Google I/O Extended

Harnessing Transformers for various Computer Vision tasks

Rajesh Shreedhar Bhat
Sr. Data Scientist, Walmart & GDE ML

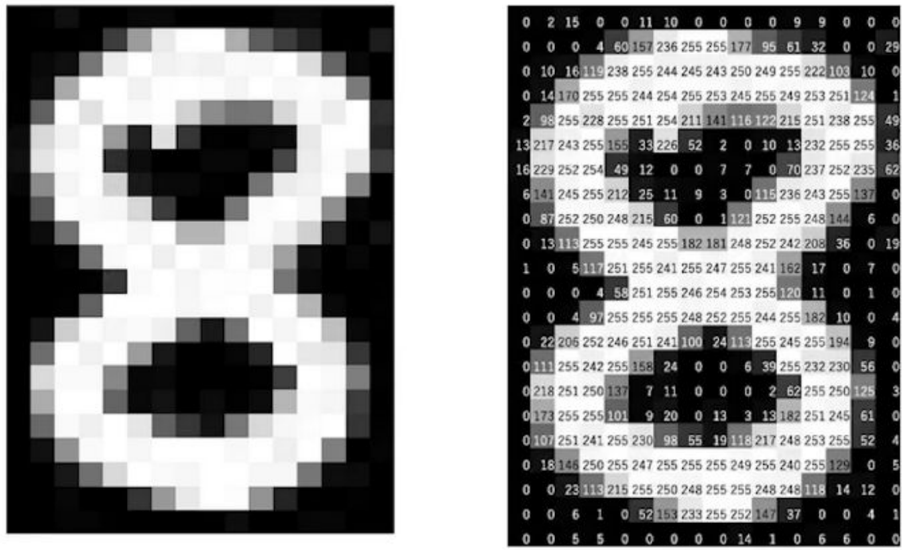
Section 01

Computer Vision and various vision tasks

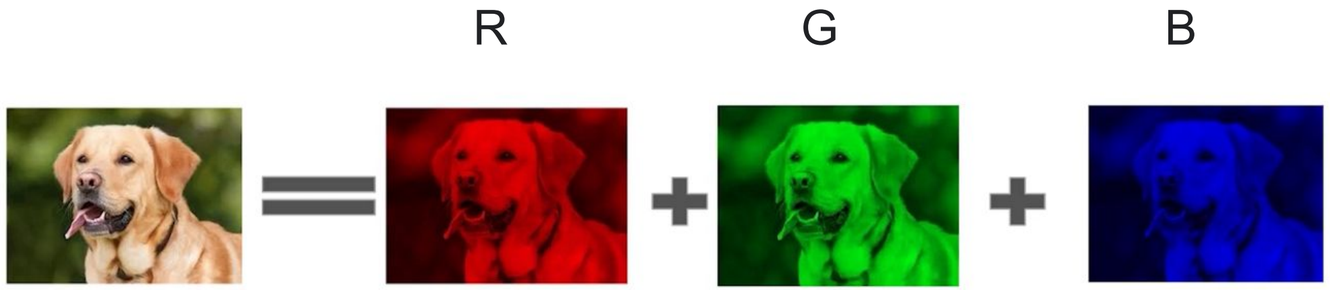
Section 1

Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images and videos.



