

机器学习

第十三讲: 机器视觉

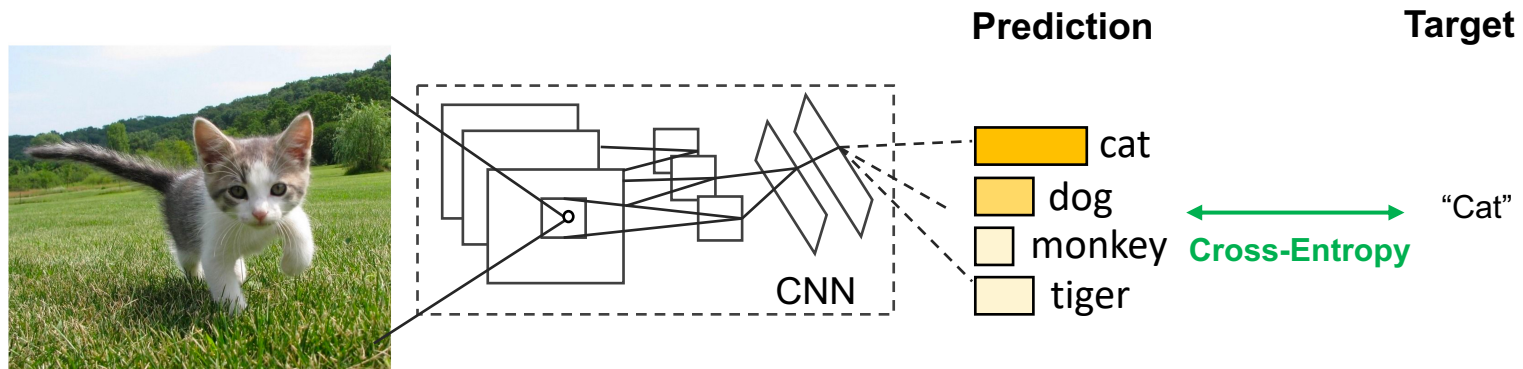
顾小东

上海交通大学计算机学院



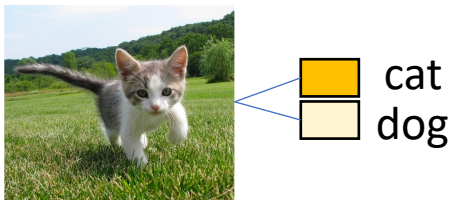
Recall: Image Classification

- Convolution + MLP + Softmax

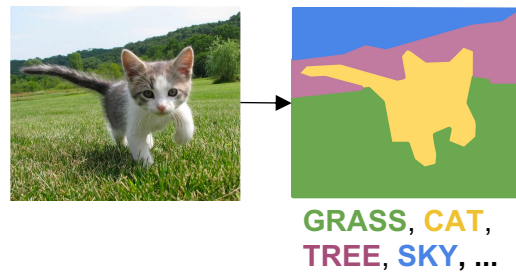


Computer Vision Tasks

Classification



Semantic Segmentation



Object Detection

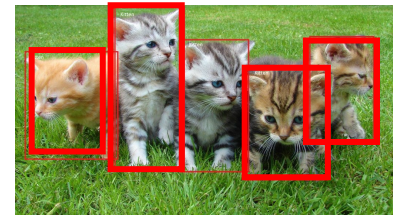
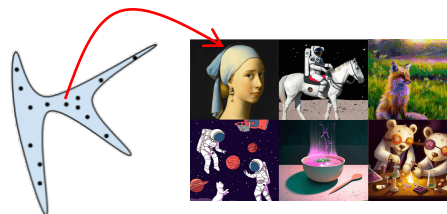


Image Captioning



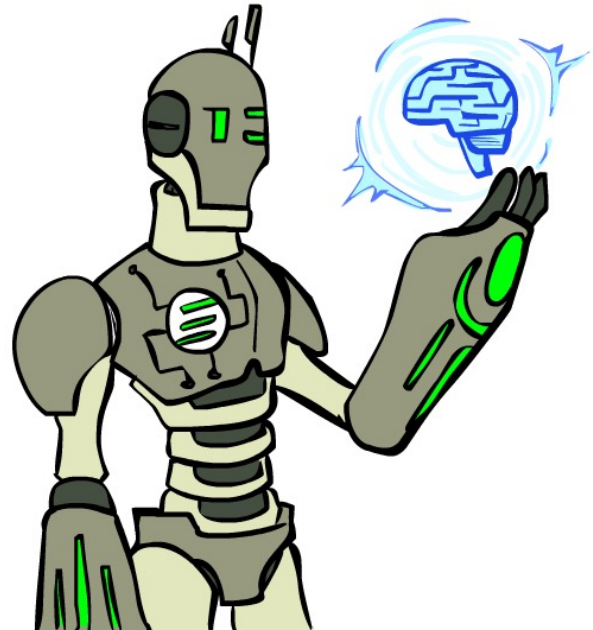
Image Generation



.....

Today

- Image Segmentation
- Object Detection
- Image Captioning



Semantic Segmentation

- Segment an image into disjoint regions that have independent semantics.

Label each pixel in the image with a category



输入: 图片



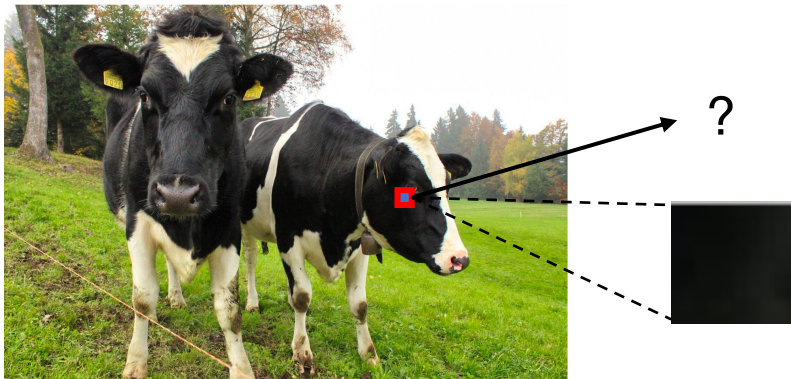
GRASS,
CAT,
TREE,
SKY, ...

输出: 每个像素所属的类别

A Naïve Idea

- Classifying Pixels?

Full image

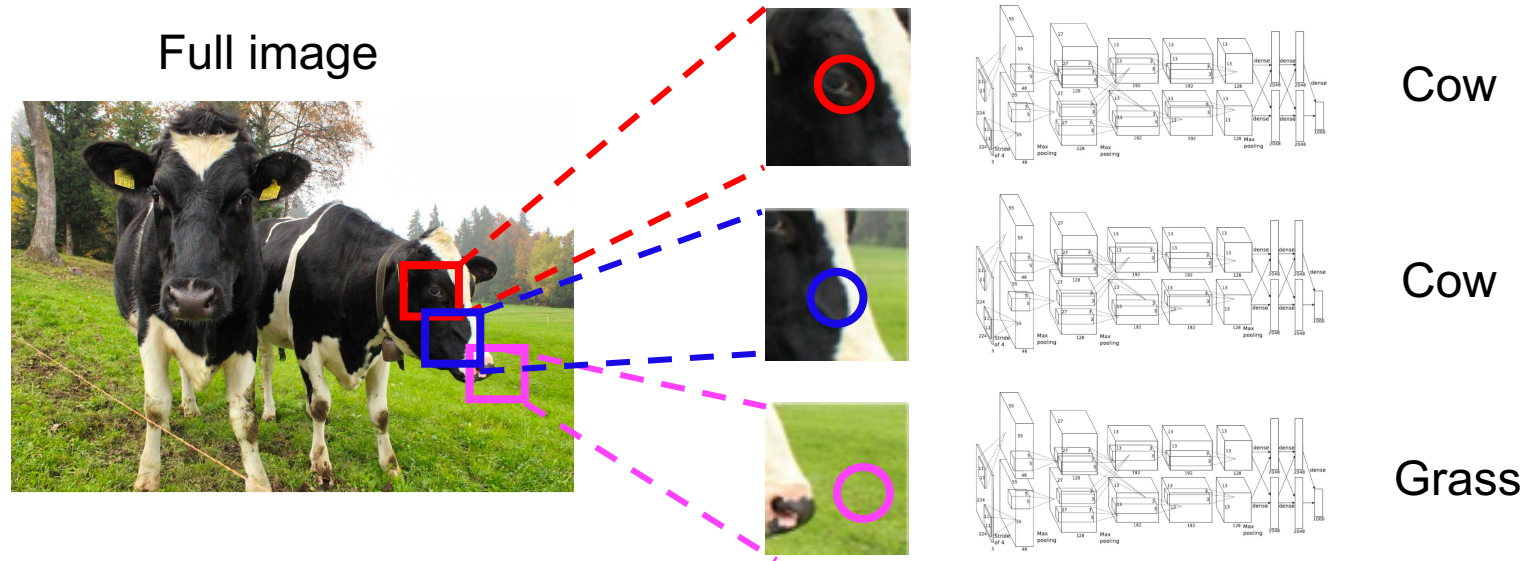


Impossible to classify without context

Q: how do we include context?

The Idea

- Sliding Window?

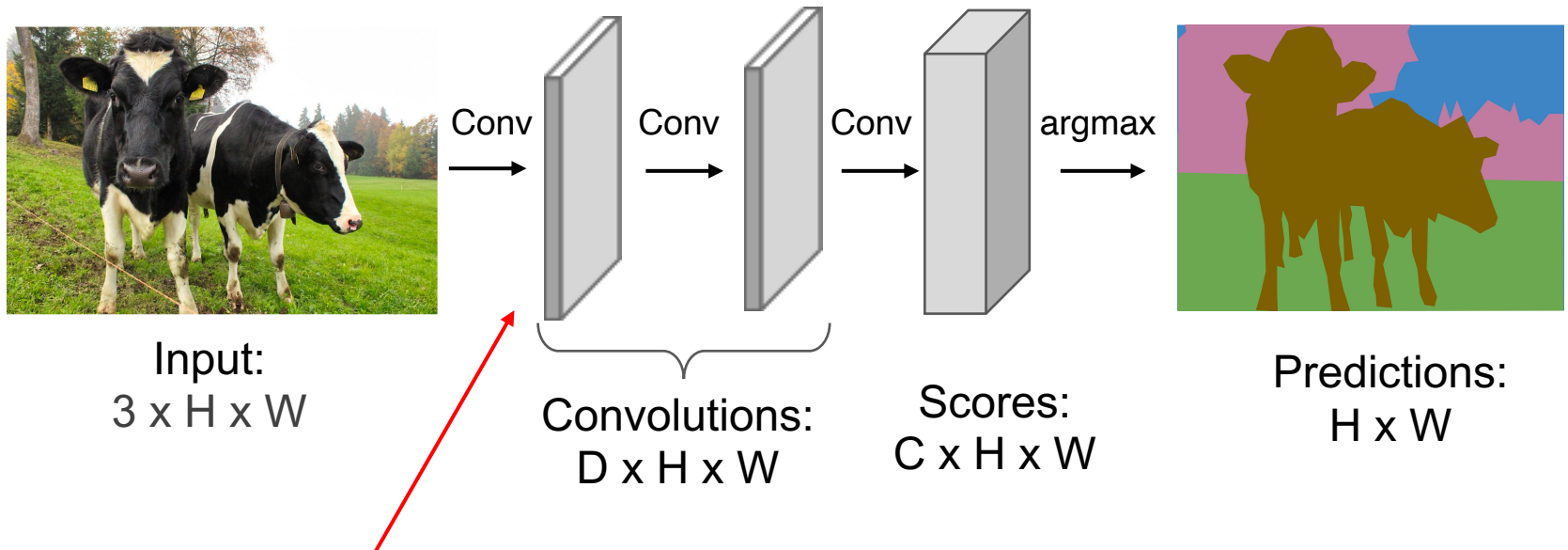


Problem: Very inefficient! Not reusing shared features between overlapping patches

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Fully Convolutional?

- Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!



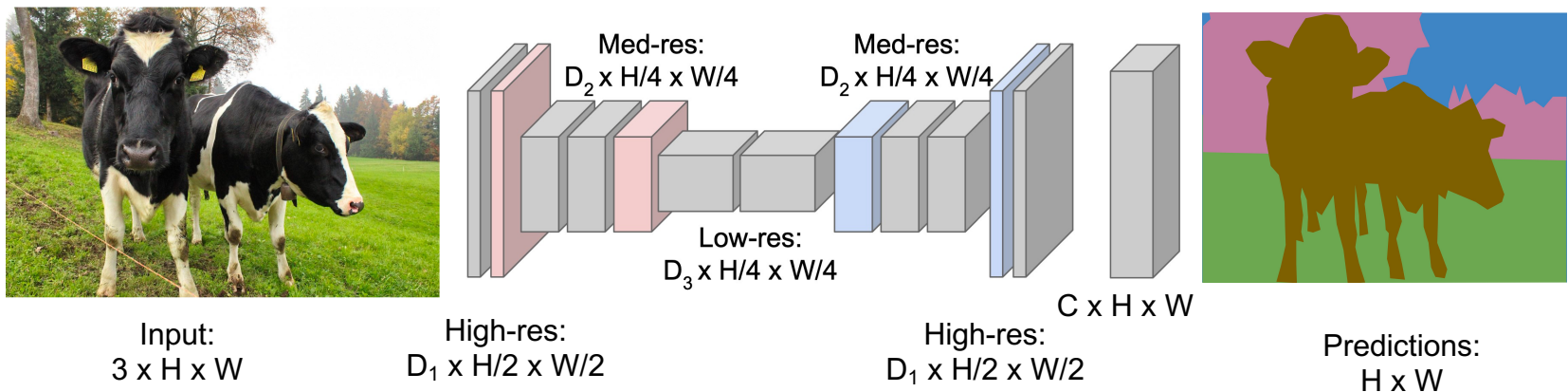
Problem: convolutions at original image resolution will be very expensive ...

