Transform in Computer Vision

分享人：梁瑛平

北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

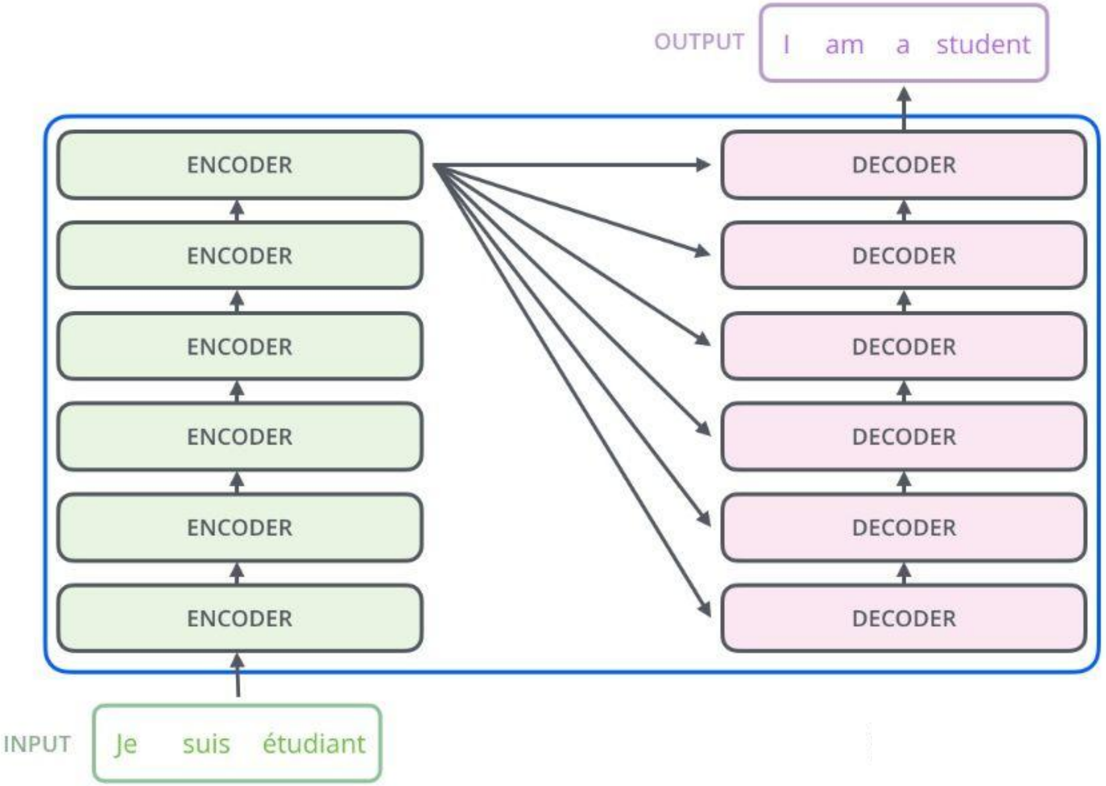
时间：2021-04-11

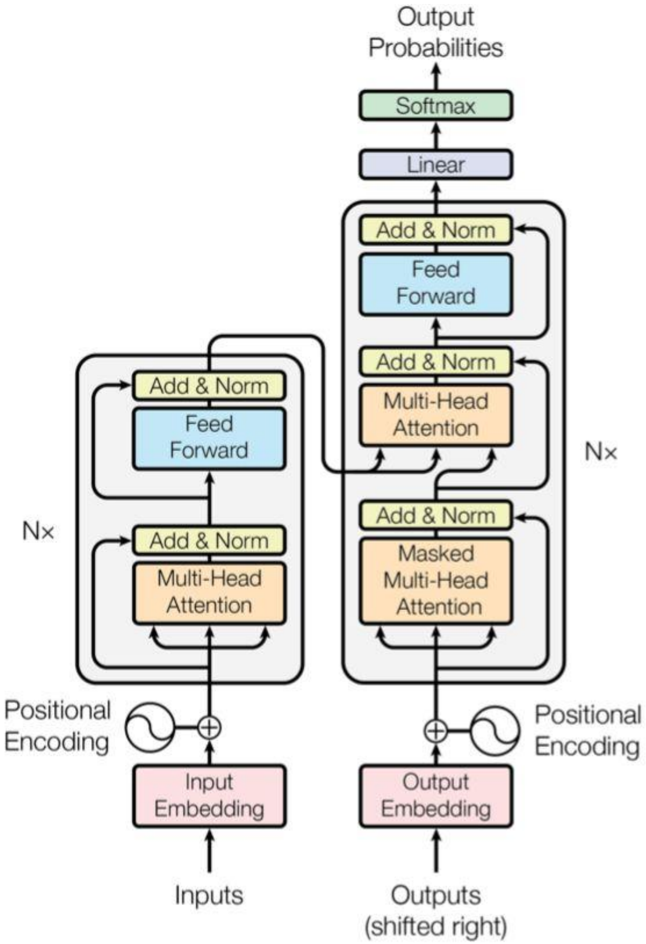
北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

目录

CONTENTS

《Attention Is All You Need》





《Attention Is All You Need》

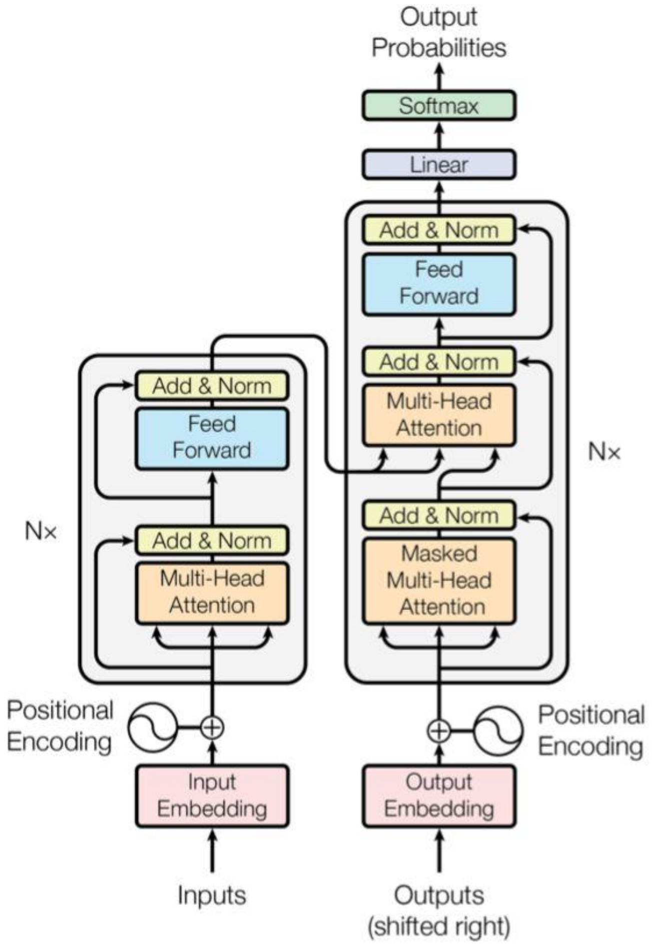
Encoder:

$$LayerNorm(x + Sublayer(x))$$

Attention:

$$Attention(Q, K, V) = softmax(\frac{Q^T K}{\sqrt{d_k}})V$$

$Q \in R^{m \times d_k}$, $K \in R^{m \times d_k}$, $V \in R^{m \times d_v}$



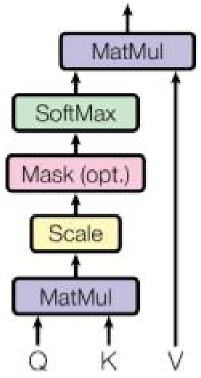
Introduction

《Attention Is All You Need》

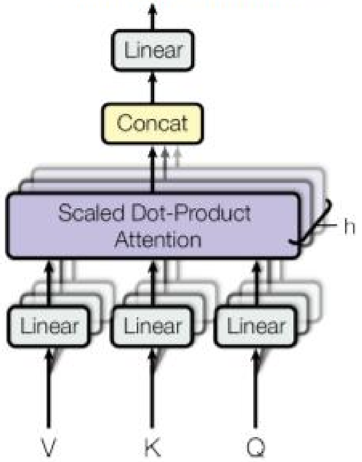
$$Attention(Q,K,V) = softmax(\frac{Q^T K}{\sqrt{d_k}})V$$

$$Q \in R^{m \times d_k}\text{ , }K \in R^{m \times d_k}\text{ , }V \in R^{m \times d_v}$$