Grayscale Image



Color Image

Image source: https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/

Section 1

Various Computer Vision tasks

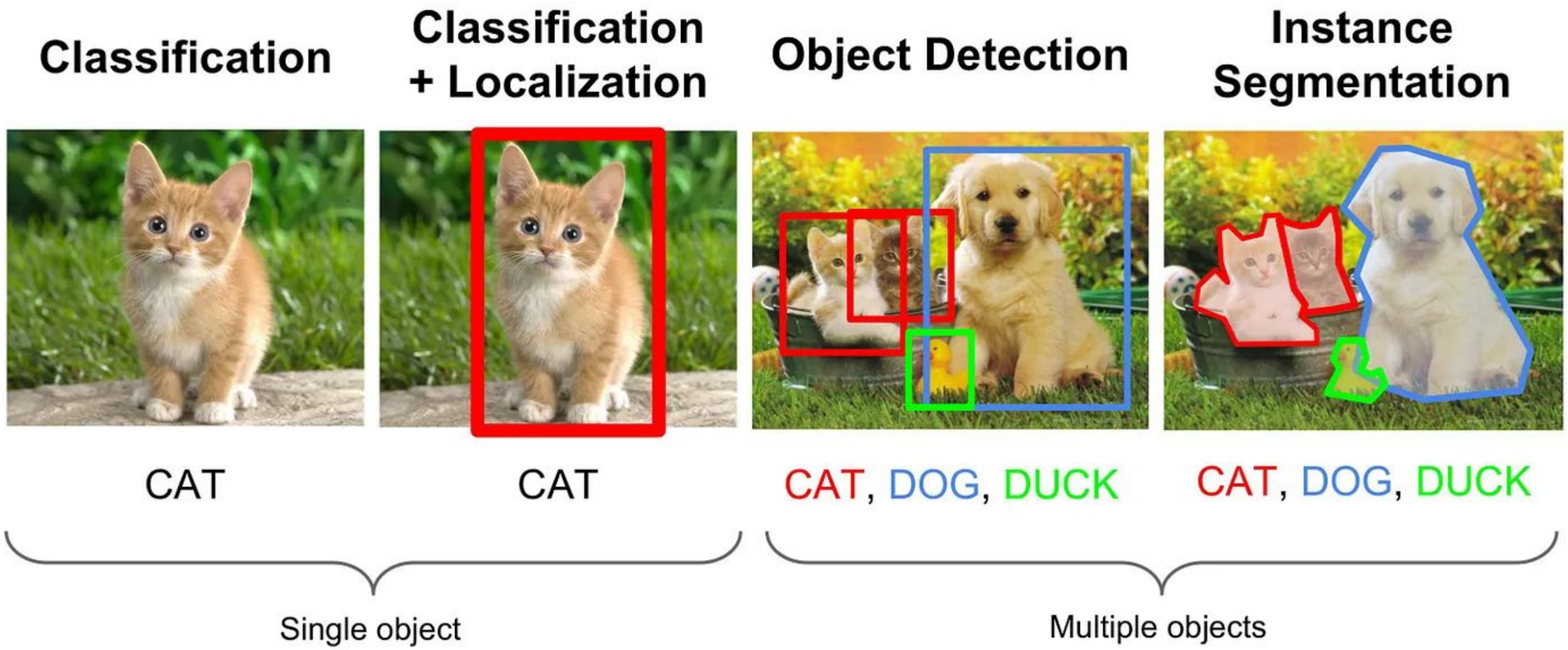
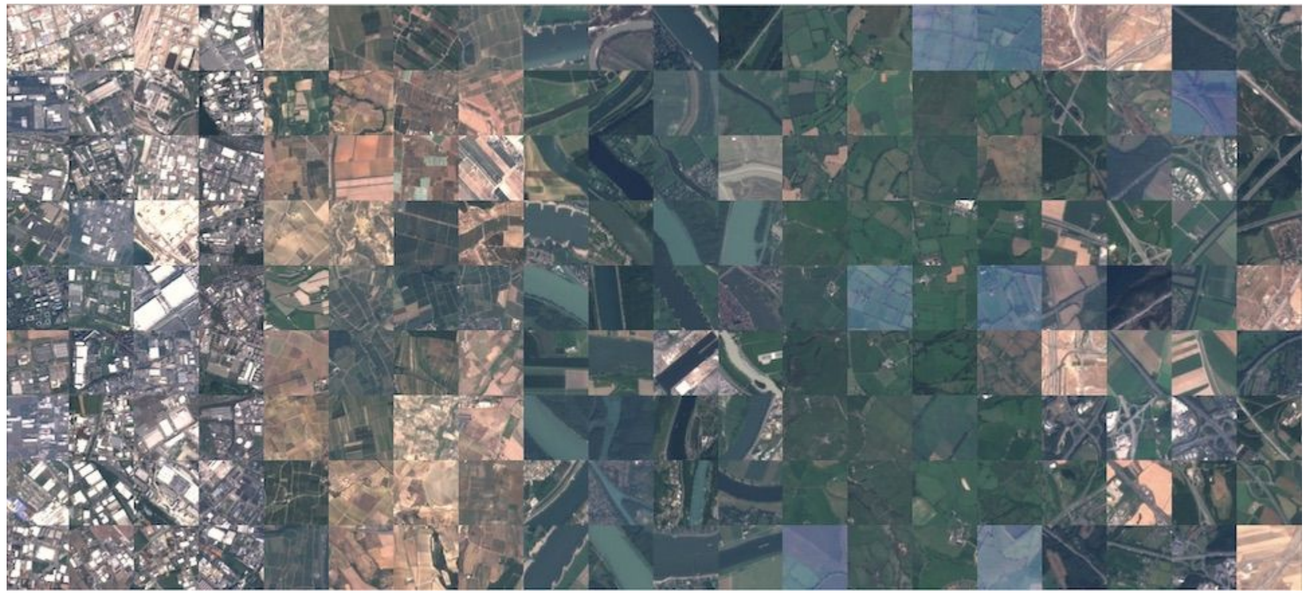