Fully Convolutional?

Instead: Design networks as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

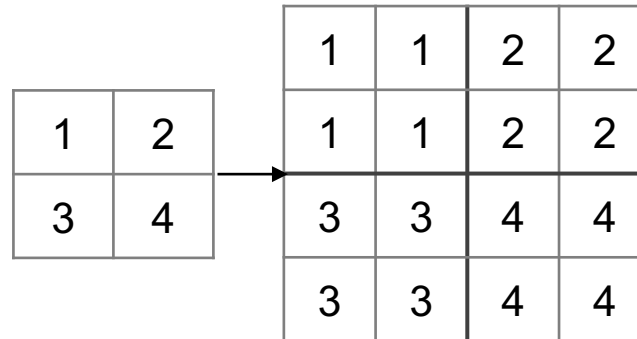
Downsampling:
Pooling, strided
convolution

Upsampling:
???

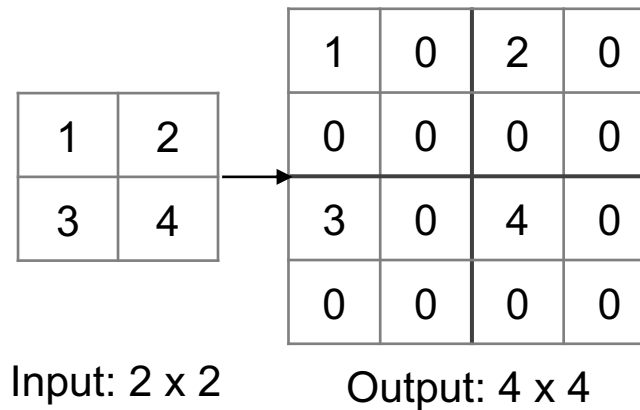


In-Network Upsampling: “Unpooling”

Nearest Neighbor



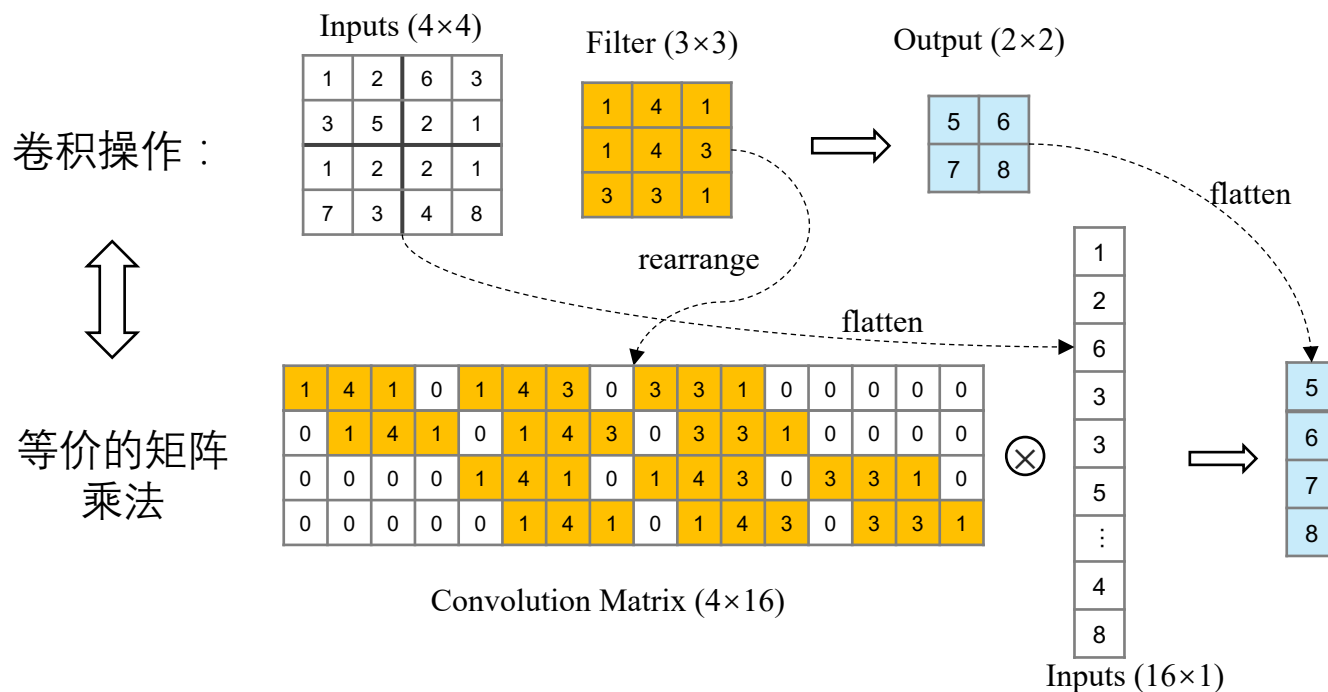
“Bed of Nails”



Upsampling with Transposed Convolution

- Convolution Matrix

We can reformulate convolution as **matrix multiplication** by rearranging the kernel matrix into a general matrix.



Upsampling with Transposed Convolution

- **Transposed Convolution Matrix**

Transpose the convolution matrix C (4×16) to C^T (16×4).

We can matrix-multiply C^T (16×4) with a column vector (4×1) to generate an output matrix (16×1).

Deconvolution:

Transposed Convolution
Matrix (4×16)

connects 1 value to 9
values in the output.

1	0	0	0
4	1	0	0
1	4	0	0
0	1	0	0
1	0	1	0
4	1	4	1
3	4	1	4
0	3	0	1
3	0	1	0
3	3	4	1
1	3	3	4
0	1	0	3
0	0	3	0
0	0	3	3
0	0	1	3
0	0	0	1

Inputs (2×2)

5
6
7
8

$\otimes$



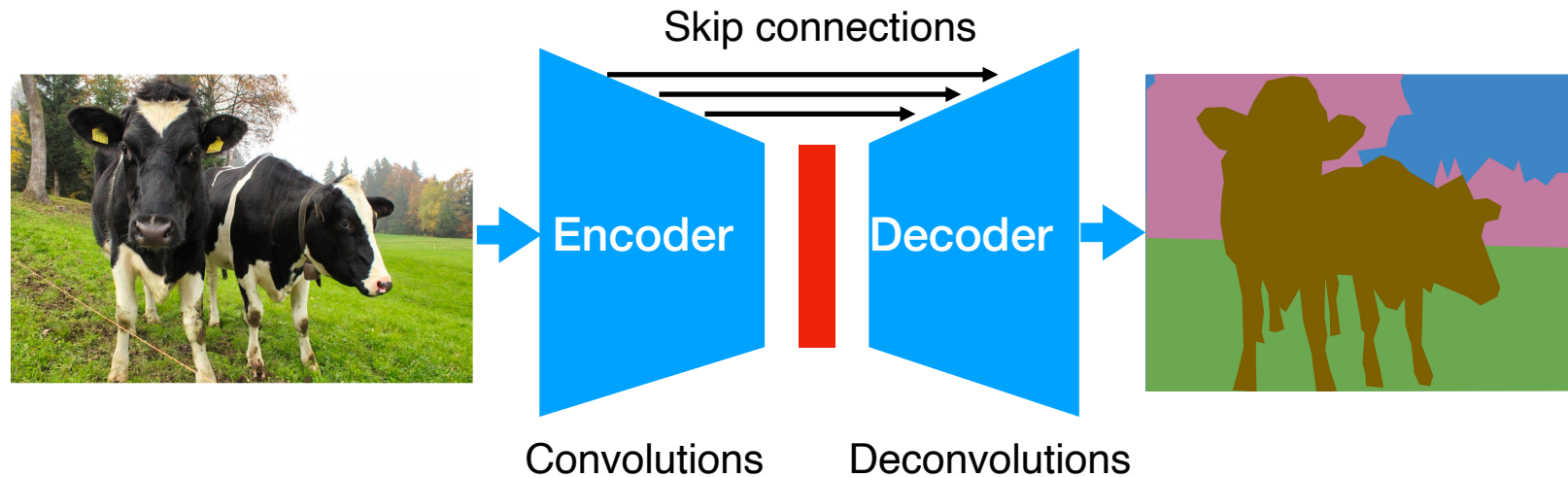
1
2
6
3
3
5
2
1
1
2
2
1
7
3
4
8

reshape

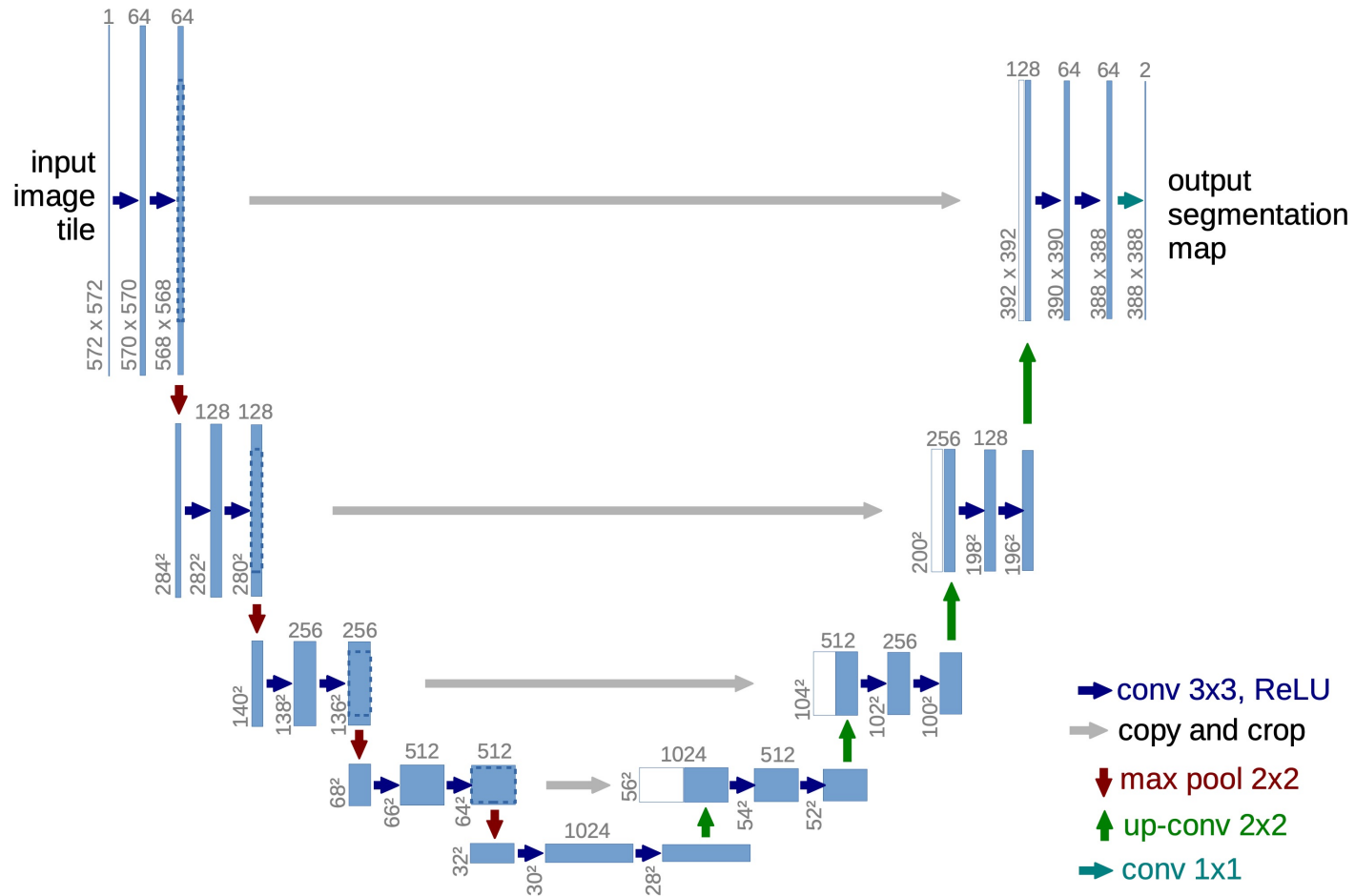
1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Encoder-Decoder Architectures

- U-Net



U-Net



Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation

TIME for Coding



- **Tutorial: Image segmentation with U-Net**

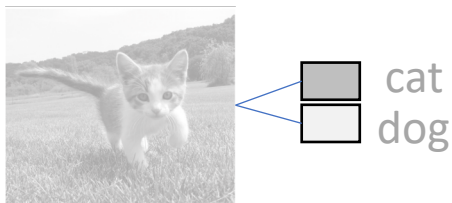
<https://www.kaggle.com/code/gokulkarthik/image-segmentation-with-unet-pytorch>

NOTE: How to enable GPU in your Kaggle kernel?