Scaled Dot-Product Attention



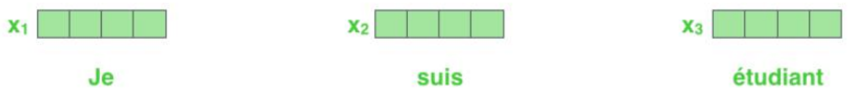
Multi-Head Attention



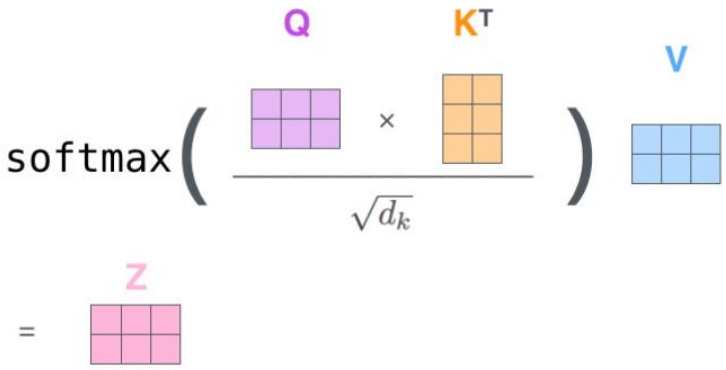
Introduction

《Attention Is All You Need》

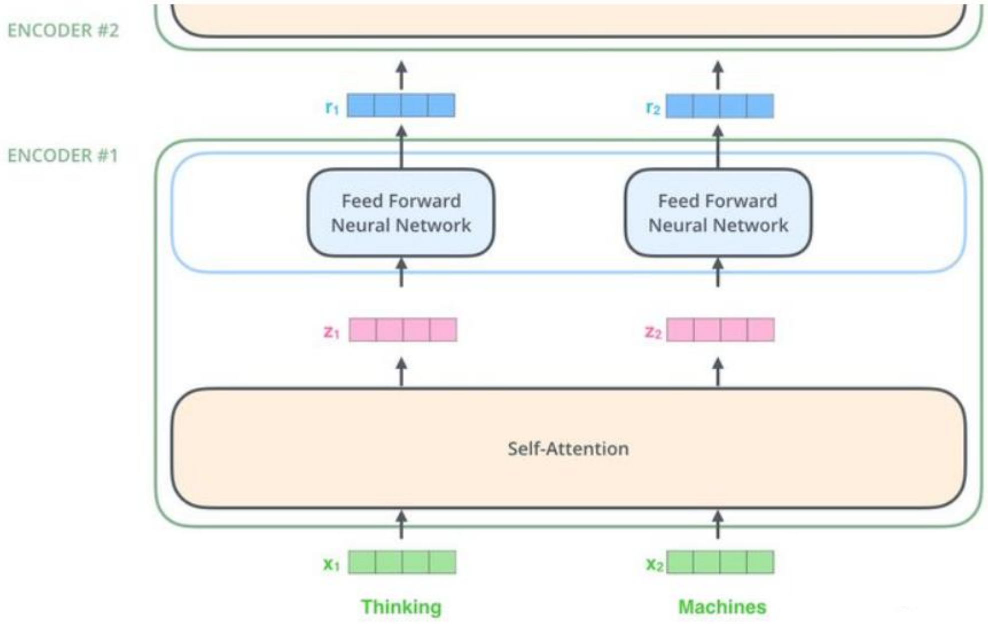
Input：



Self-Attention：

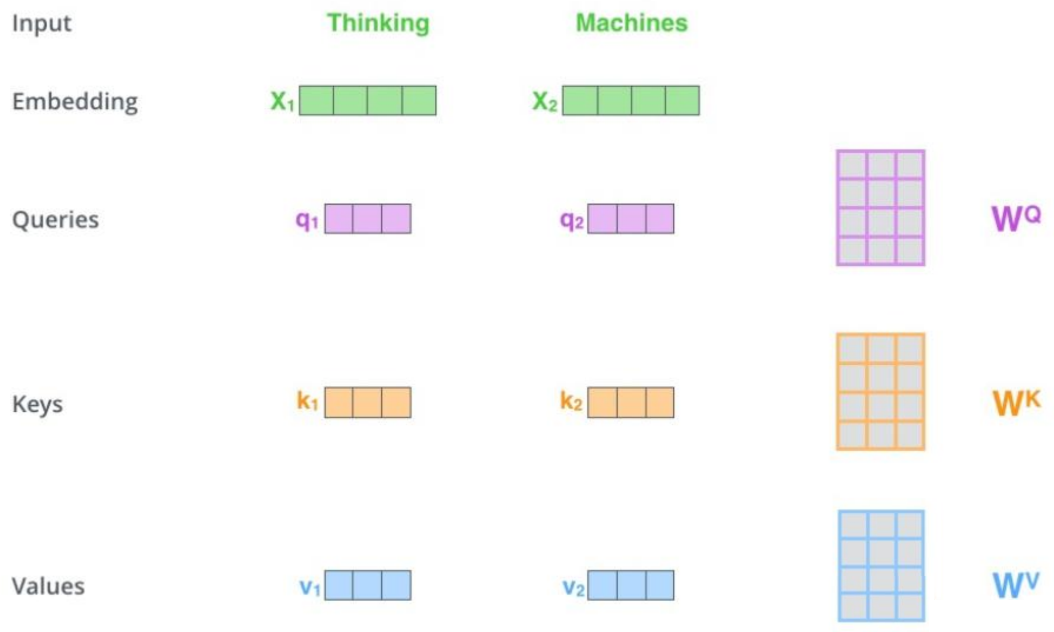


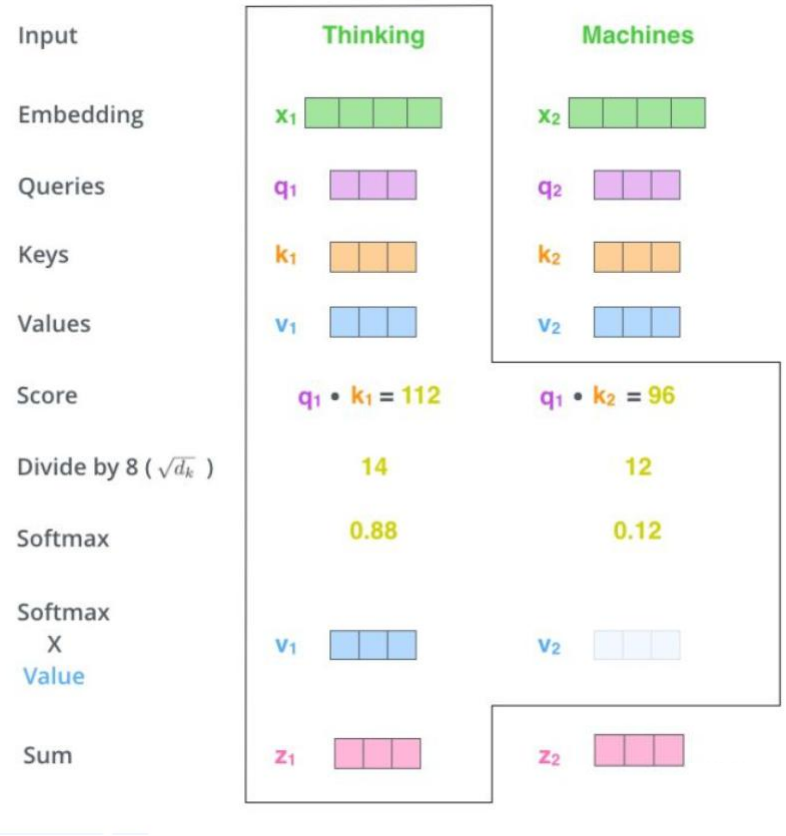
Encoders：