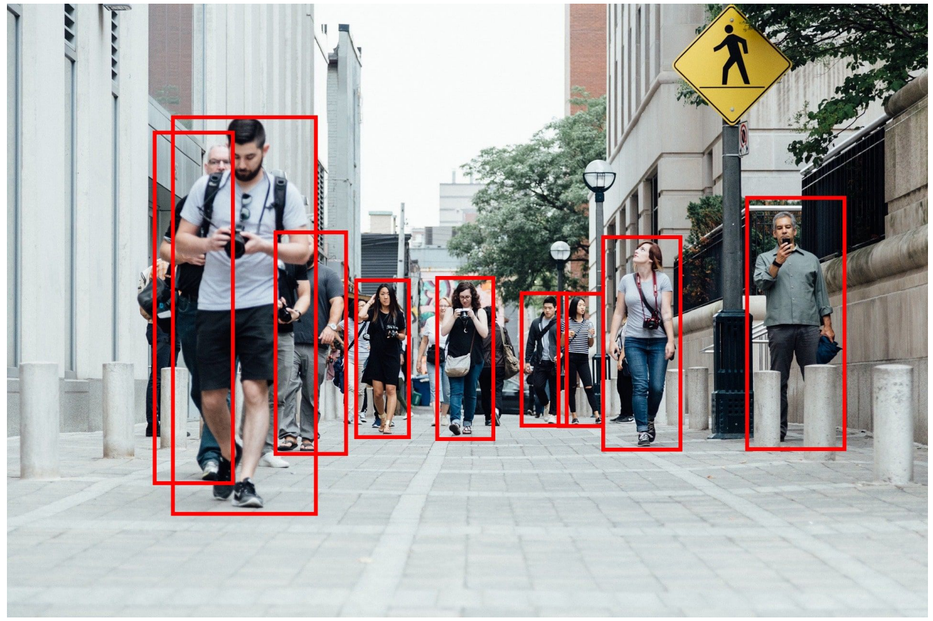
Image source: https://sidereal.medium.com/all-you-need-to-know-about-computer-vision-3997bc6318a6

Section 1

Other examples



Classification: Land Cover Classification Dataset with classes like Forest area, River, Highway etc ..



Object Detection: People Detection

Image source: https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset, https://visailabs.com/technical-metrics-required-to-asses-of-people-detection-solution/

Section 1

Other examples ..

Segmentation: Lungs Segmentation

Instance Segmentation: cars, road, pedestrian, cyclists etc ..