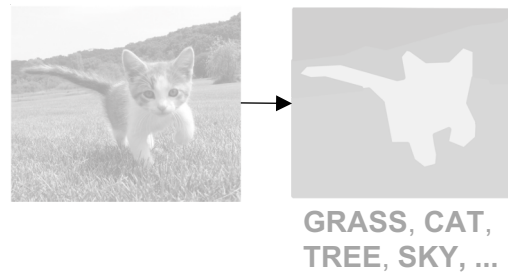
- Verify the phone number of your Kaggle account
- Click “accelerator” in the notebook setting (top right) and then select the GPU (<https://www.kaggle.com/general/97939>)

Computer Vision Tasks

Classification



Semantic Segmentation



Object Detection

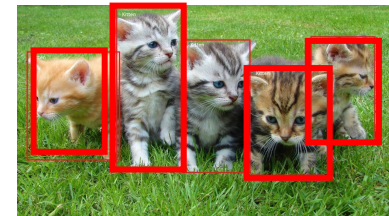
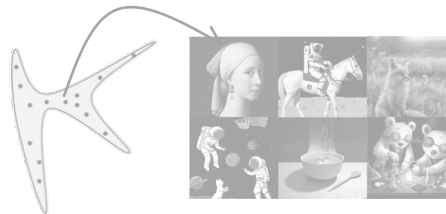


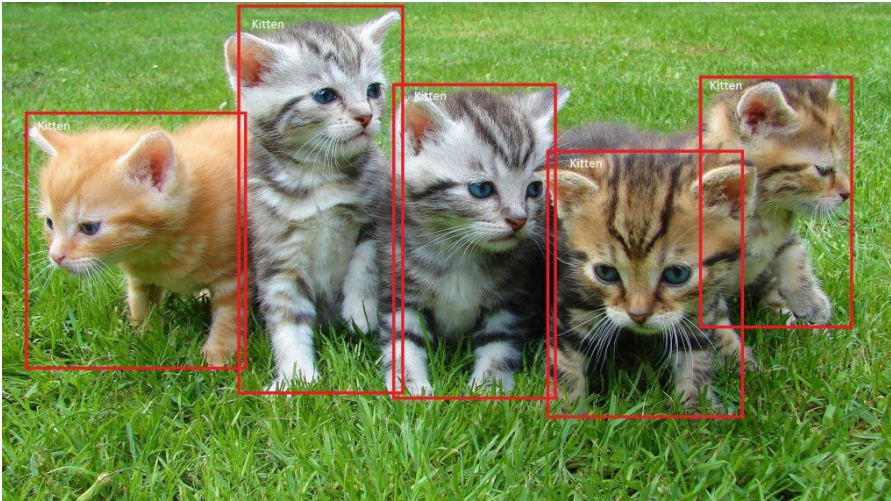
Image Captioning



Image Generation

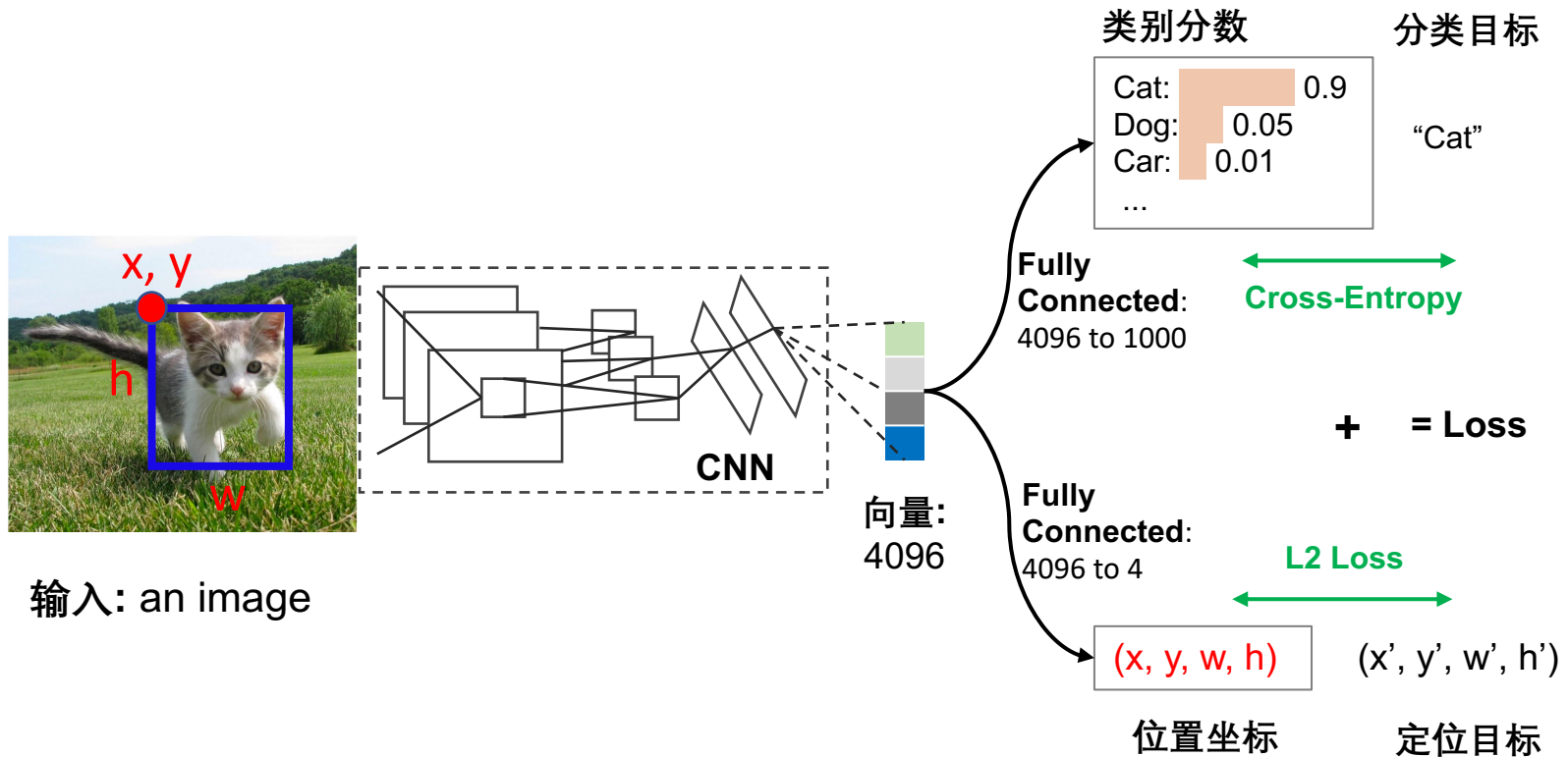


Object Detection

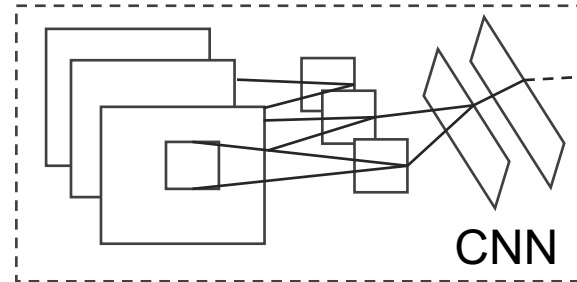
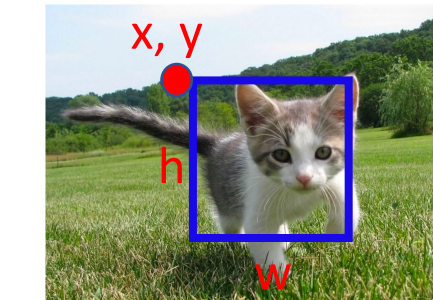


Single Object

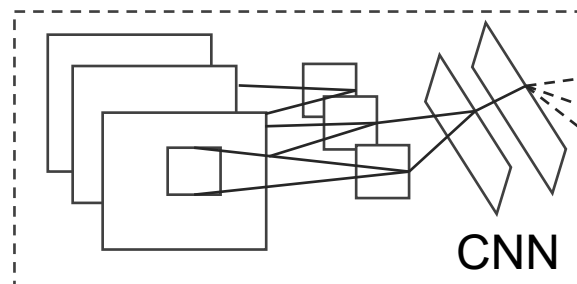
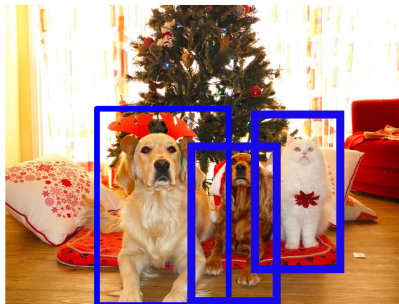
- Object detection = Classification + Localization



Multiple Objects?



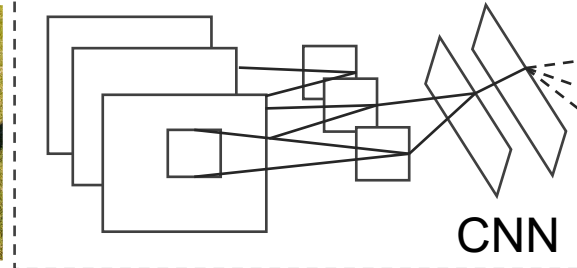
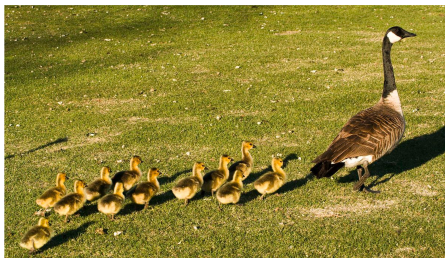
CAT: (x, y, w, h)



DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)



DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

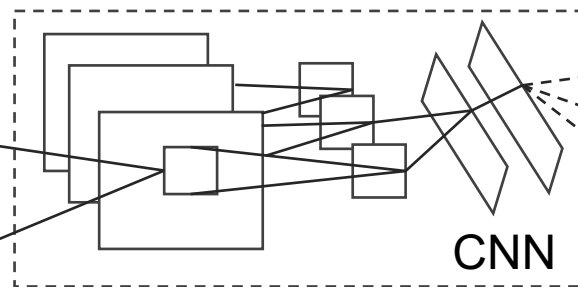
....

Too many predictions

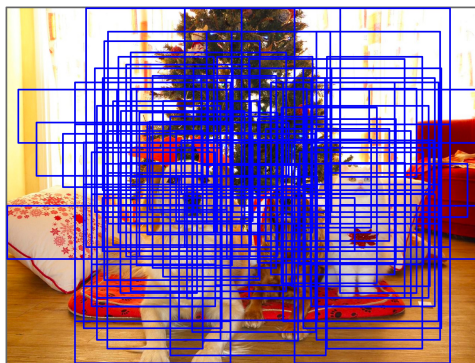
- **Problem:** each image (CNN) needs a different number of outputs!

Multiple Objects

- Idea: enumerate different crops of the image, apply a CNN to classify **each** crop as object or background.



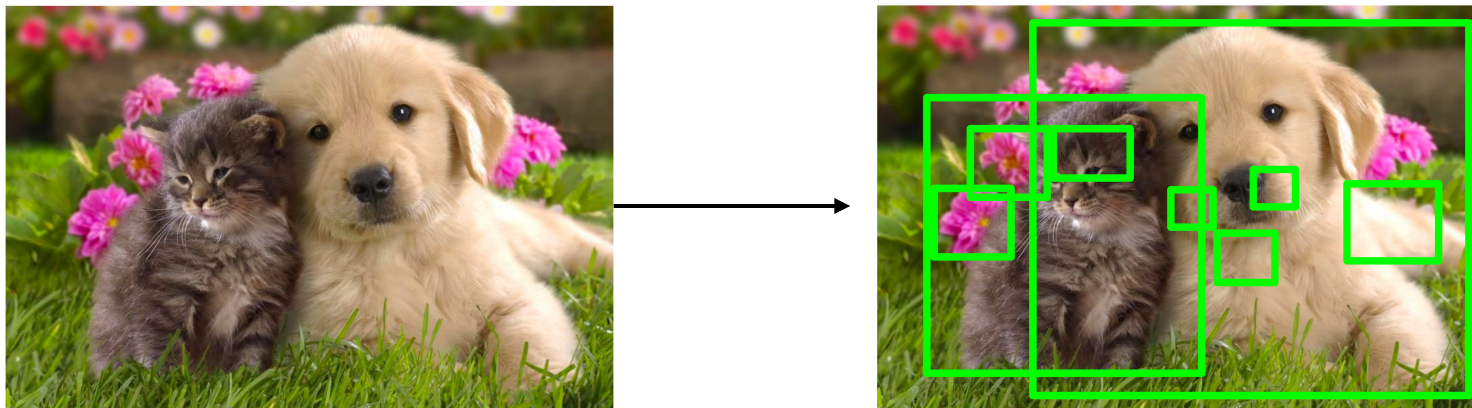
Dog? NO
Cat? NO
Background? YES



Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Region Proposals: Selective Search

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

R-CNN

④ 预测目标坐标 (dx, dy, dw, dh)

Bbox regr

SVM

③ 采用SVM对区域进行分类

Bbox regr

SVM

CNN

CNN

② 每个ROI图像输入ImageNet预训练的卷积神经网络

Bbox regr

SVM

CNN

ROI对应的图片块
(224x224像素)

① 采用提议算法获得初始ROI (约2k个)