《Attention Is All You Need》

Self-Attention：

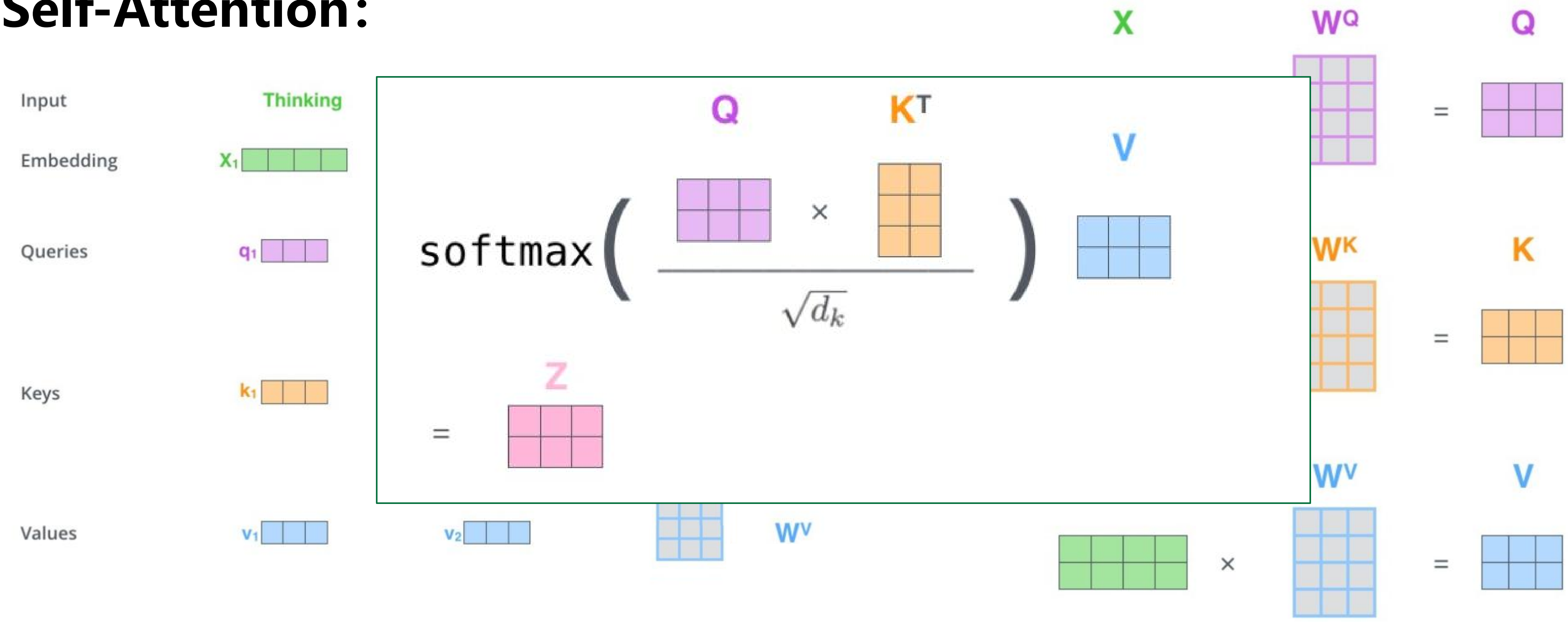




Introduction

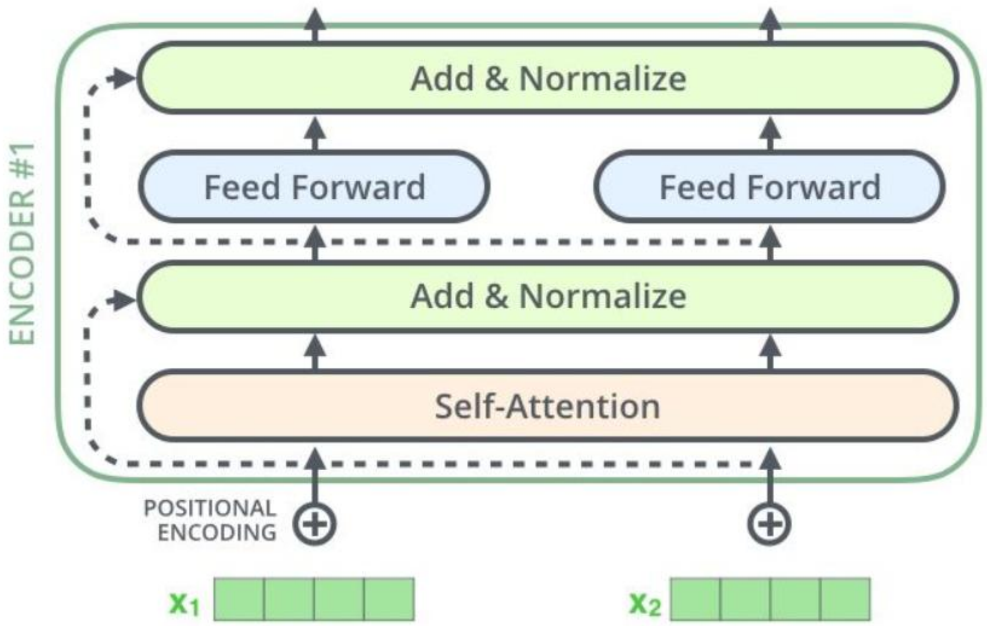
《Attention Is All You Need》

Self-Attention：



《Attention Is All You Need》

Skip Connection:



Multi-Head:

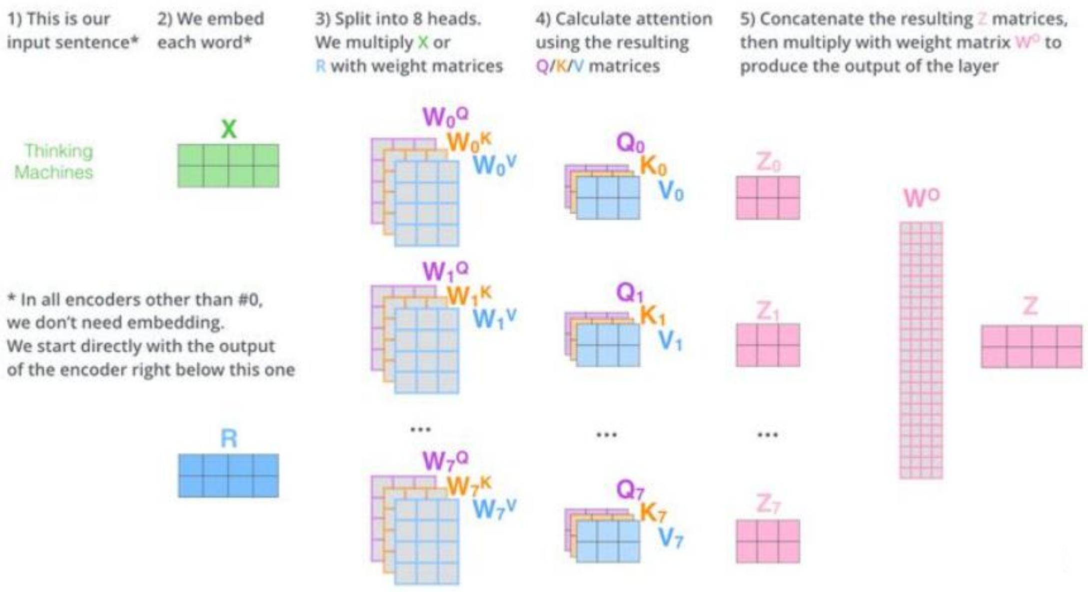
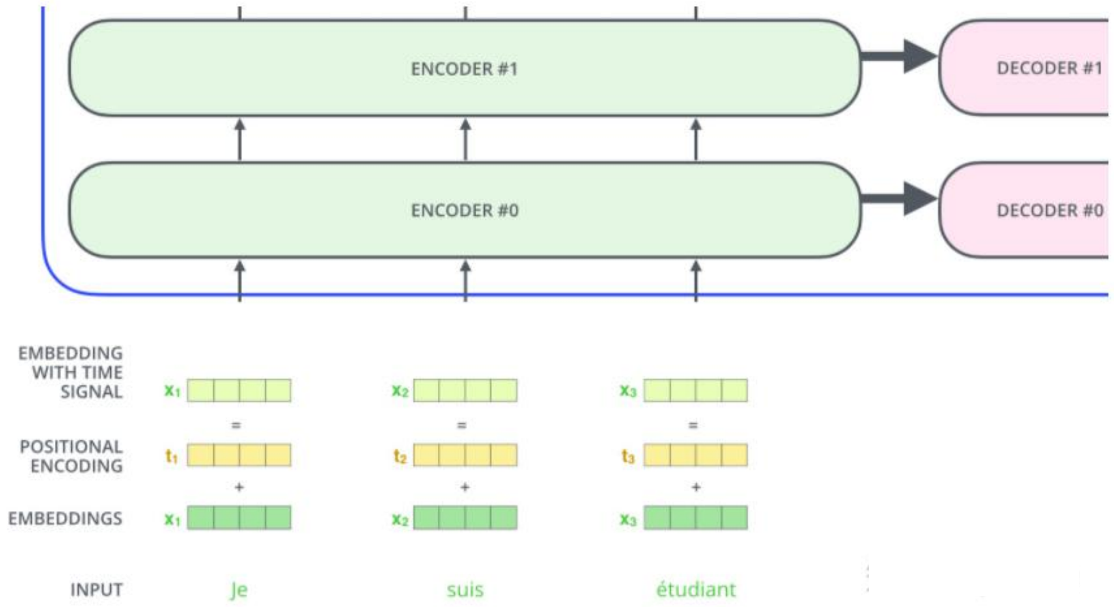