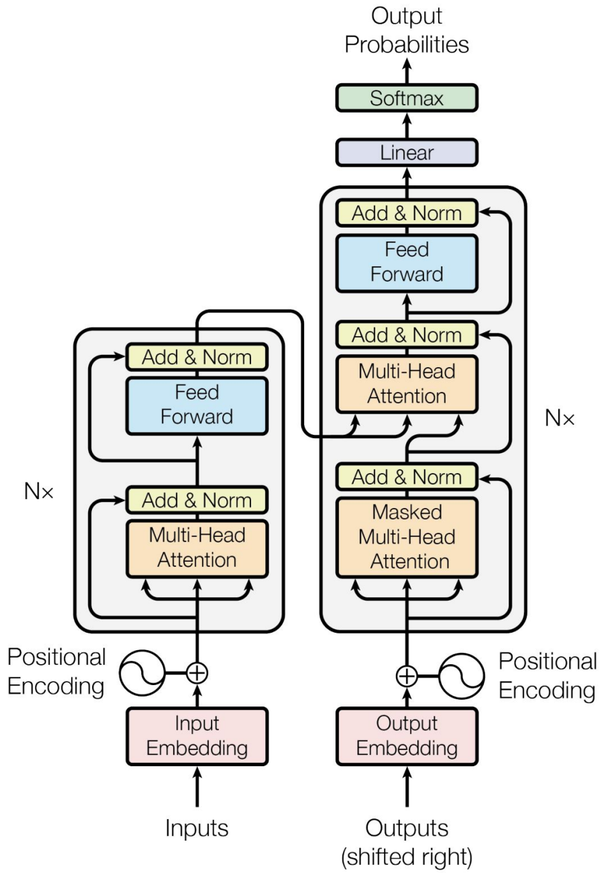
Image source: https://www.sciencedirect.com/science/article/abs/pii/S0169260719303517, https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/

Section 02

Evolution of Transformer models

Section 3

Attention is all you need !