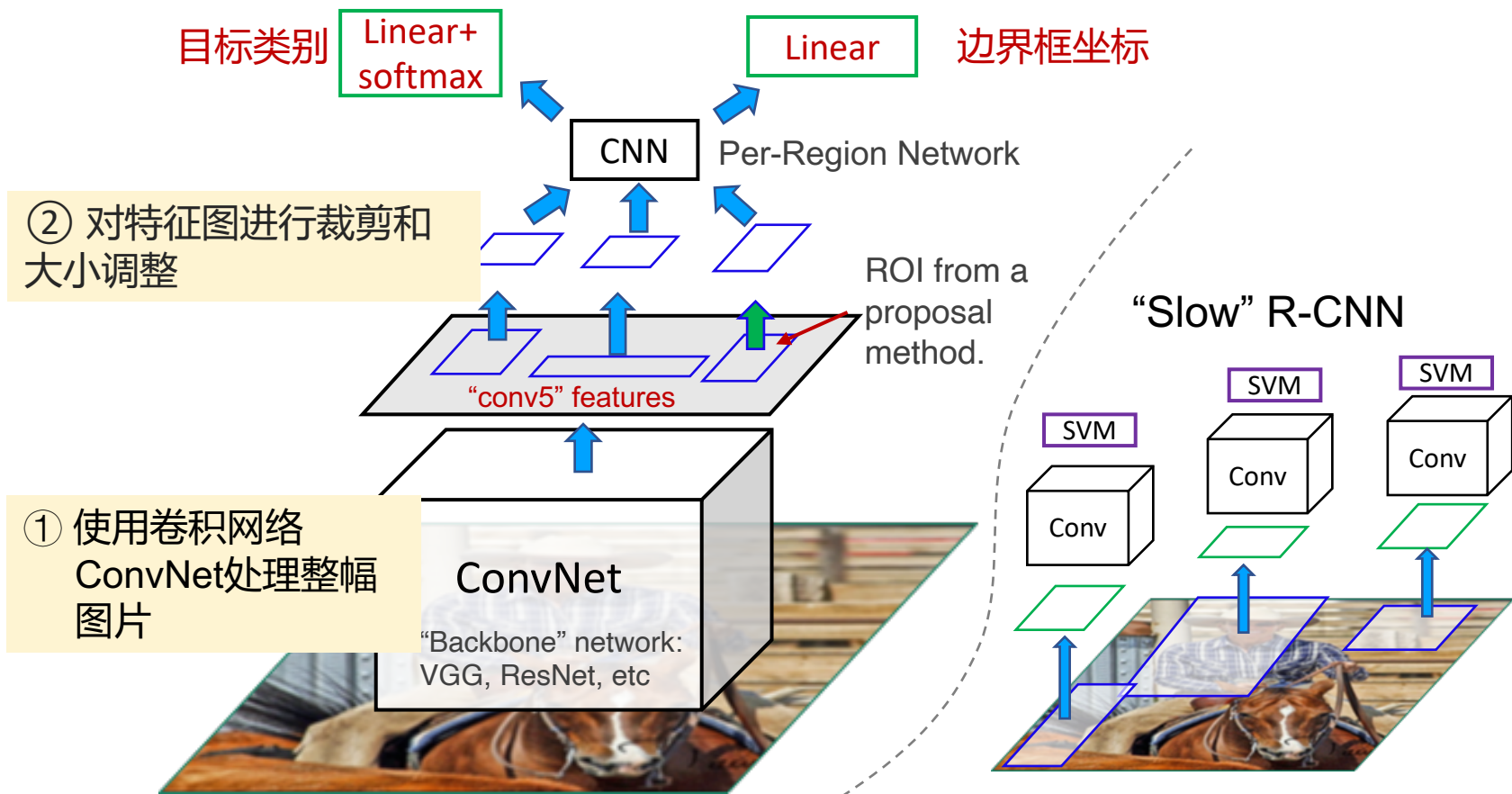
Problem: Very slow!

Need to do ~2k independent forward passes for each image!

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

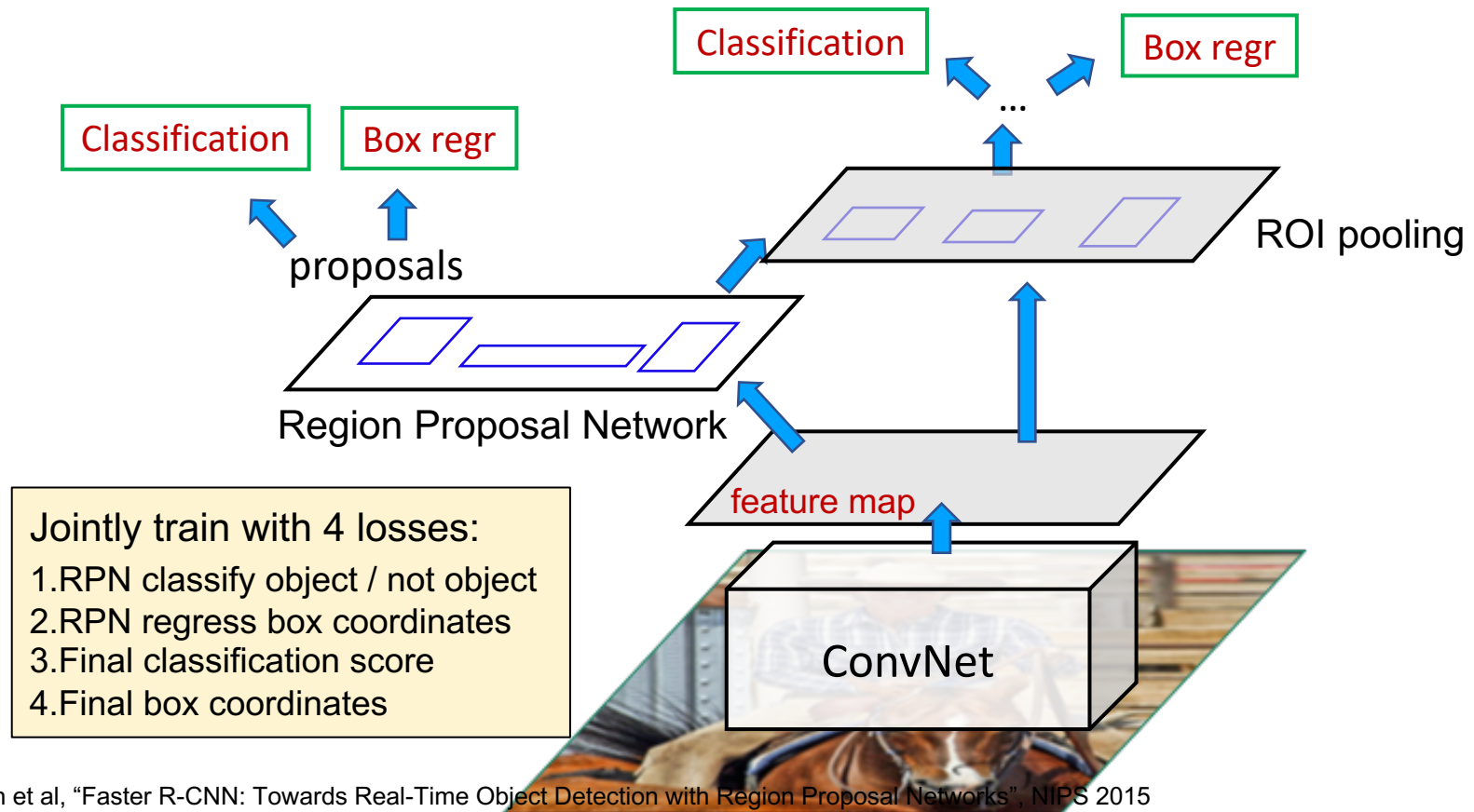
Fast R-CNN Crop the conv feature instead!

- Idea: Pass the image through convnet before cropping!



Faster R-CNN *Make CNN do proposals !*

- Insert a Region Proposal Network (RPN) to predict proposals from features



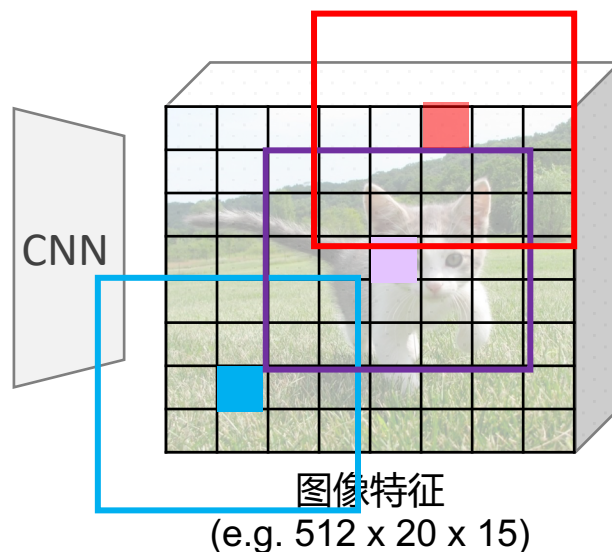
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Region Proposal Network

- Imagine an **anchor box (锚盒)** of fixed size at each point in the feature map



输入图片
(e.g. 3 x 640 x 480)



在每个点预测有没有包含目标的锚盒(二值分类)

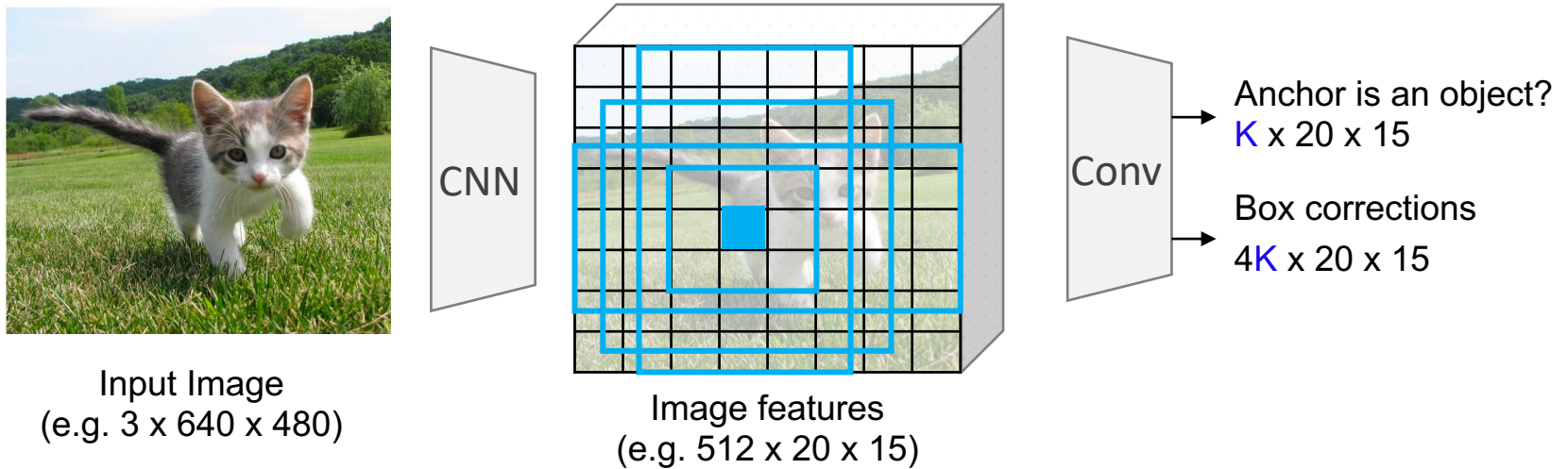
锚点是否存在目标?
1 x 20 x 15

目标具体位置
4 x 20 x 15

对于预测存在目标的锚盒, 进一步预测对应的框位置(回归预测四个坐标数字)

Region Proposal Network

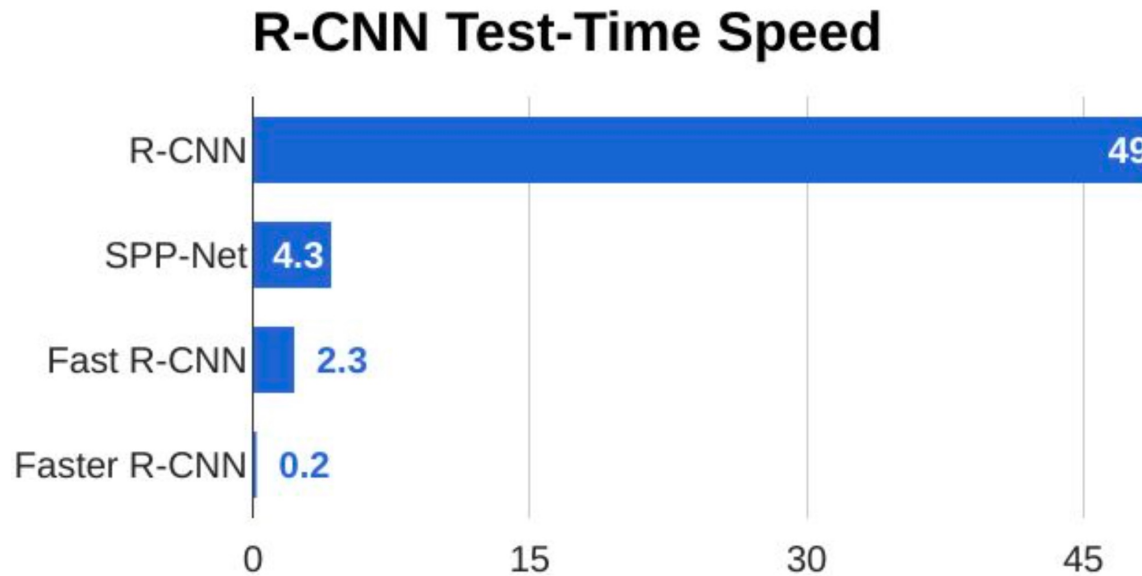
- In practice use K different anchor boxes of different size / scale at each point



- Sort the $K \times 20 \times 15$ boxes by their “objectness” score, take top ~300 as our proposals

Faster R-CNN

- How fast?