图14：Multi-Head Attention

《Attention Is All You Need》

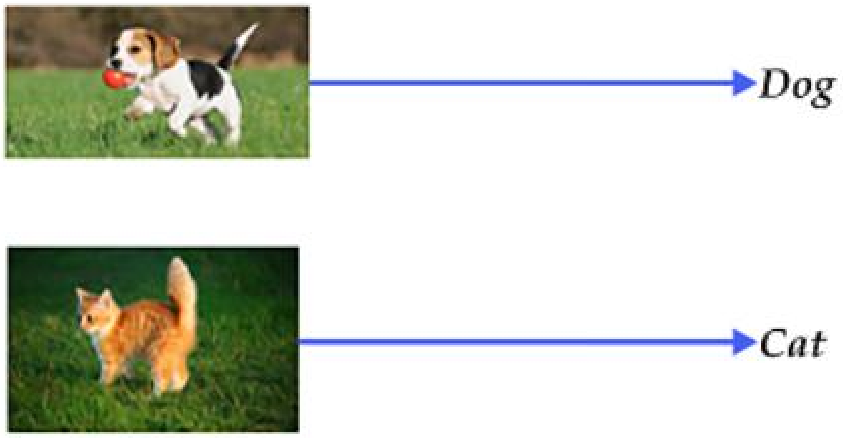
Position Embedding：



$$PE(pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}})$$

$$PE(pos, 2i+1) = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}})$$

Image Classification：Introduction



General image classification

Fine-grained image classification

Image Classification：Challenge

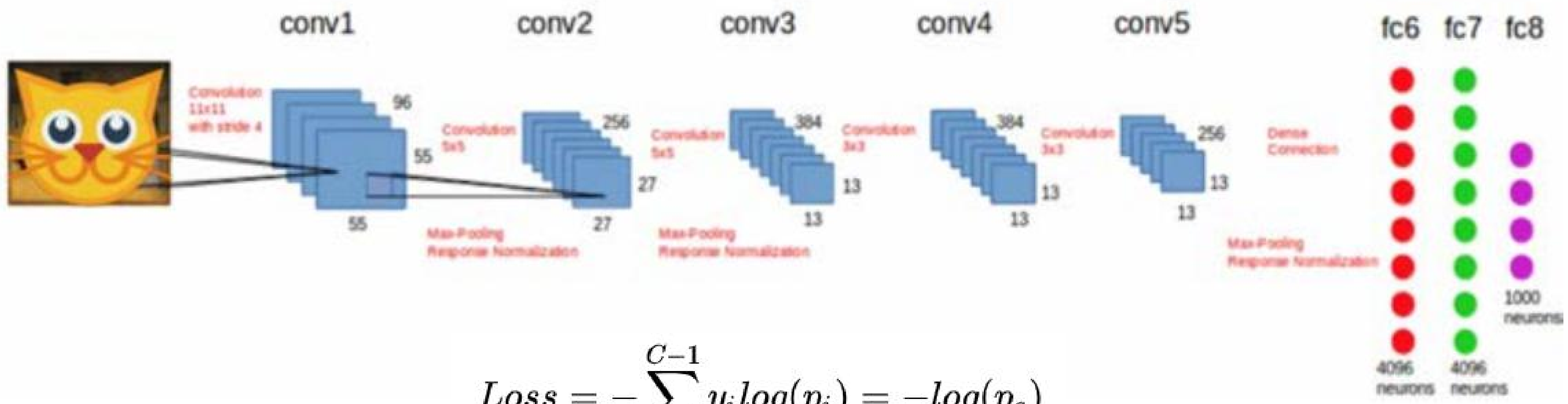
Image Classification：classical model

The most common form of a ConvNet architecture stacks a few CONV-RELU layers, follows them with POOL layers, and repeats this pattern until the image has been merged spatially to a small size.

ConvNet Architecture

$INPUT \rightarrow [[CONV \rightarrow RELU]*N \rightarrow POOL?]*M \rightarrow [FC \rightarrow RELU]*K \rightarrow FC$

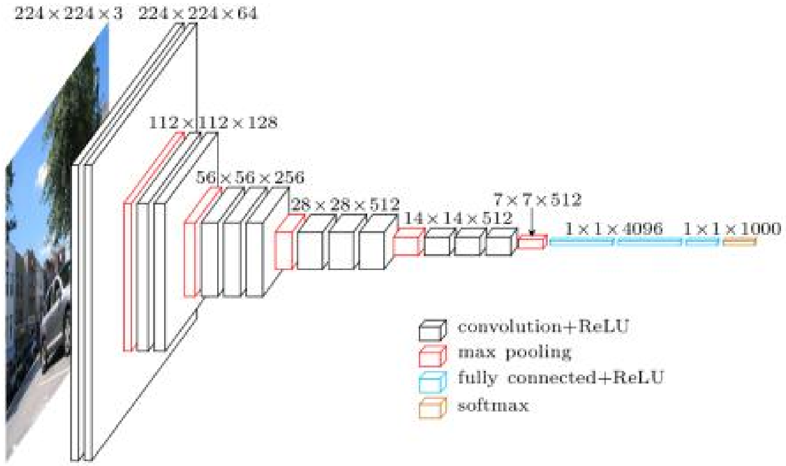
where the $*$ indicates repetition, and the $POOL?$ indicates an optional pooling layer.



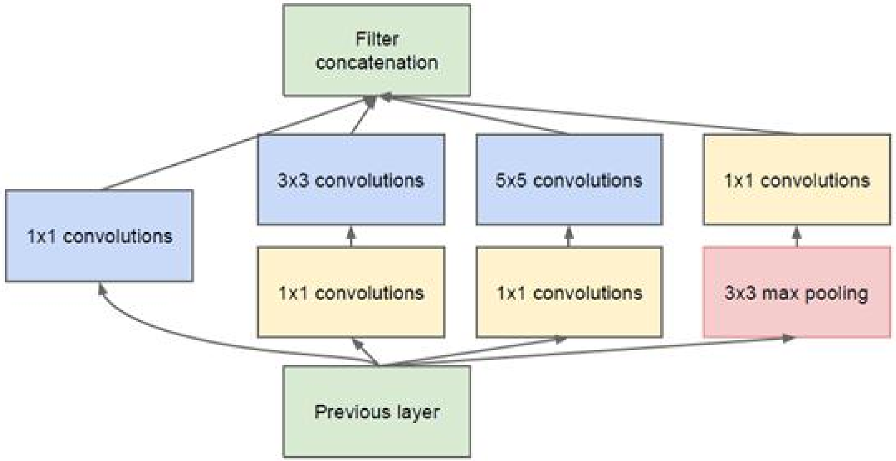
$$Loss = -\sum_{i=0}^{C-1} y_i log(p_i) = -log(p_c)$$