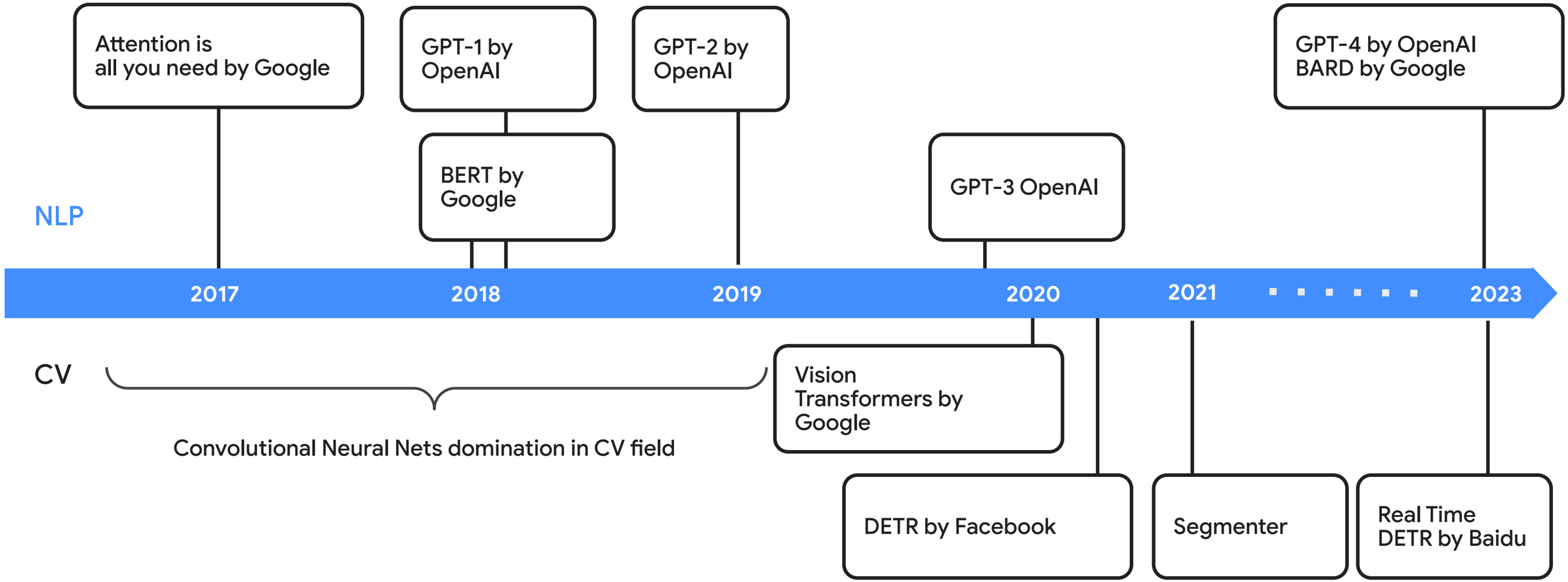


[The Transformer model architecture]

- Outperformed previous state-of-the-art models on a variety of natural language processing tasks
- Landmark paper in the field of natural language processing(NLP) by Vaswani et.al