YOLO – Real Time Image Segmentation

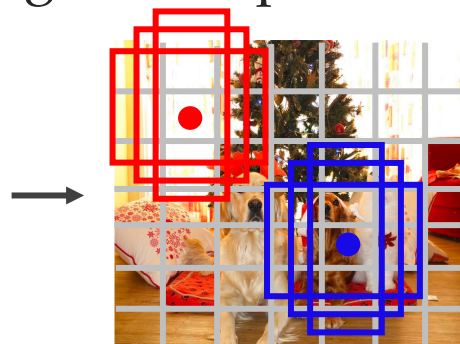
- Formulate object detection as a regression problem



输入图片
 $3 \times H \times W$

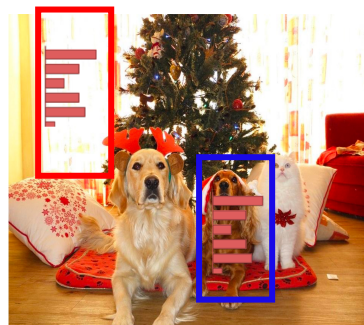
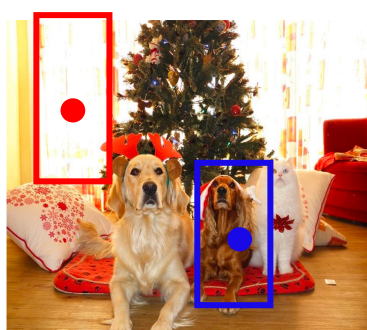


将图片切成 $S \times S$ 个
网格 (这里 $S=7$)



每个网格单元预测 B 个带置信
度的边界框: $(x, y, h, w, \text{confid})$

根据 B 个边界框
回归一个最终
的预测框

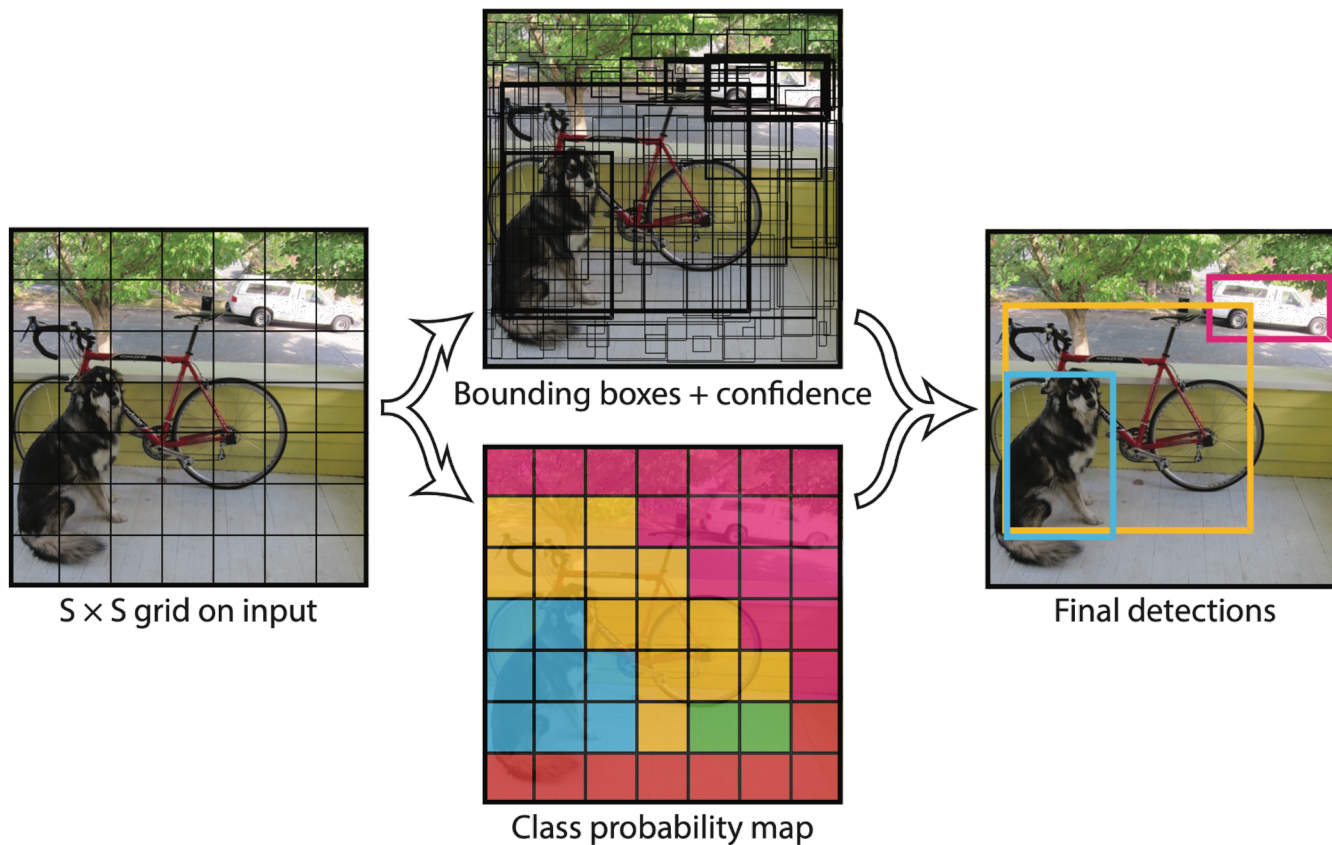


每个单元同
时预测属于 C
个目标类(含
背景)的概率
分布

输出向量:
 $7 \times 7 \times (5 * B + C)$

Redmon et al. You Only Look Once: Unified, Real-Time Object Detection. CVPR 2016

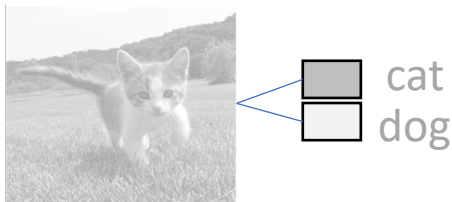
YOLO – Real Time Image Segmentation



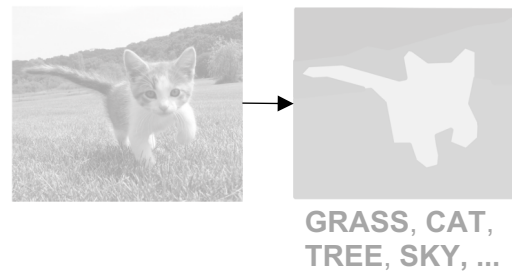
Redmon et al. You Only Look Once: Unified, Real-Time Object Detection. CVPR 2016

Computer Vision Tasks

Classification



Semantic Segmentation



Object Detection

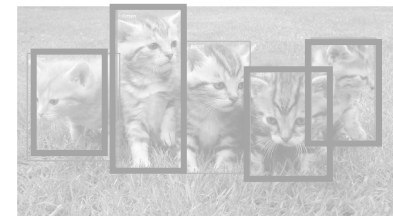


Image Captioning



Image Generation

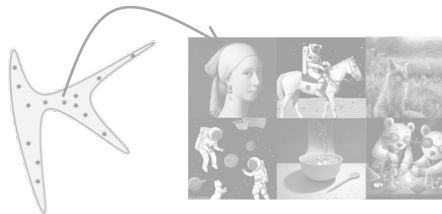


Image Captioning

- Generate a natural language description for an image.



输入: 图像 I

“一个人戴着一顶帽子正在河边钓鱼”

输出: 自然语言序列 $y = y_1, y_2, \dots, y_T$

Idea: Encoder-Decoder?

Encoder: $c = f_w(\mathbf{z})$

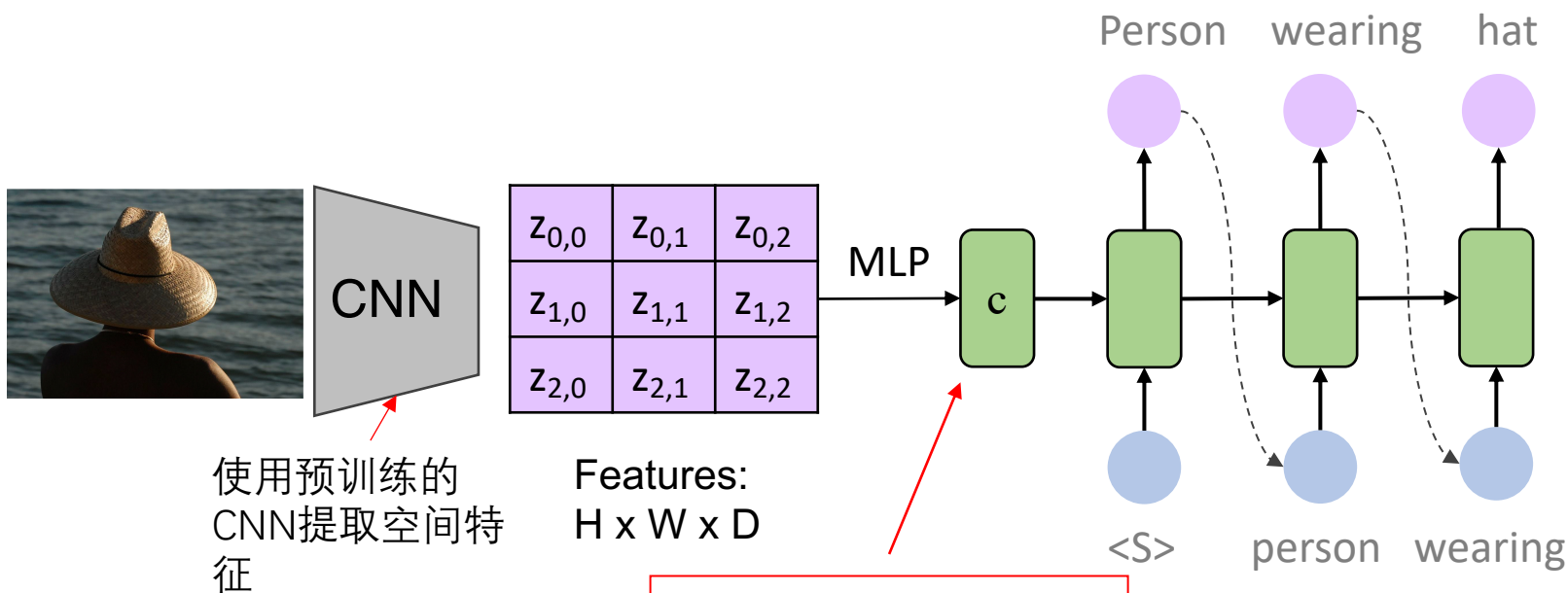
where $\mathbf{z}$ is spatial CNN features

$f_w(\cdot)$ is an MLP

Decoder: $y_t = g_v(y_{t-1}, h_{t-1}, c)$

where context vector c is

often $h_0 = c$



**Problem: Input is
"bottlenecked" through c**

Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

CNN with Self-Attention

- We can also apply **self-attention** to CNN features.