Image Classification：classical model

VGG：



GoogleNet：



ResNet：

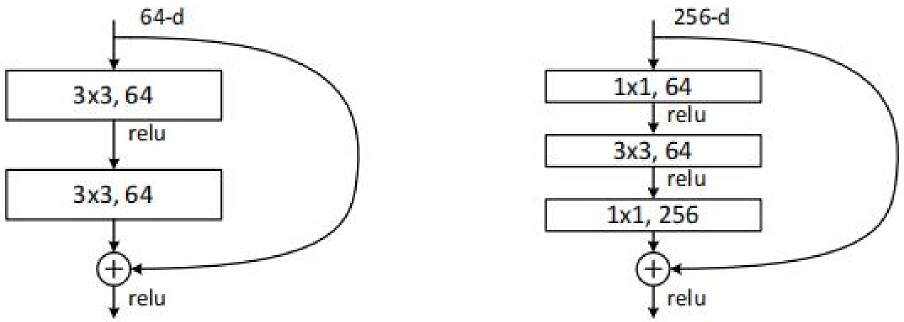


Image Classification：using transformer

《An Image is Worth 16x16 Words:Transformers for Image Recognition at Scale》

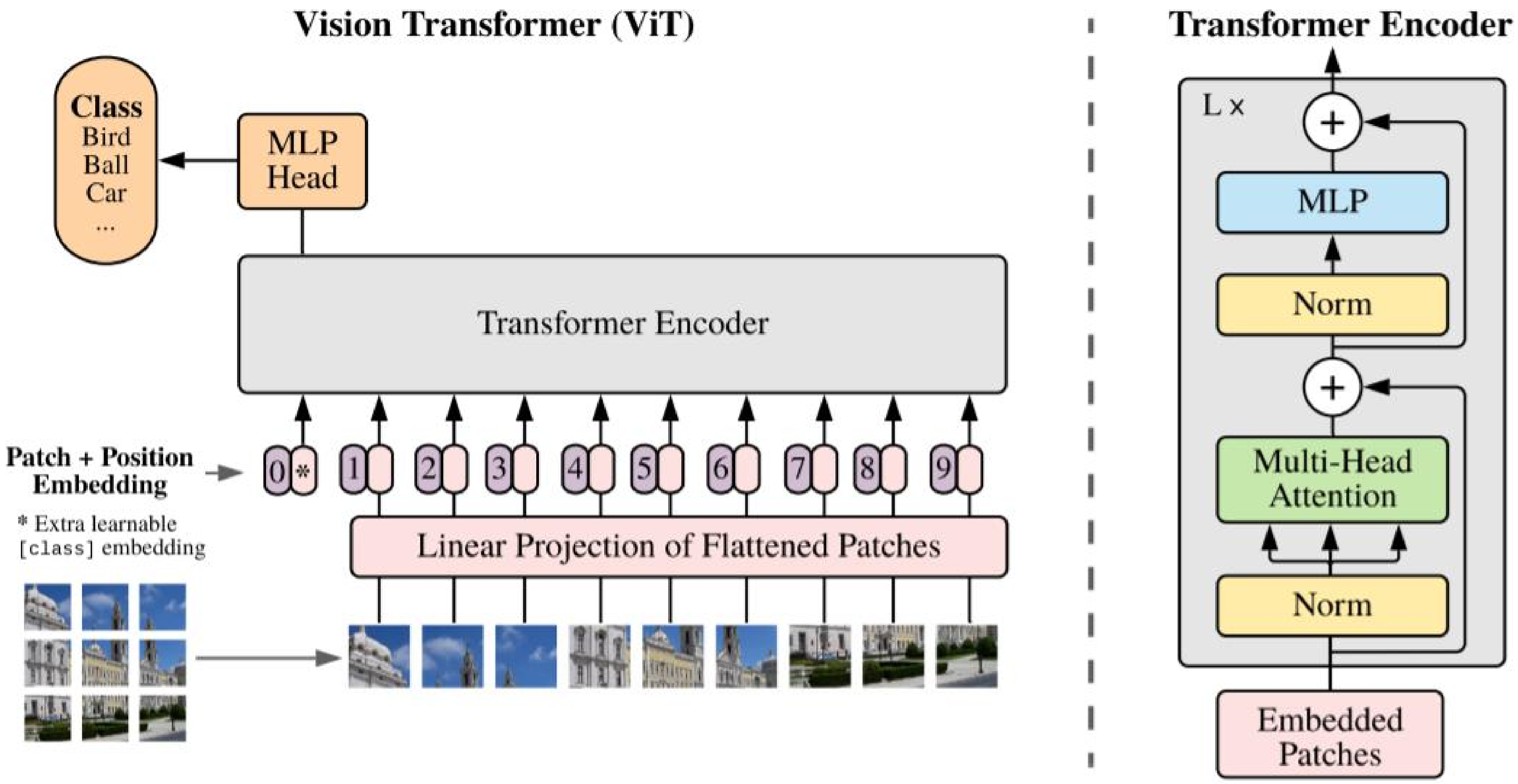
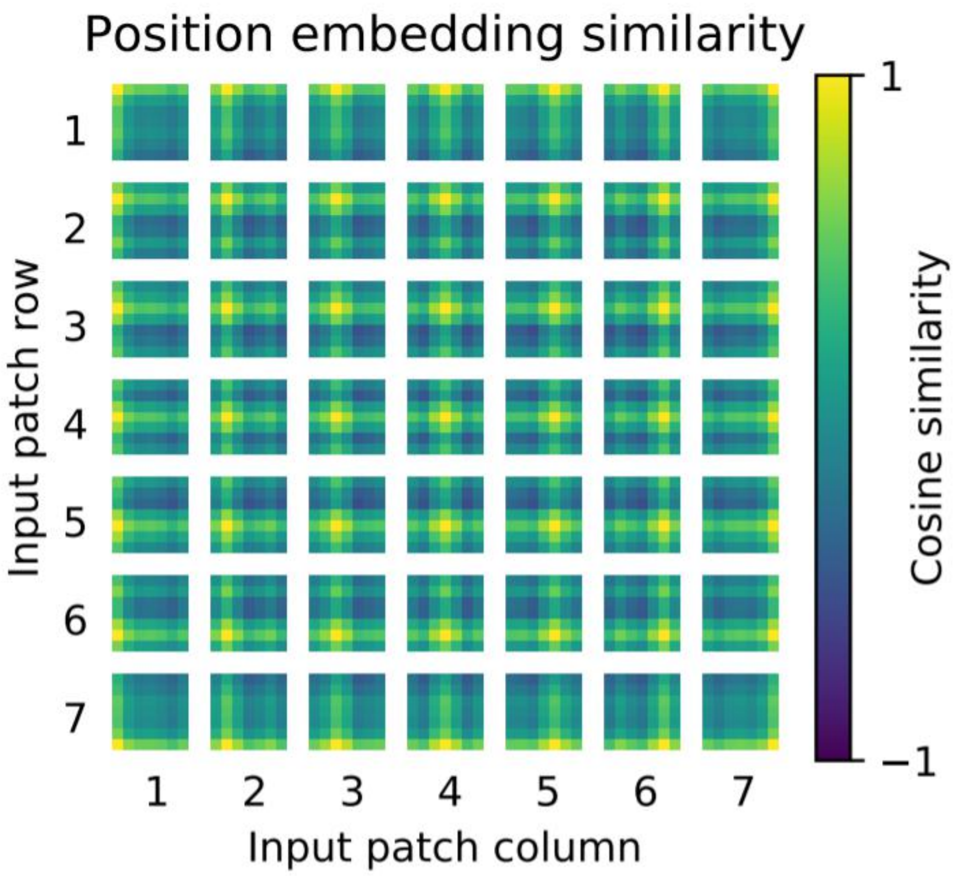
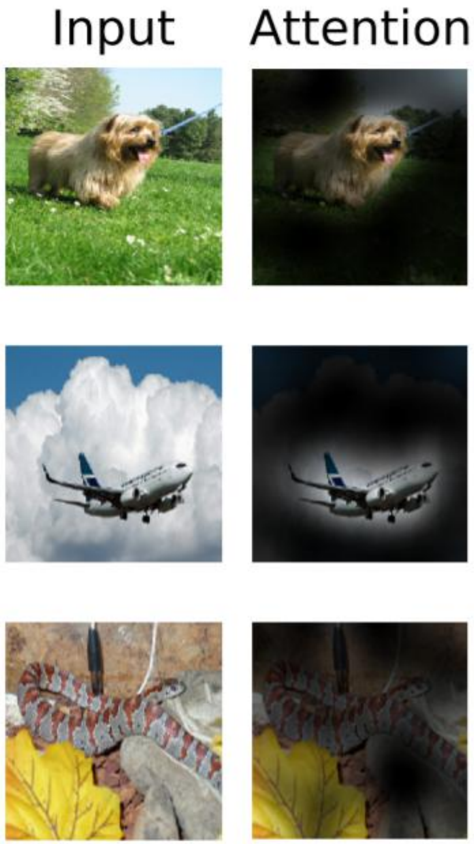


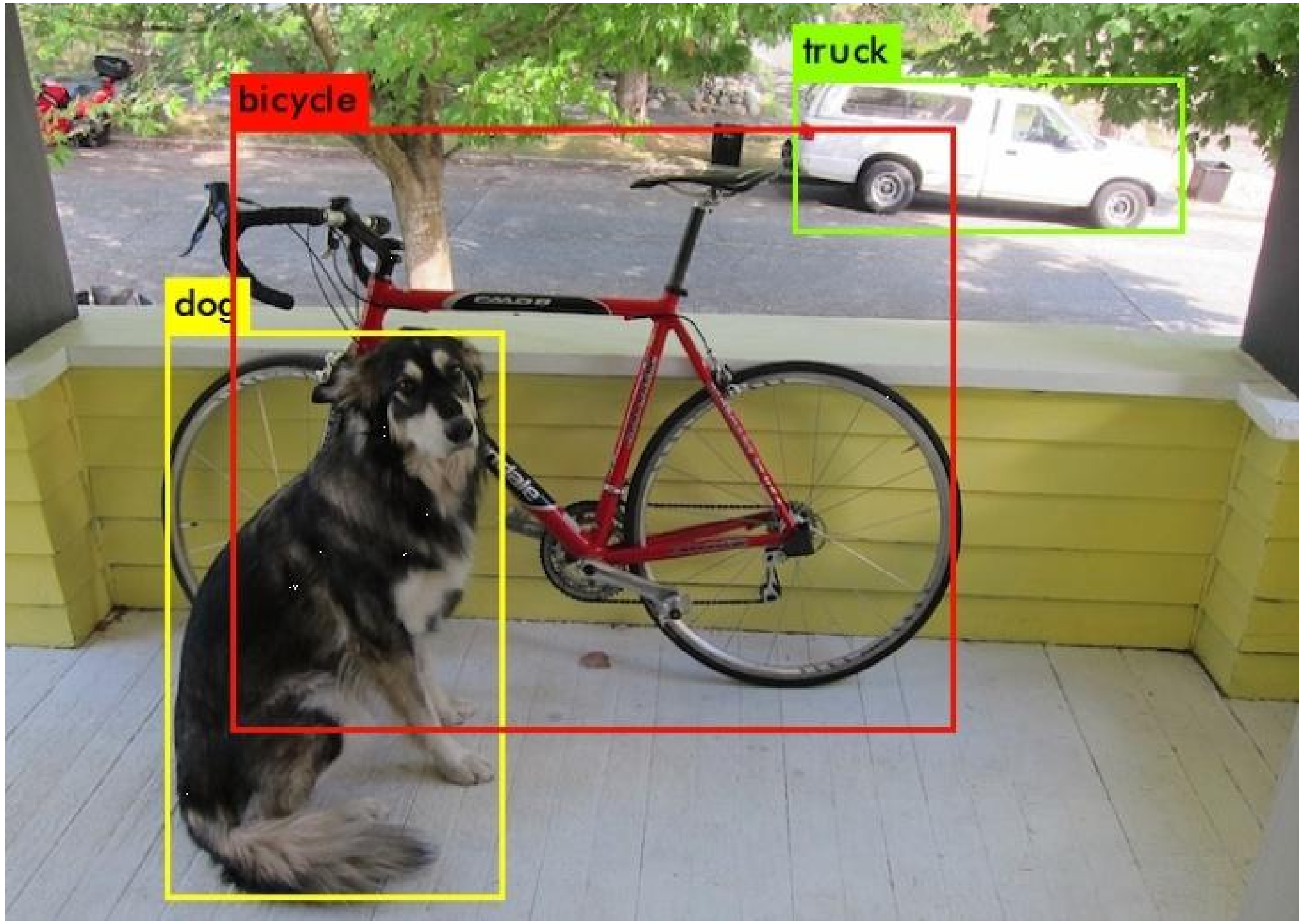
Image Classification：using transformer

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

《An Image is Worth 16x16 Words:Transformers for Image Recognition at Scale》







Problems to be solved:

1. Positions
2. Classifications
3. Unordered set

Object Detection：classical model

Faster-RCNN：

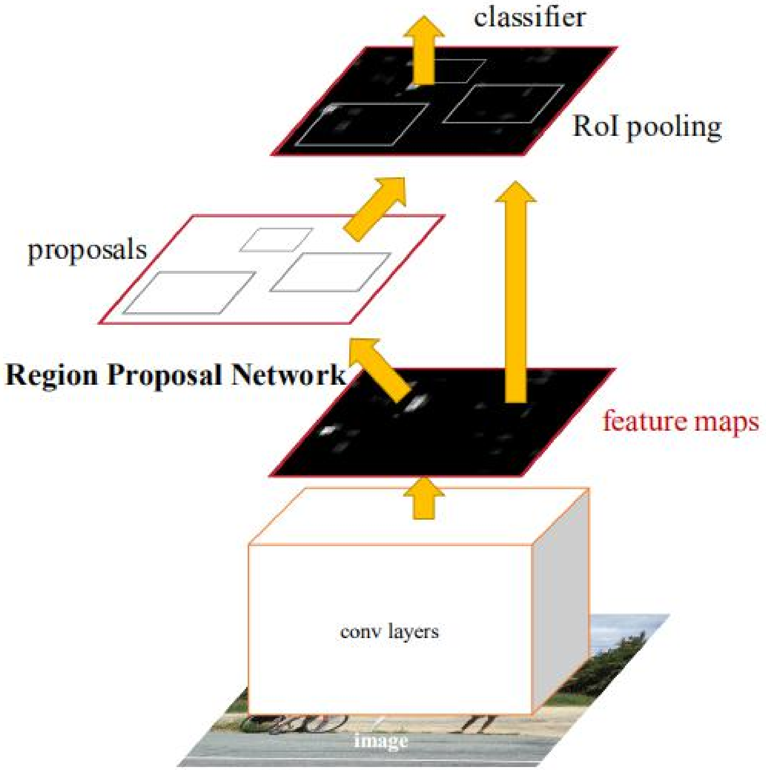
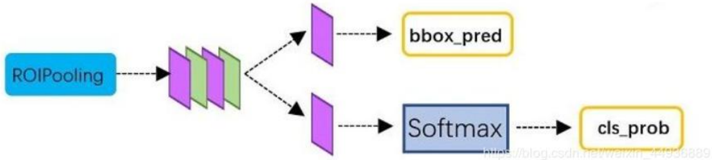
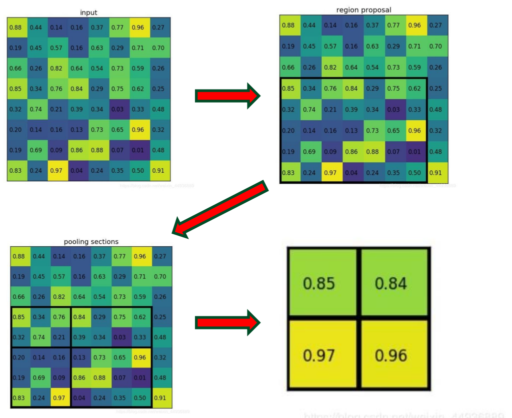
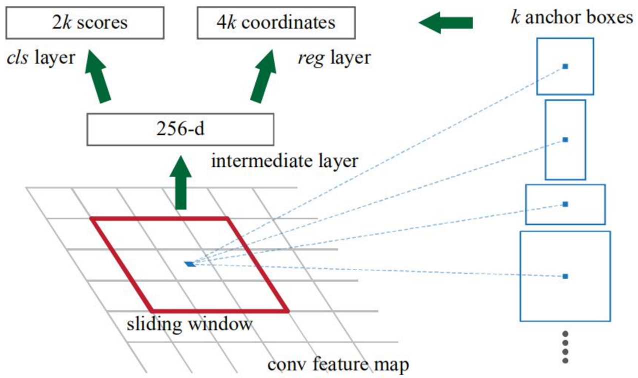


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the ‘attention’ of this unified network.

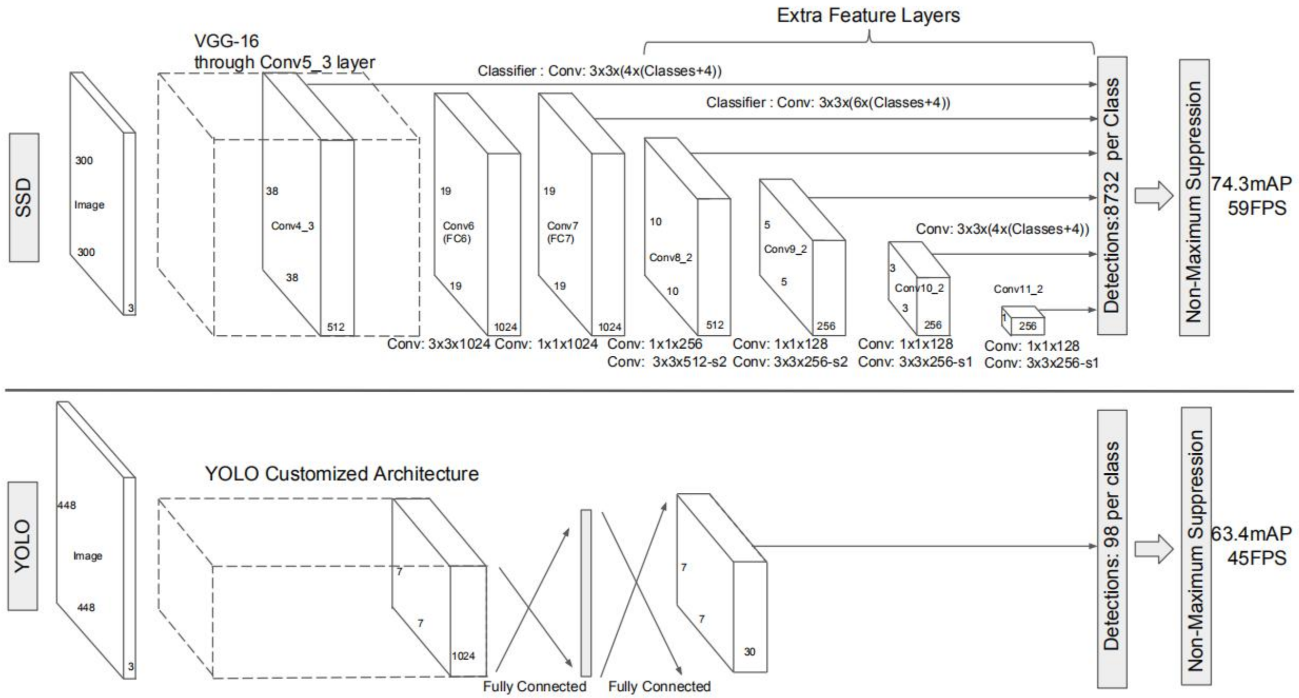


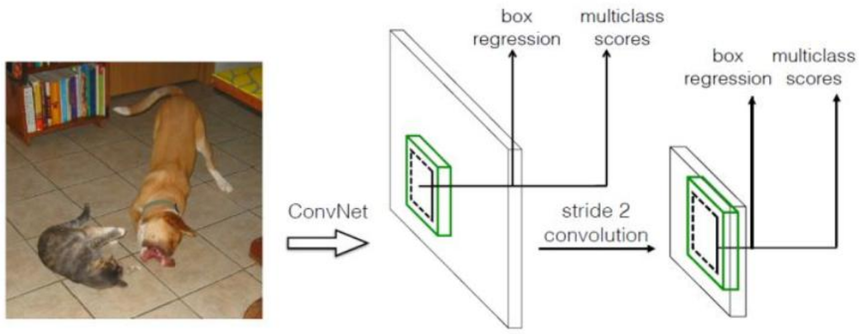




Object Detection：classical model

SSD & YOLO：





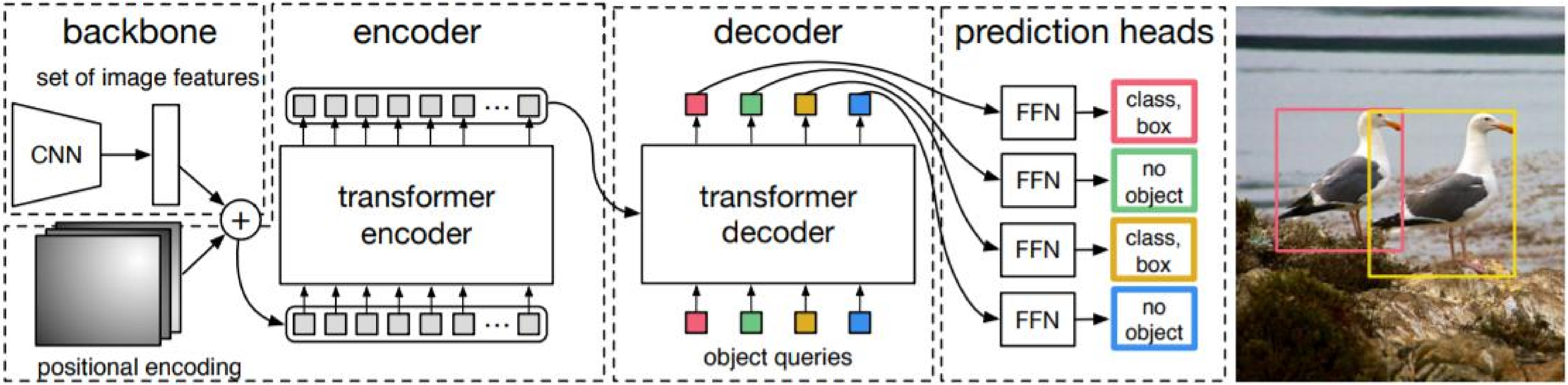
Loss Function:

$$
\begin{aligned}
t_{\mathrm{x}} &= (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}}, \\
t_{\mathrm{w}} &= \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}), \\
t^{*}_{\mathrm{x}} &= (x^{*} - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t^{*}_{\mathrm{y}} = (y^{*} - y_{\mathrm{a}})/h_{\mathrm{a}}, \\
t^{*}_{\mathrm{w}} &= \log(w^{*}/w_{\mathrm{a}}), \quad t^{*}_{\mathrm{h}} = \log(h^{*}/h_{\mathrm{a}}),
\end{aligned}
$$

$$
\begin{aligned}
L(\{p_i\}, \{t_i\}) &= \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\
&+ \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).
\end{aligned}
$$