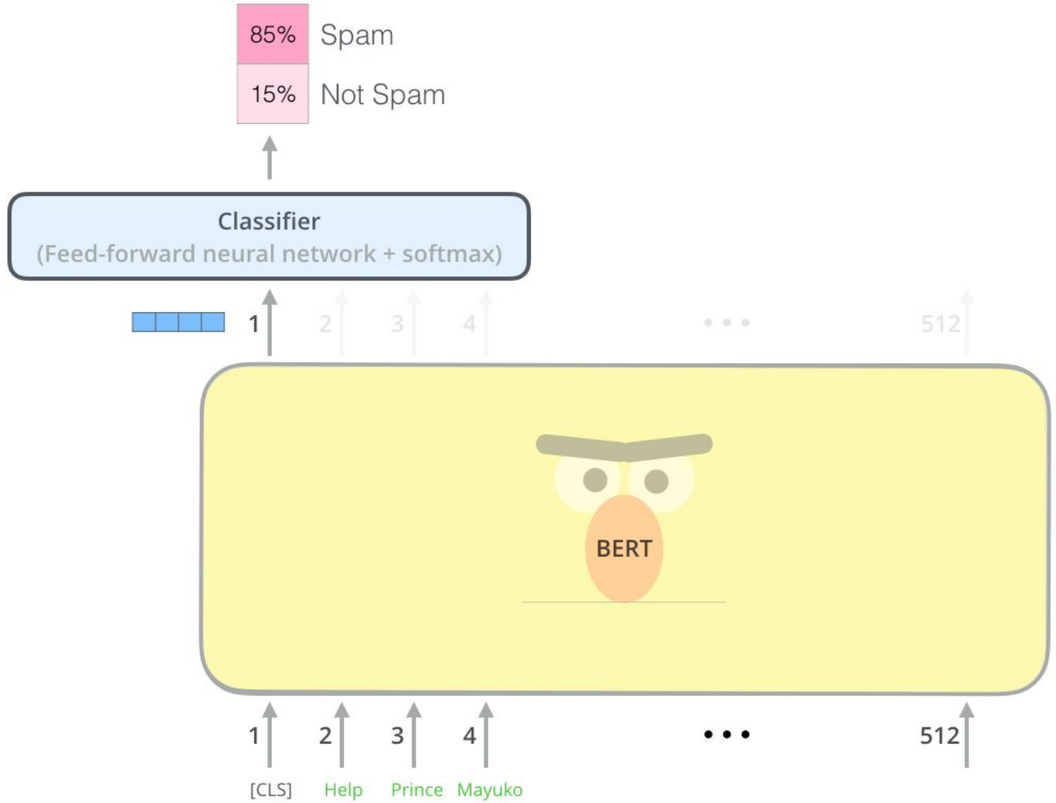
Section 3

Timeline of different transformer based models



Section 3

BERT: Bidirectional Encoder Representations from Transformer



BERT

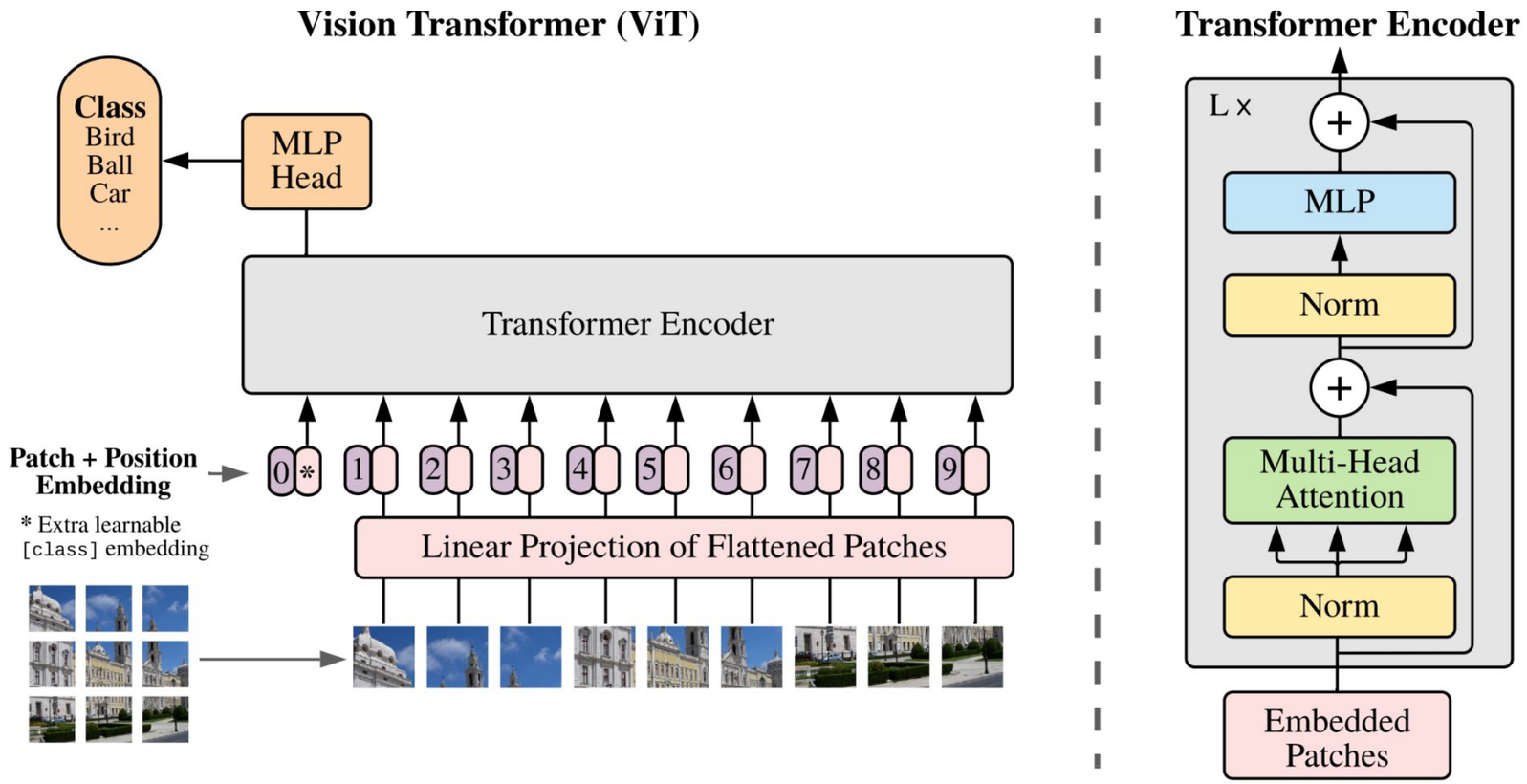
Section 03

Vision Transformer(ViT)