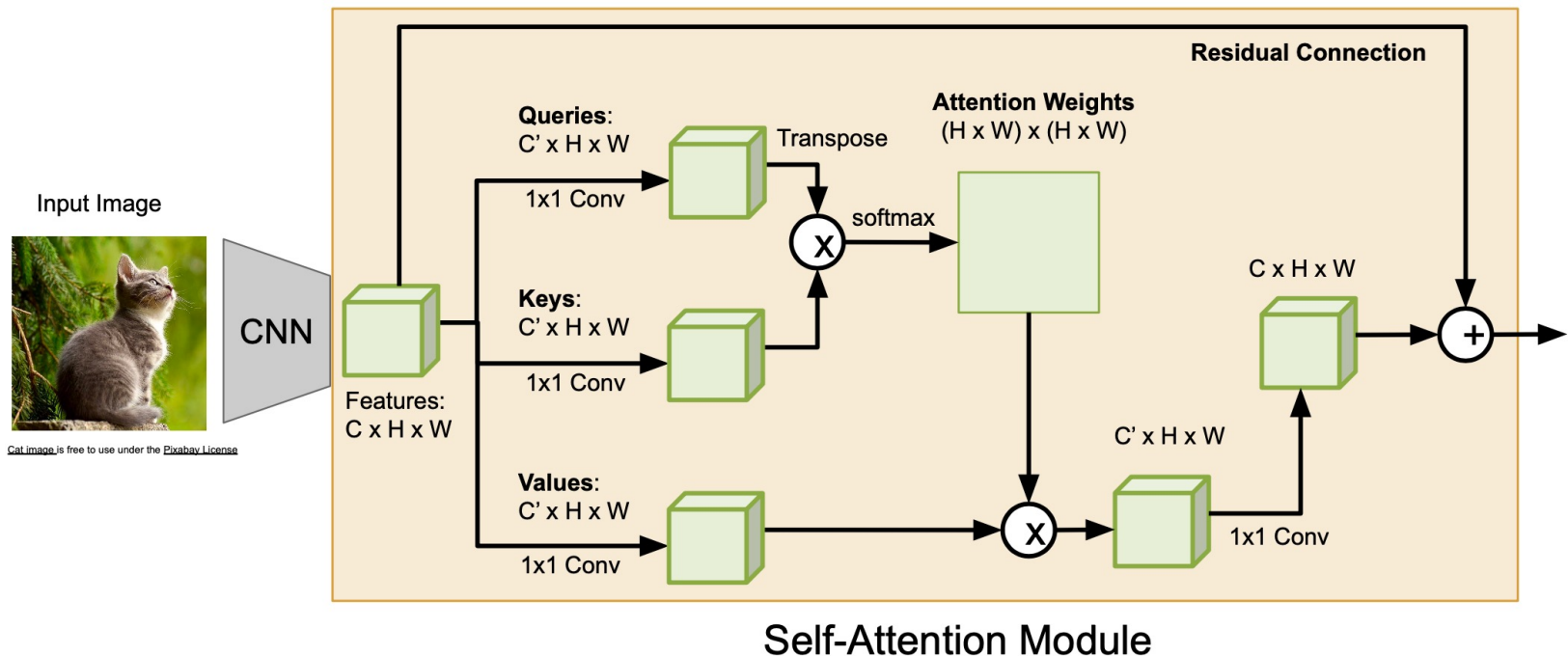
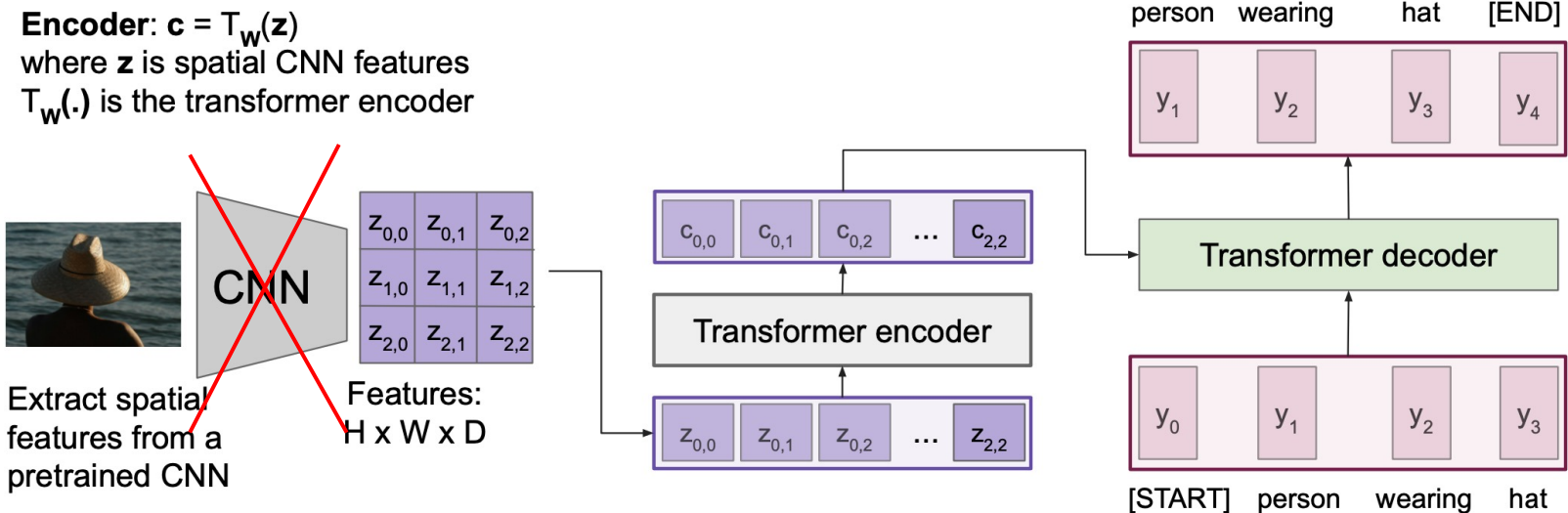
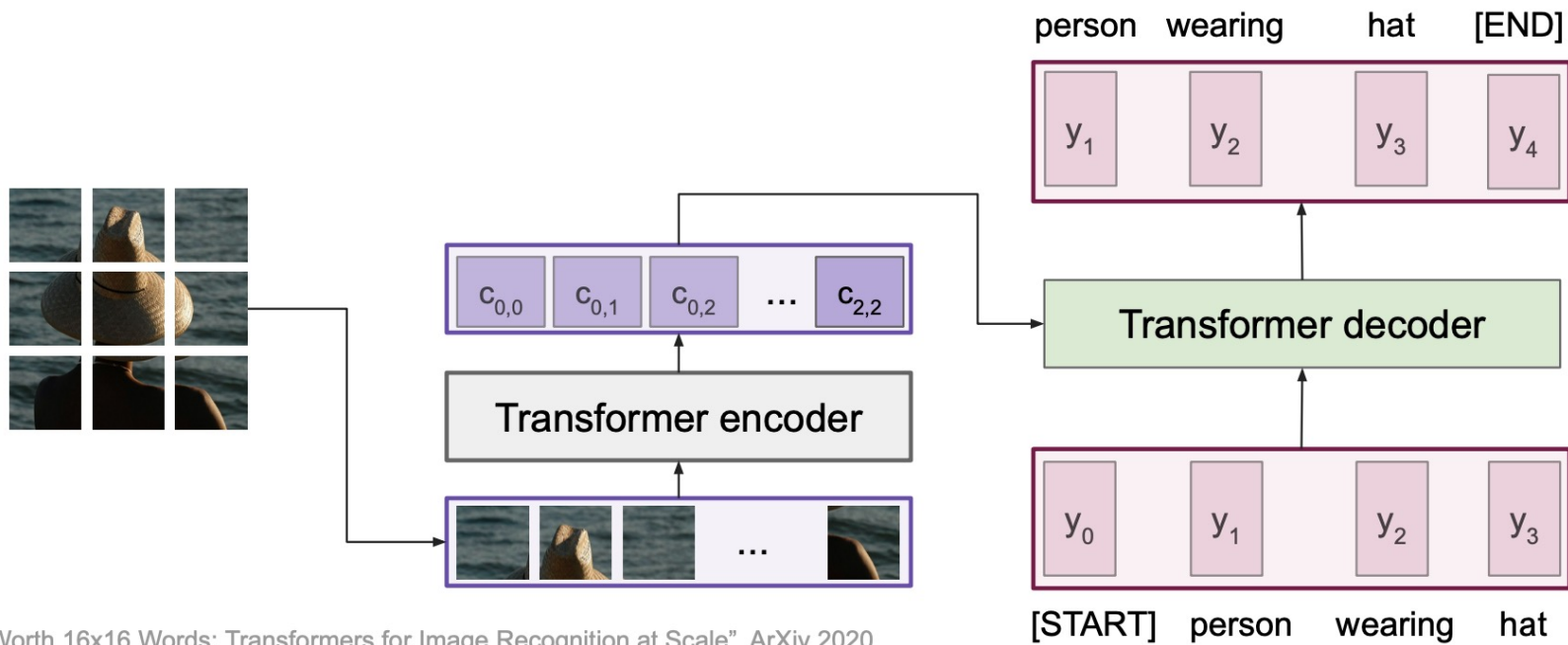


Image Captioning ~~with~~ using Transformers



Perhaps we don't need convolutions at all?

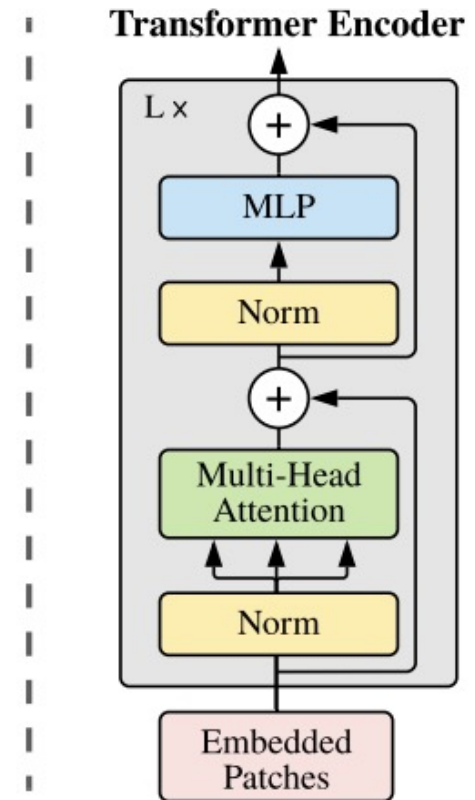
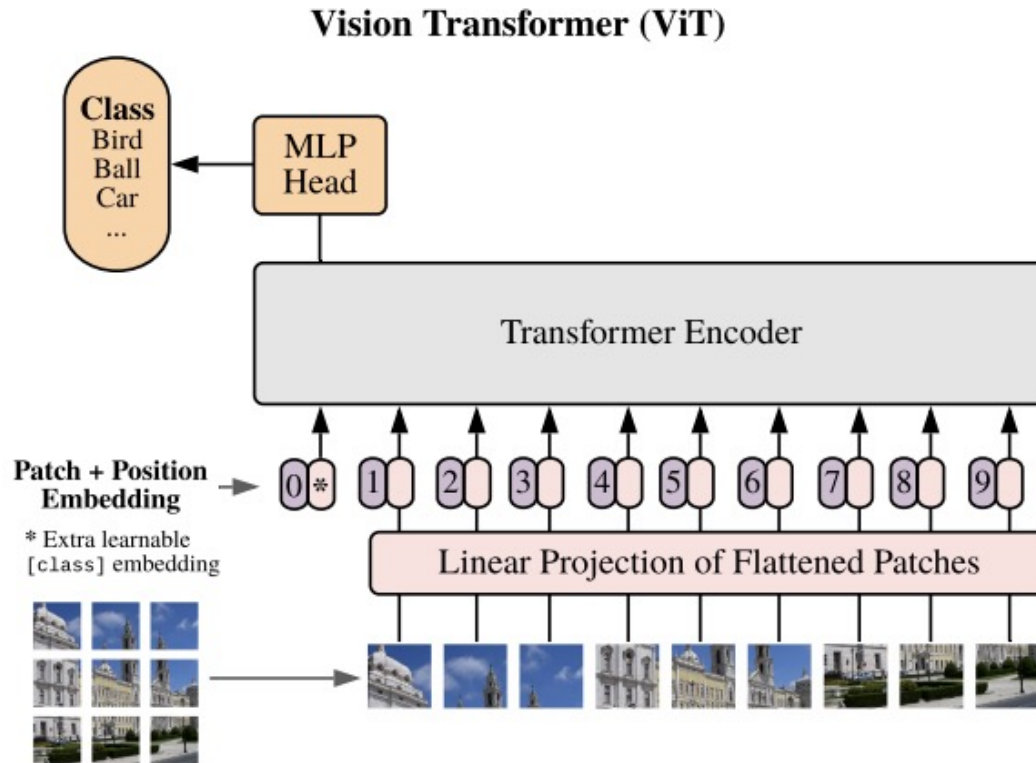
Image Captioning using **ONLY** Transformers



Worth 16x16 Words: Transformers for Image Recognition at Scale". ArXiv 2020

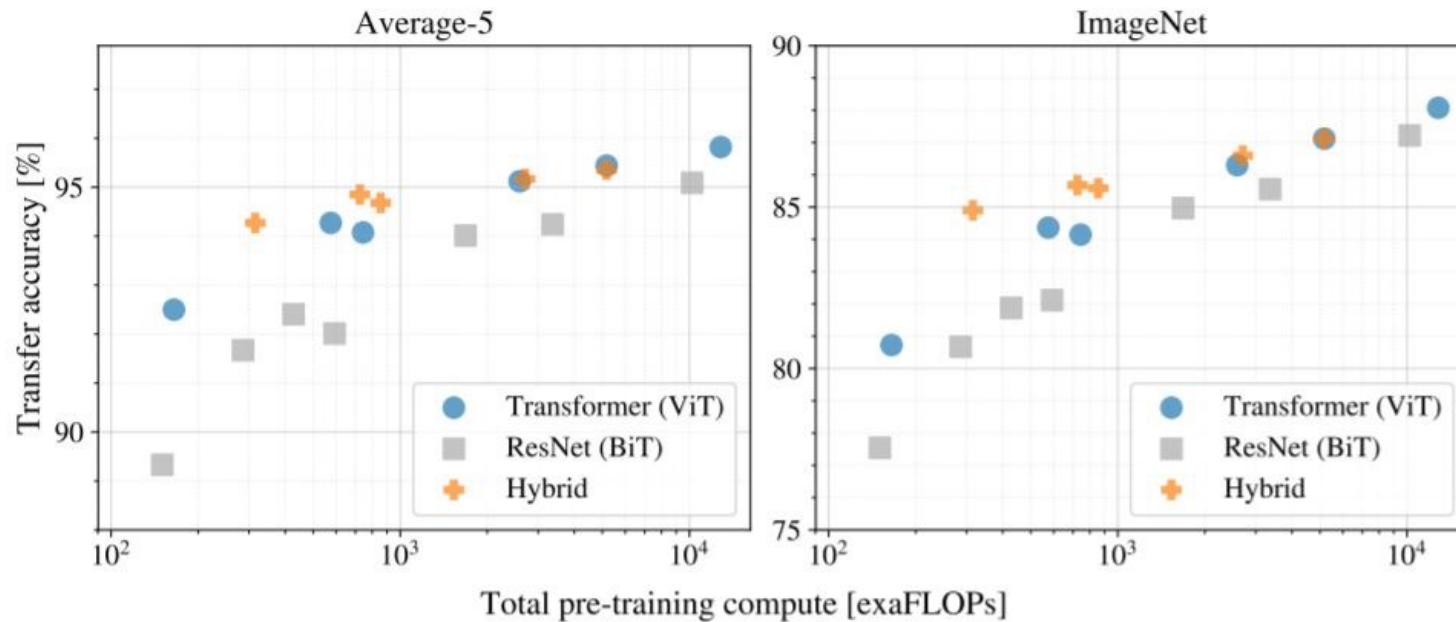
Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ArXiv 2020