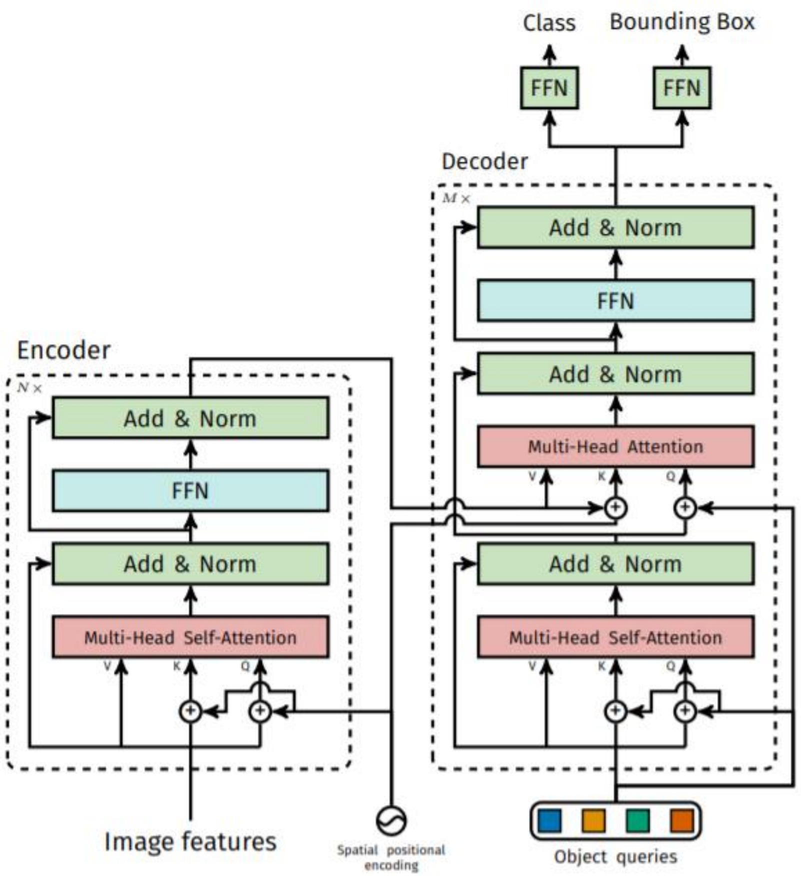
Object Detection：using transformer

《End-to-End Object Detection with Transformers》



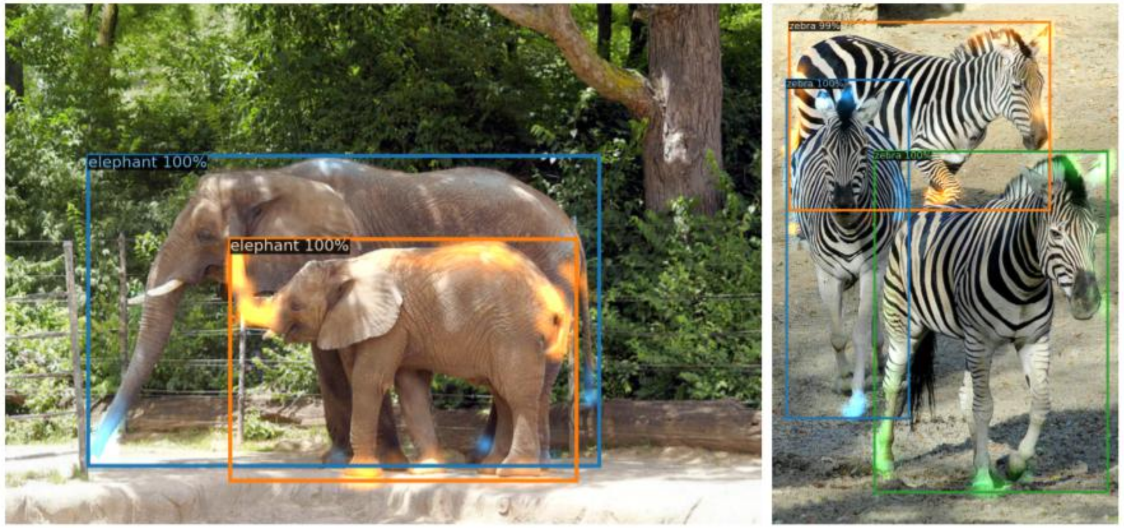
Object Detection：using transformer

《End-to-End Object Detection with Transformers》



$$\hat{\sigma} = \arg\min_{\sigma\in\mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}),$$

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right] ,$$



Object Detection：using transformer

《End-to-End Object Detection with Transformers》

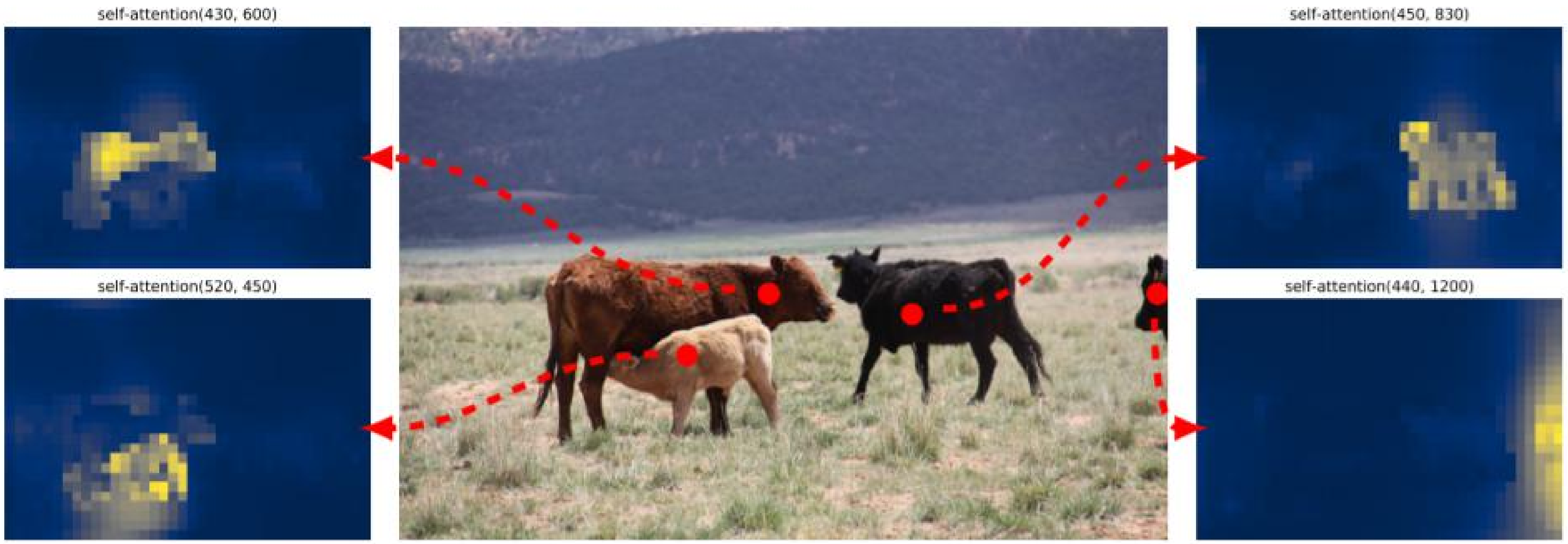
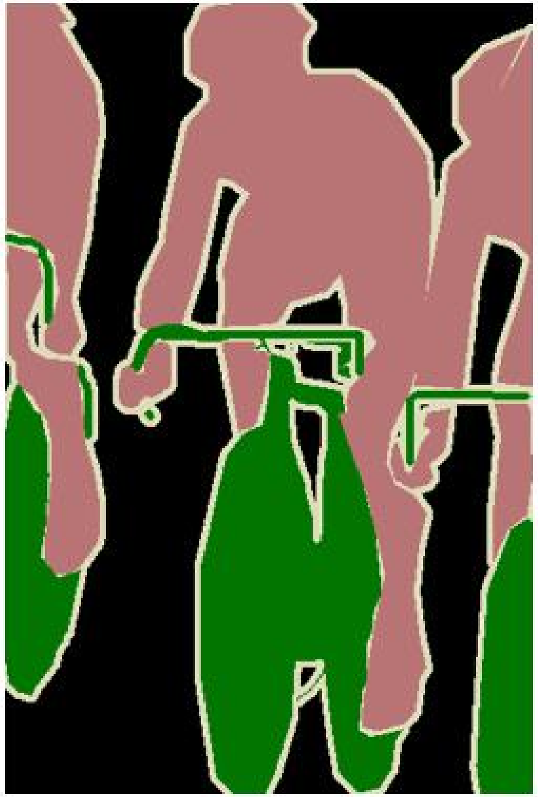


Fig. 3: Encoder self-attention for a set of reference points. The encoder is able to separate individual instances. Predictions are made with baseline DETR model on a validation set image.

