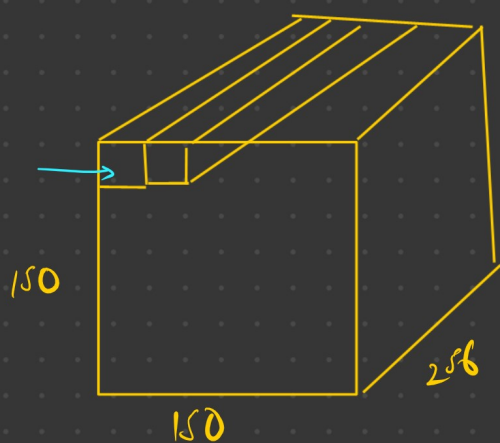


150x150x1
150x150x3

we need to
reduce the dimension for better computation.



150x150x1



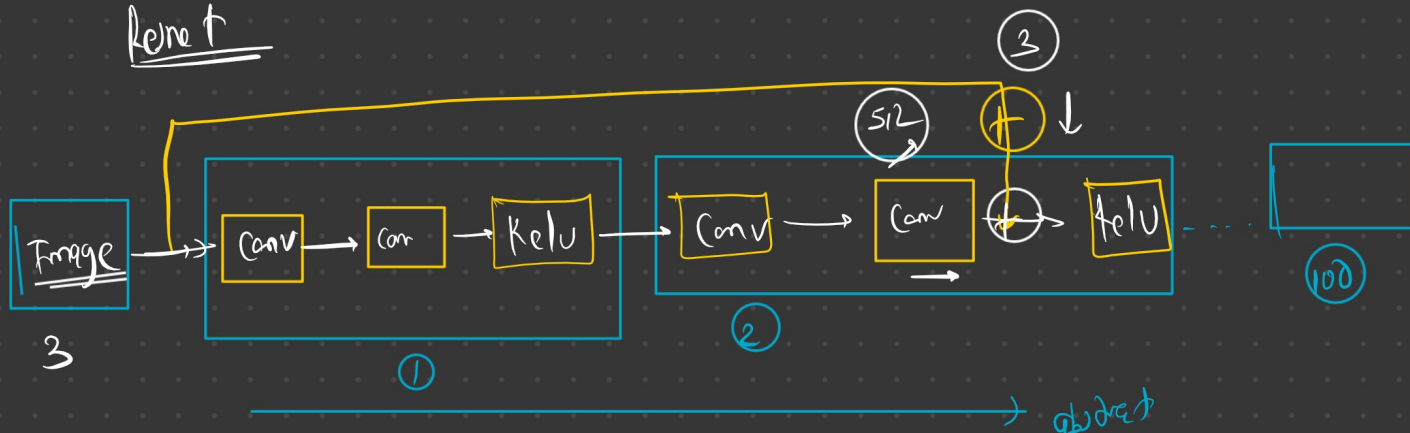
150x150x1

$$w_1x + w_2x_2 + w_3x_3 \rightarrow A$$

256 x 64

1x1x256 x 64
512

kernel



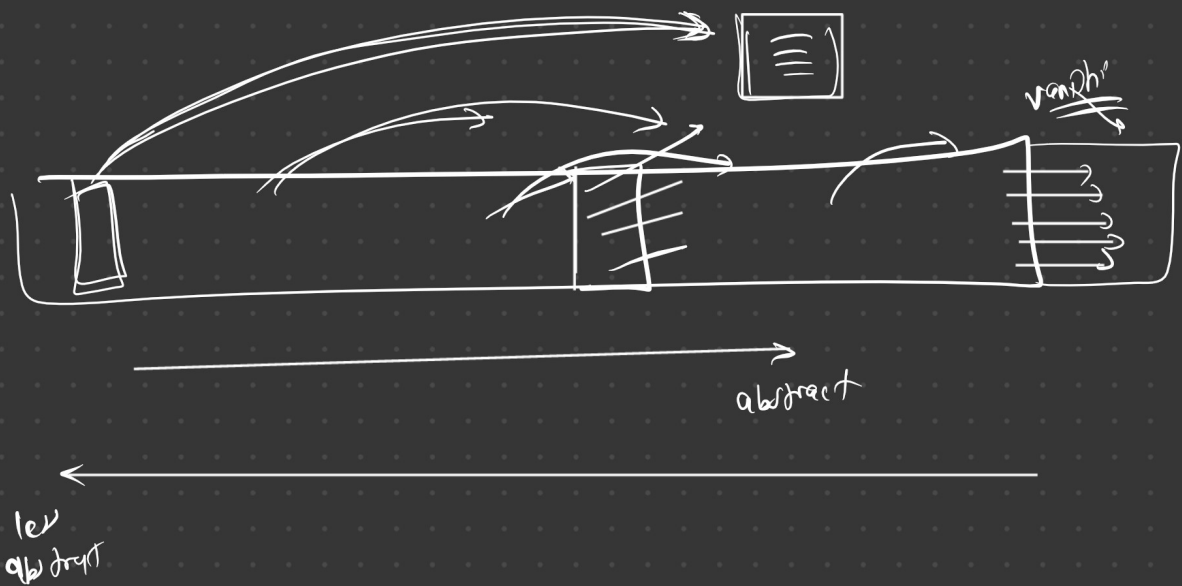
(rgb
chan)