Vision Transformer (ViT)



Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ArXiv 2020

Vision Transformers vs. ResNets

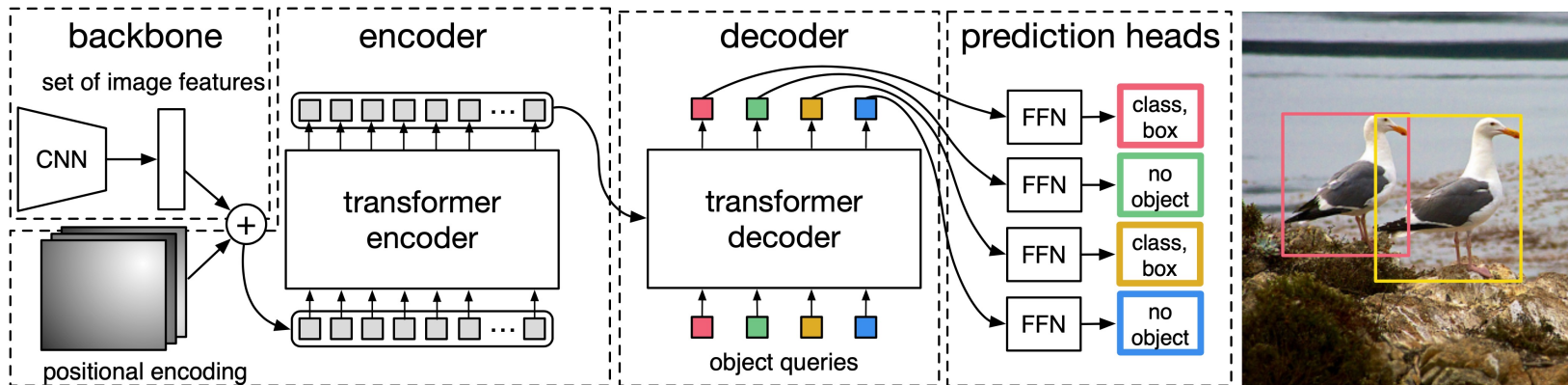
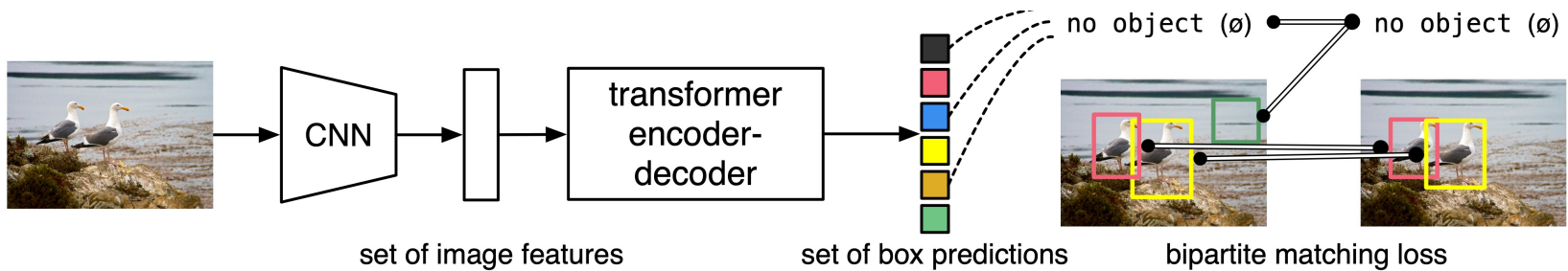


Farewell CNN ?



Vision Transformers for Object Detection

- DETR



视觉大模型与多模态对齐



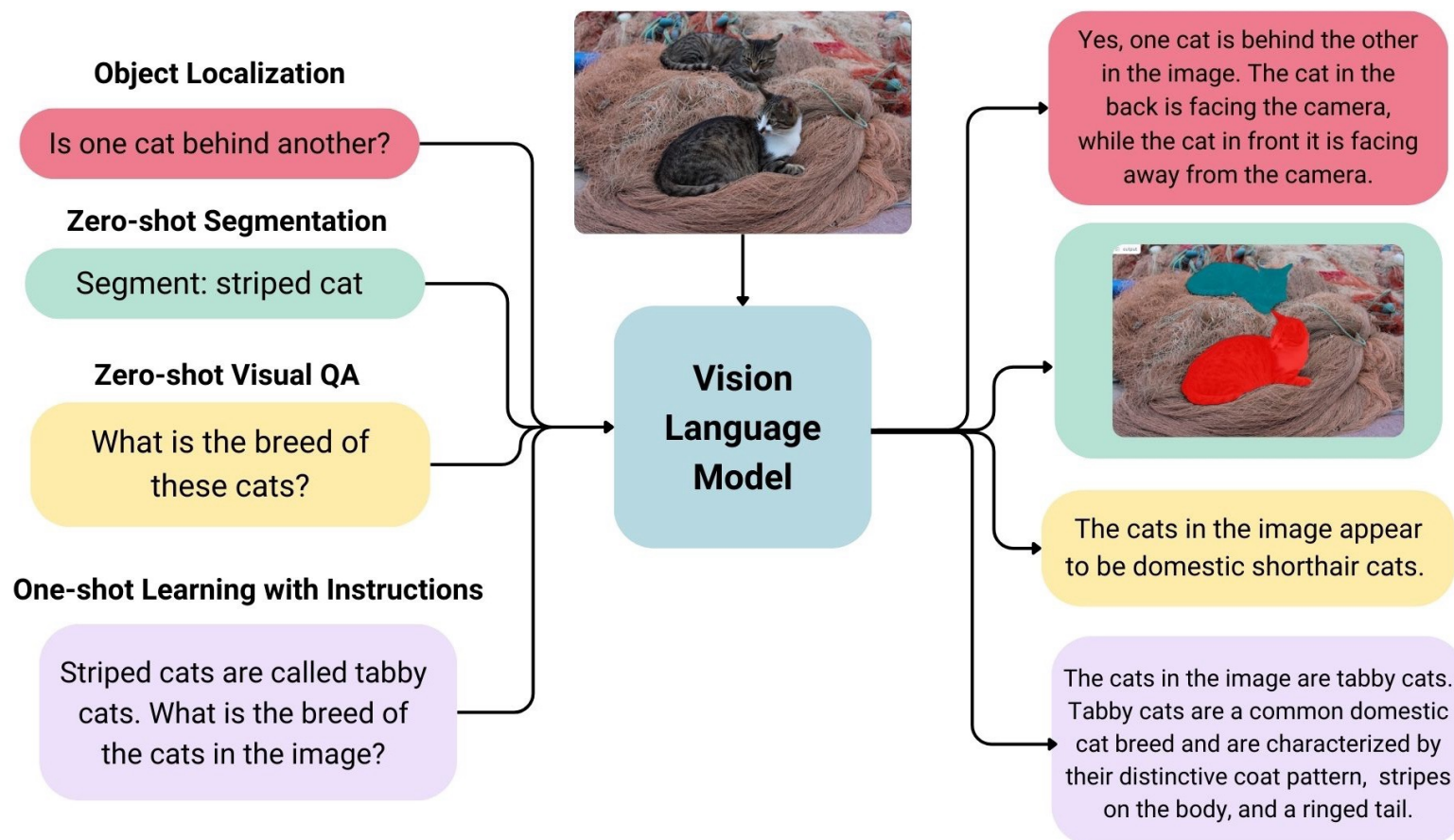
["Lion and cheetah",
"Cat and dog",
"Rabbit and bird"]

Vision-Language
Model

[{'score': 0.9950, 'label': 'cat and dog'},
{ 'score': 0.0048, 'label': 'rabbit and lion'},
{ 'score': 8.81e-05, 'label': 'lion and cheetah'}]

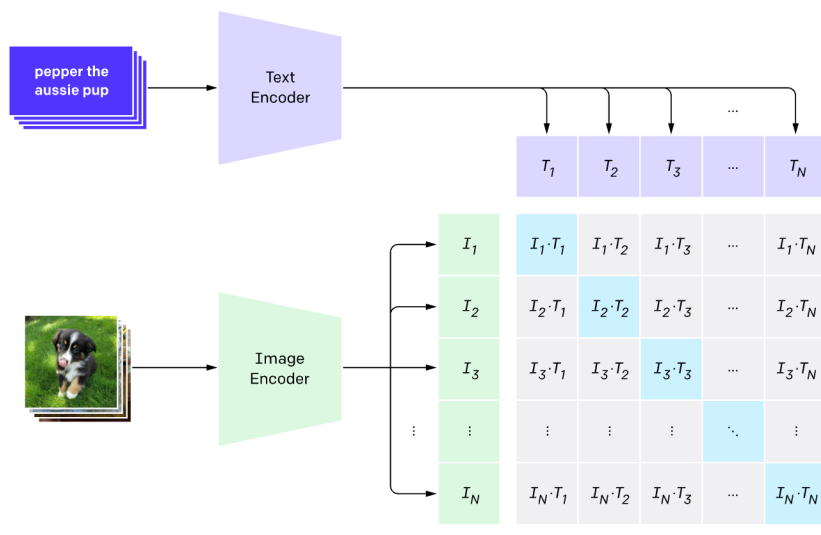
视觉大模型 (VLMs)