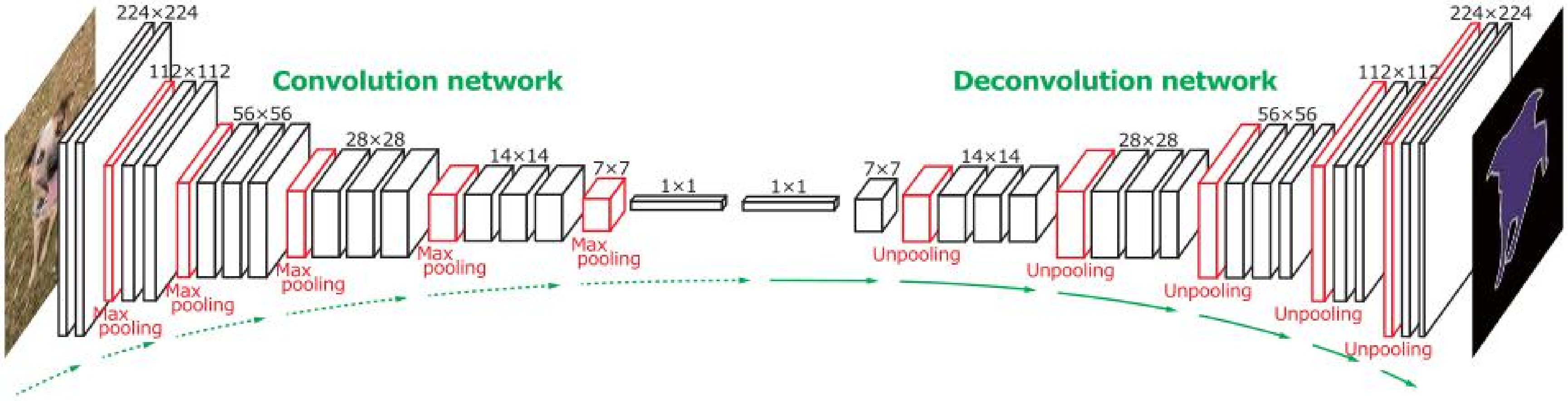
Semantic Segmentation：Introduction

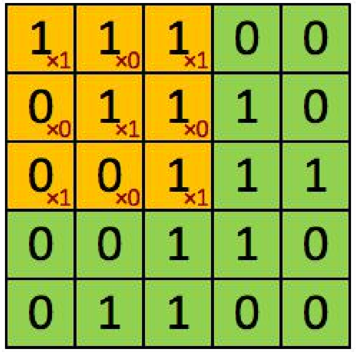


Person
Bicycle
Background

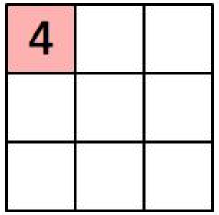
Semantic Segmentation：classical model

FCN：

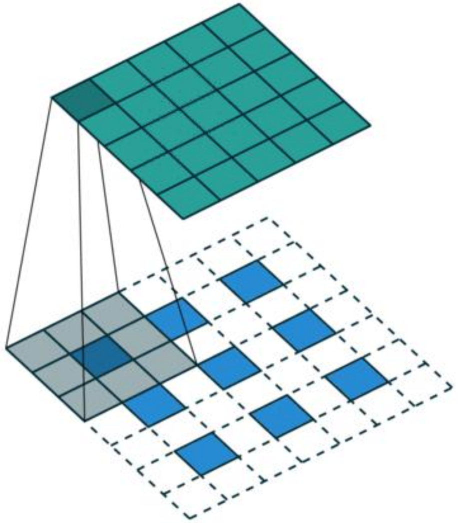




Image

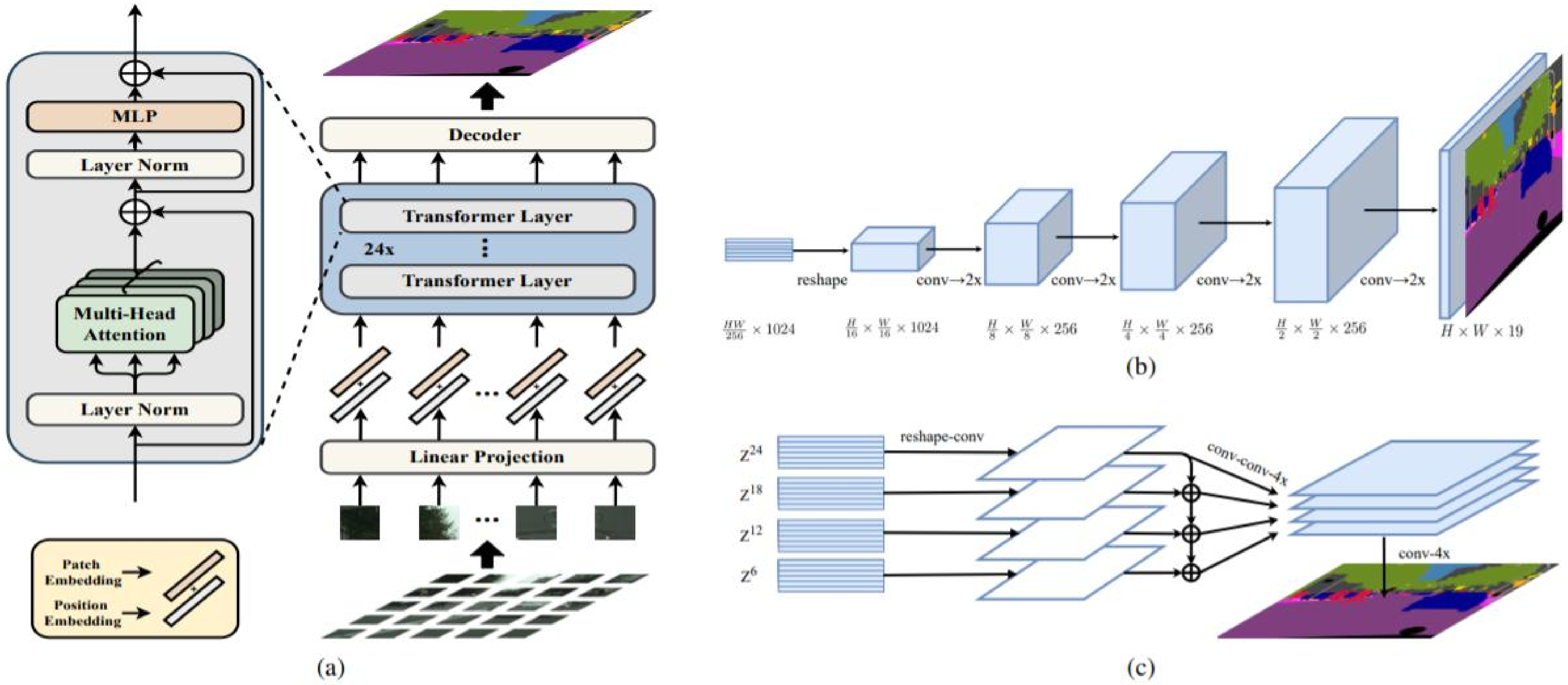


Convolved Feature



Semantic Segmentation：classical model

《Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers》



《Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers》

1 query $= Z^{l-1}\mathbf{W}_Q$, key $= Z^{l-1}\mathbf{W}_K$, value $= Z^{l-1}\mathbf{W}_V$,

2
$$SA(Z^{l-1}) = Z^{l-1} + \mathrm{softmax}(\frac{Z^{l-1}\mathbf{W}_Q(Z\mathbf{W}_K)^\top}{\sqrt{d}})(Z^{l-1}\mathbf{W}_V).$$

3 and project their concatenated outputs: $MSA(Z^{l-1}) = [SA_1(Z^{l-1}); SA_2(Z^{l-1}); \cdots; SA_m(Z^{l-1})]\mathbf{W}_O$, where $\mathbf{W}_O \in \mathbb{R}^{md\times C}$. d is typically set to C/m. The output of

4
$$Z^l = MSA(Z^{l-1}) + MLP(MSA(Z^{l-1})) \in \mathbb{R}^{L\times C}.$$

5
$$\{Z^1, Z^2, \cdots, Z^{L_e}\}$$

Figure 3. **Qualitative results on Pascal Context:** SETR (right column) vs. dilated FCN baseline (left column) in each pair. Best viewed in color and zoom in.

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

谢谢观看
有疑问的同学可以提问

▼

分享人：梁瑛平