3

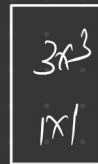
512 -> 1x1 -> downsampling

vanishing gradient

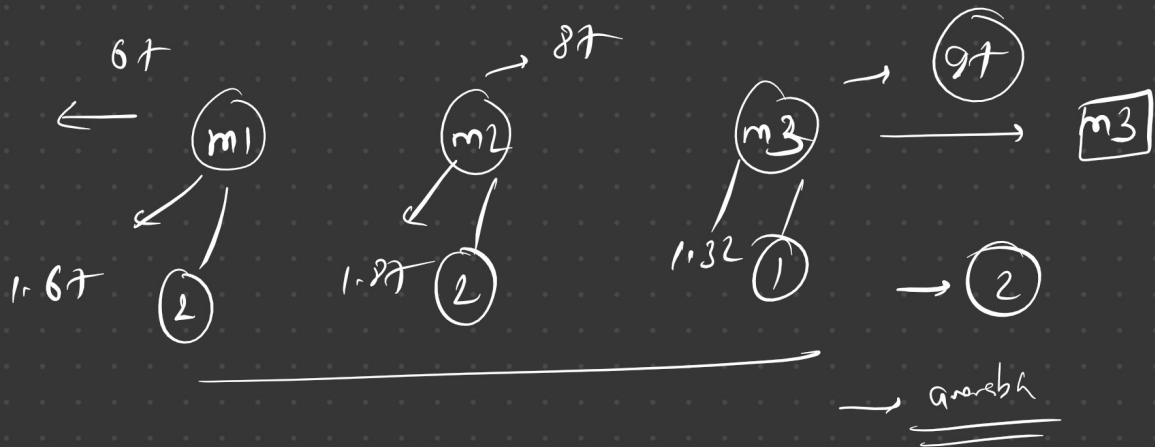




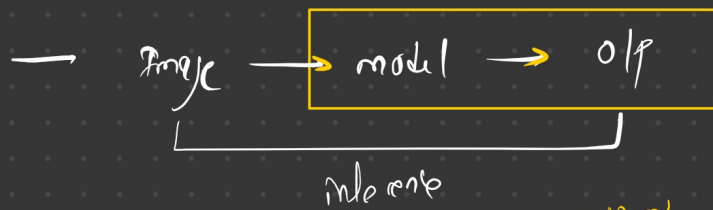
$|x|$ \rightarrow low of information



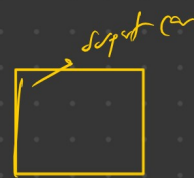
Inception Block



— pretrained model — ImageNet — 1000 classes



0.23 m



idx → class

post process

0.73 m

0.11 m

→ 1110 m

preprocess + inference + post process

①

②

③

X

> 70%

→ neg



dog

→ 0.6

→ 0.6

X → 0.2

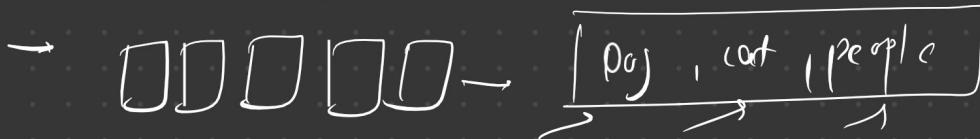
→ 0.3 X

→ 0.1

→ 0.2 X