- 面向多模态数据(text-image)的大语言模型

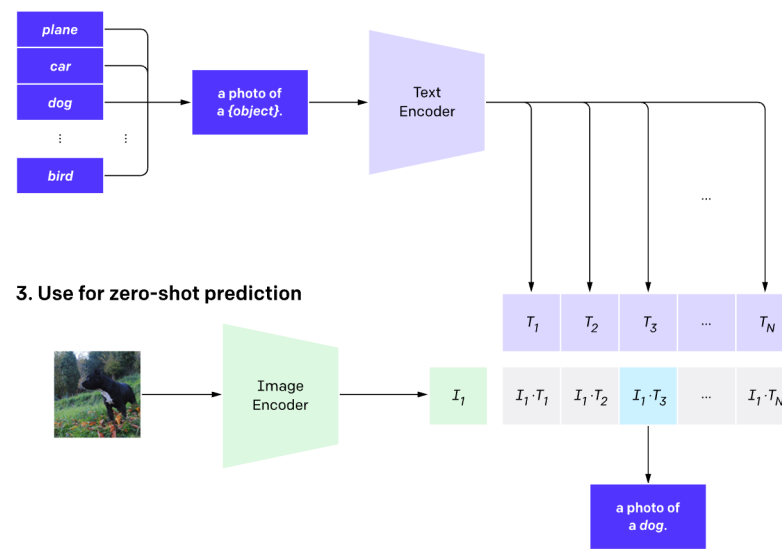


CLIP: 对比式图文预训练

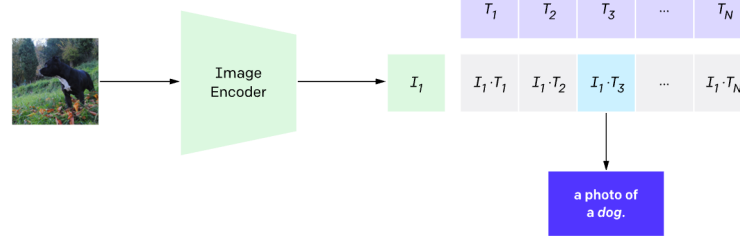
1. Contrastive pre-training



2. Create dataset classifier from label text



3. Use for zero-shot prediction

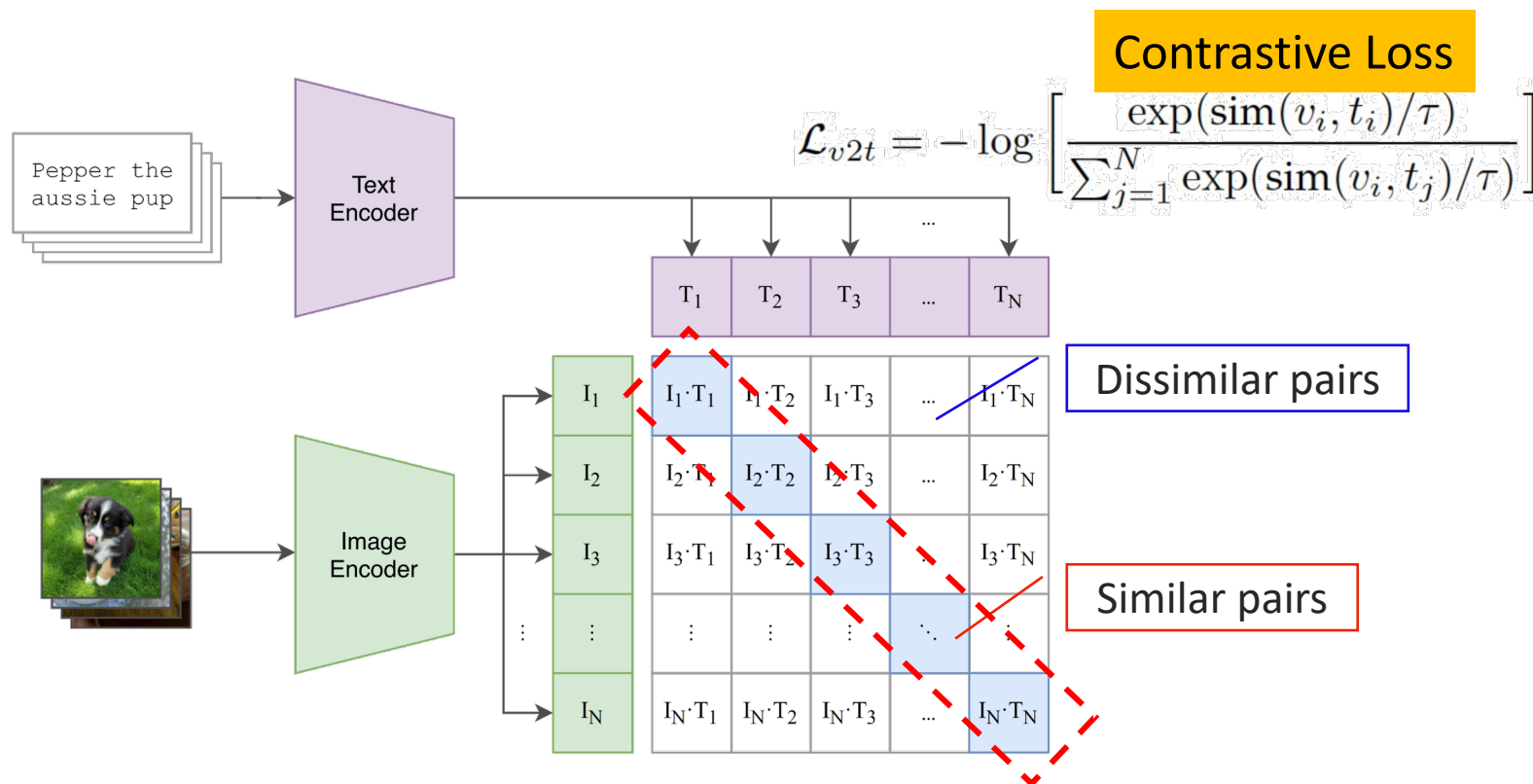


图文对比学习：分别将一组图片输入图片编码器，文本输入文本编码器，计算编码向量的两两余弦距离。

图文检索：根据标签构造文字描述，分别将图文用训练好的编码器进行编码，计算相似度最高的。

训练阶段：图文对比学习

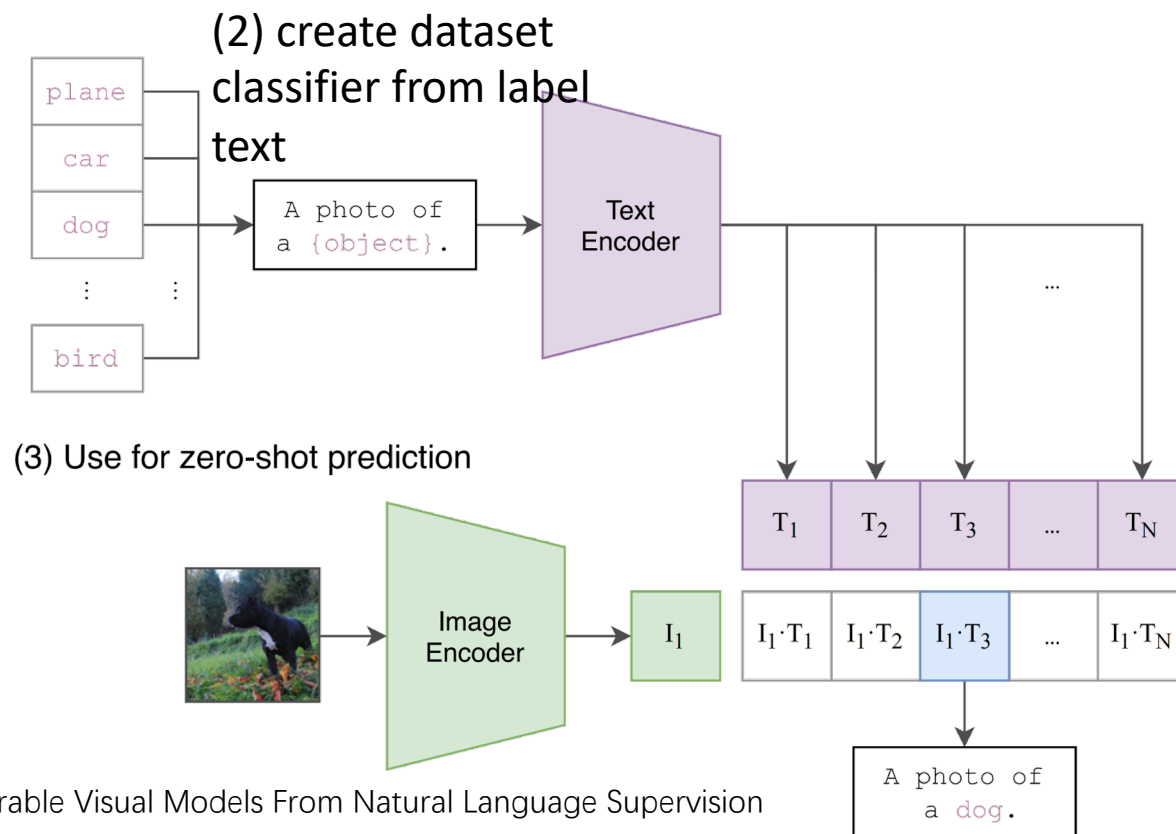
- 分别将一组图片输入图片编码器，文本输入文本编码器，计算编码向量的两两余弦距离。通过对比学习优化。



Learning Transferable Visual Models From Natural Language Supervision

推理阶段：图文检索

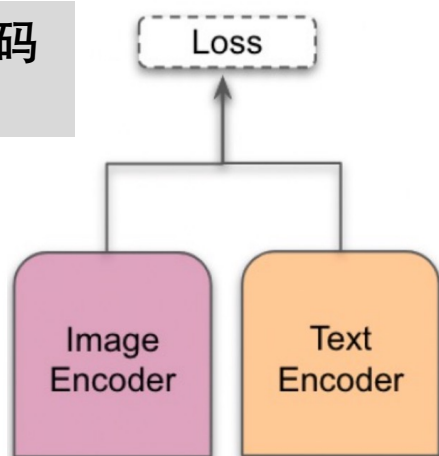
- 根据标签构造文字描述，分别将图文用训练好的编码器进行编码，计算相似度最高的。



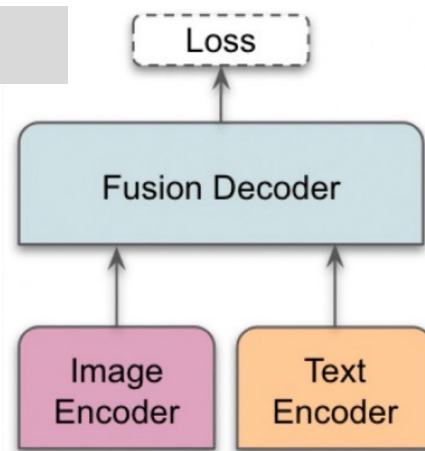
Learning Transferable Visual Models From Natural Language Supervision

不同架构的VLM

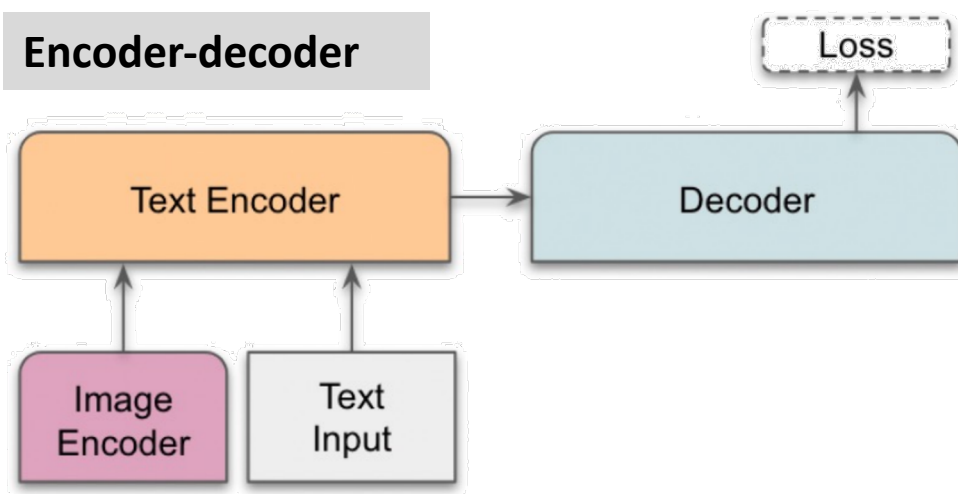
双编码器



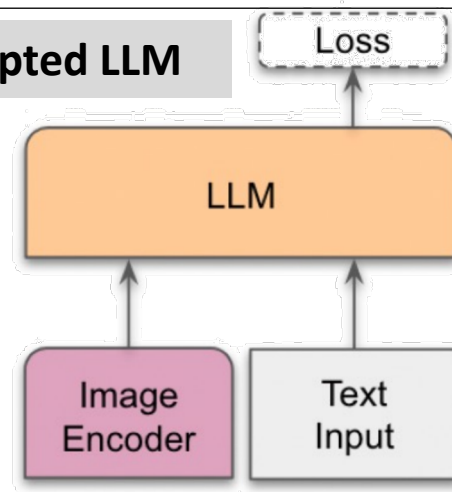
融合



Encoder-decoder

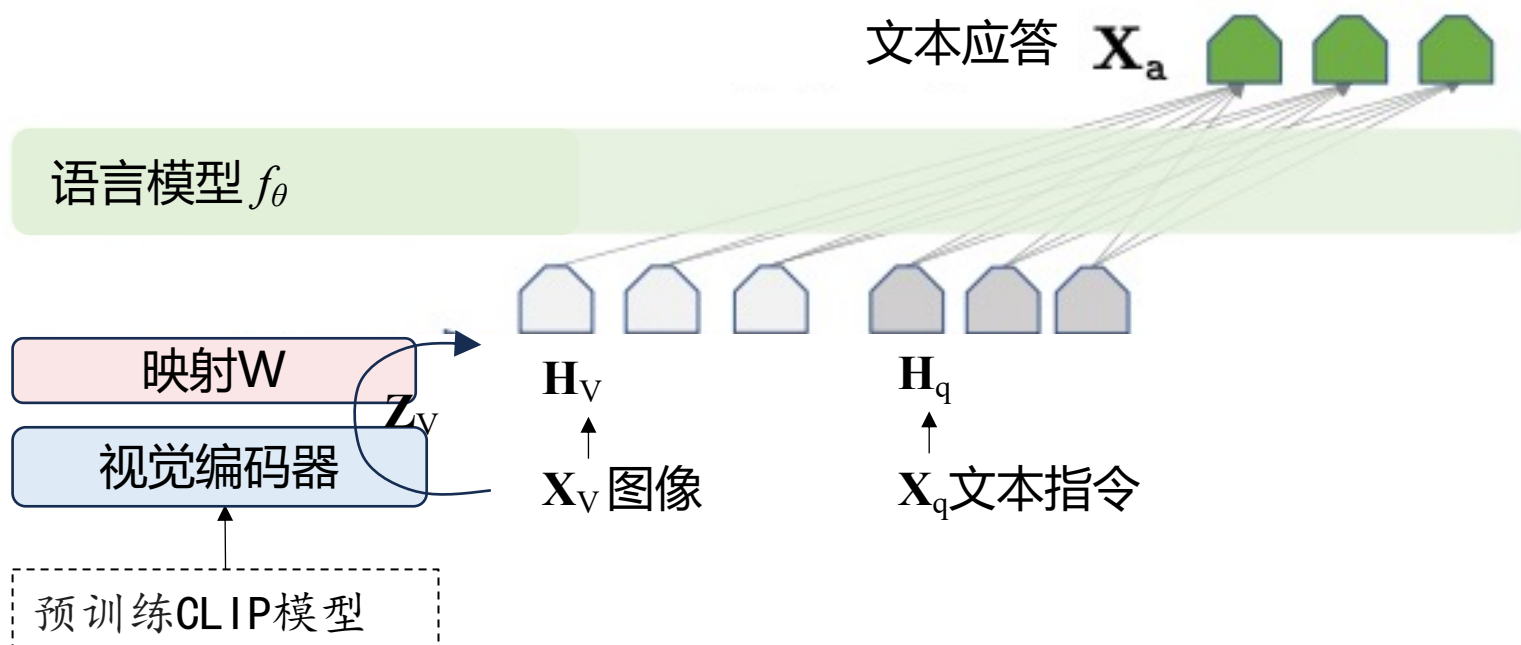


Adapted LLM



LLaVA

- Extends large language models like [LLaMa](#) with visual inputs



TIME for Coding



- **Tutorial: Visual QA with LLaVa**

https://colab.research.google.com/github/NielsRogge/Transformers-Tutorials/blob/master/LLaVa/Inference_with_LLaVa_for_multimodal_generation.ipynb#scrollTo=PuwVAAOinC8q

<https://huggingface.co/blog/vlms>

注意：

```
inputs = processor(prompts, images=[image1, image2],  
padding=True, return_tensors="pt").to("cuda")
```

需改成

```
inputs = processor(prompts[0], image1,  
padding=True, return_tensors="pt").to("cuda")
```


What's Next?

- Unsupervised Learning

物以類聚

Clustering

Learn hidden structures

- K-means
- Gaussian Mixture

無中生有

Generative

density estimation

- GAN
- VAE
- Diffusion

化繁為簡

Dimensional Reduction

Learning data compression

- Principal component analysis
- Auto-encoder