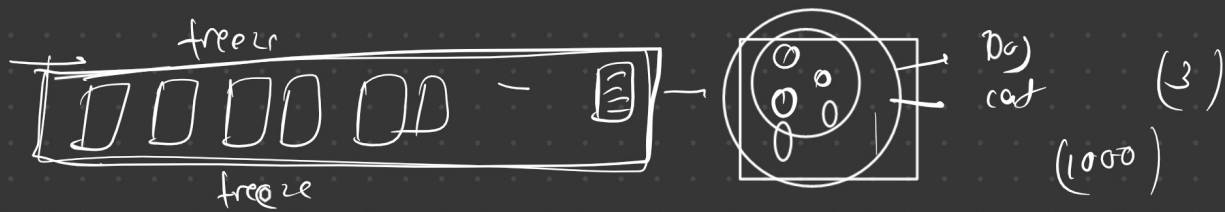
→ 0.10

→ renet → pytorch → pretrained = false



→ train → training from scratch,

→ renet — pretrained = true



train

→ conv layer = not train

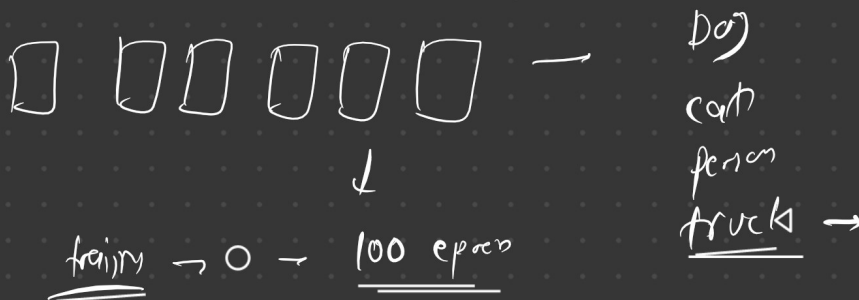
→ fc → train



fine tuning →



transfer learning → minimal adjustments to model weights,
2 layer kernel, fc



Job → 80 cats



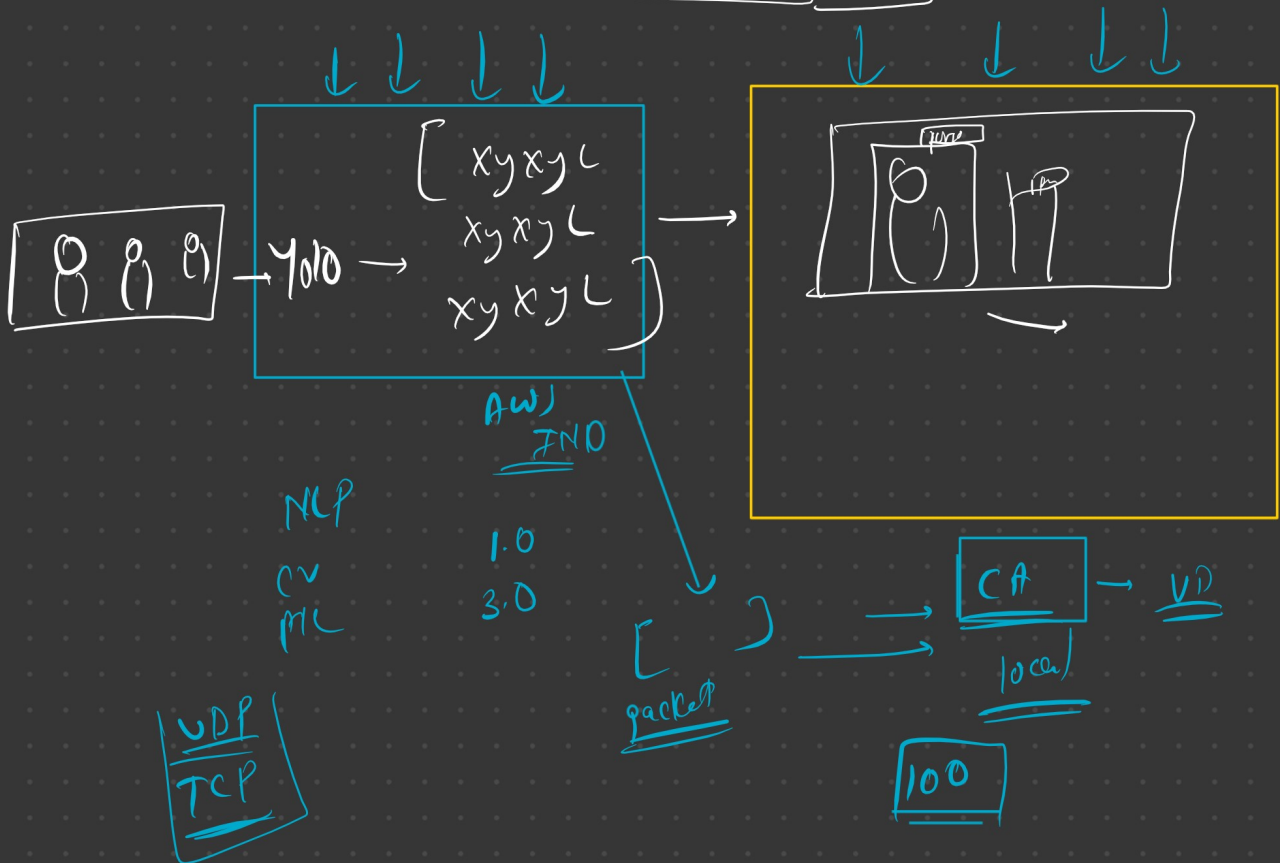


OpenAI

cam → 25 image / frames / sec

2 sec → trans

3 sec → 1 frame



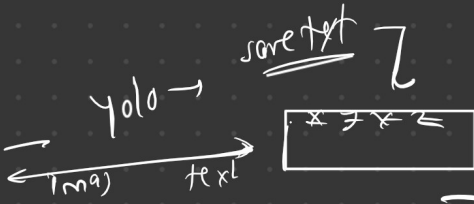


car

yolo → car
bike

person

Number plate



robotflow →