Details of ViT model architecture

Section 3

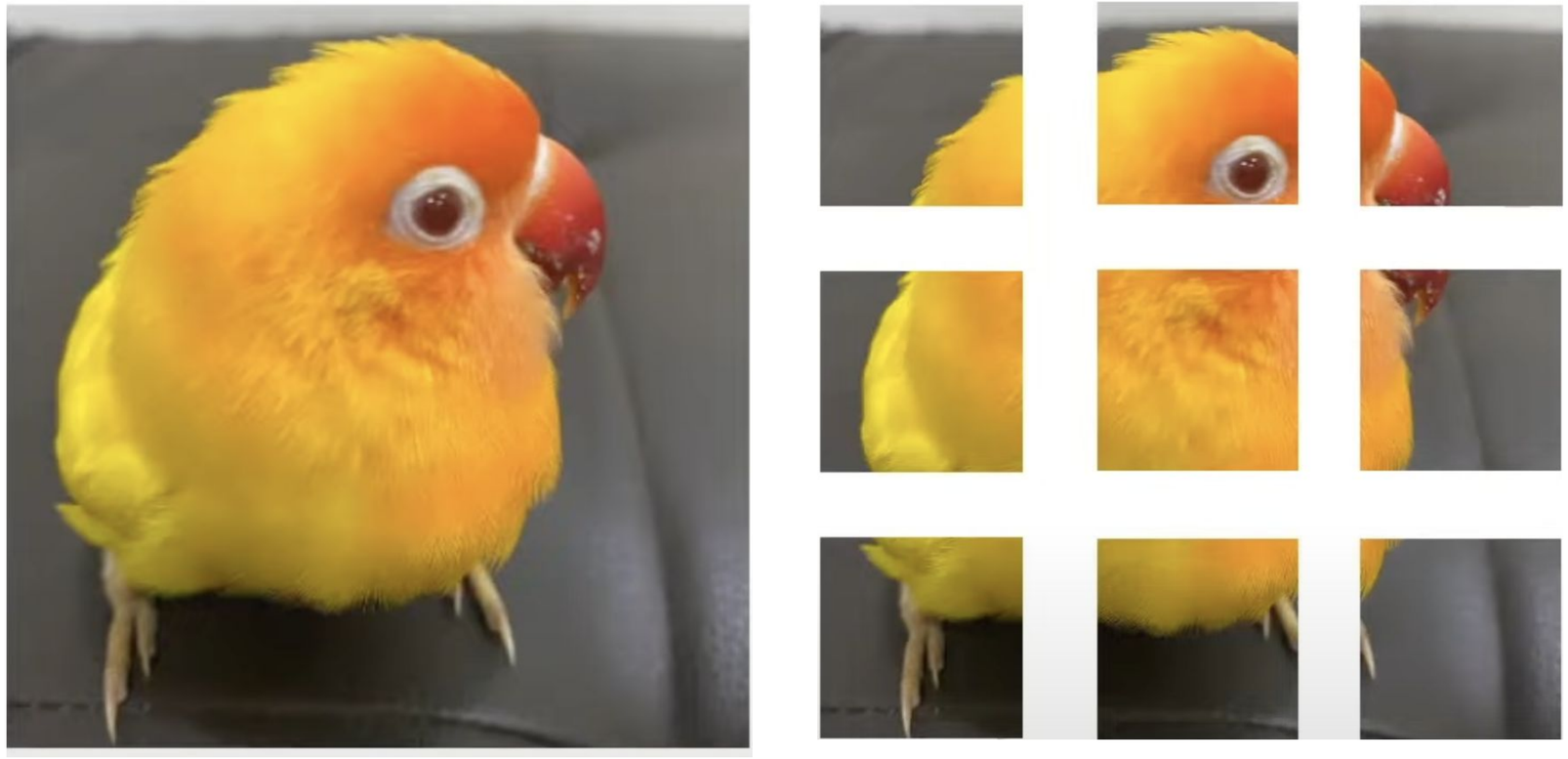
ViT Architecture



Ref: [An image is worth 16 x 16 words: Transformers for Image Recognition at Scale]

Section 3

Image to Patches



Number of patches
= W * H / (P * P)

where
W : width of the image
H : height of the image
P : Patch size

Ref: Vision Transformers and its applications

Section 3

Patch embeddings

$z(i) \in R^{1 \times D}$
where **i** from **1 to 9**

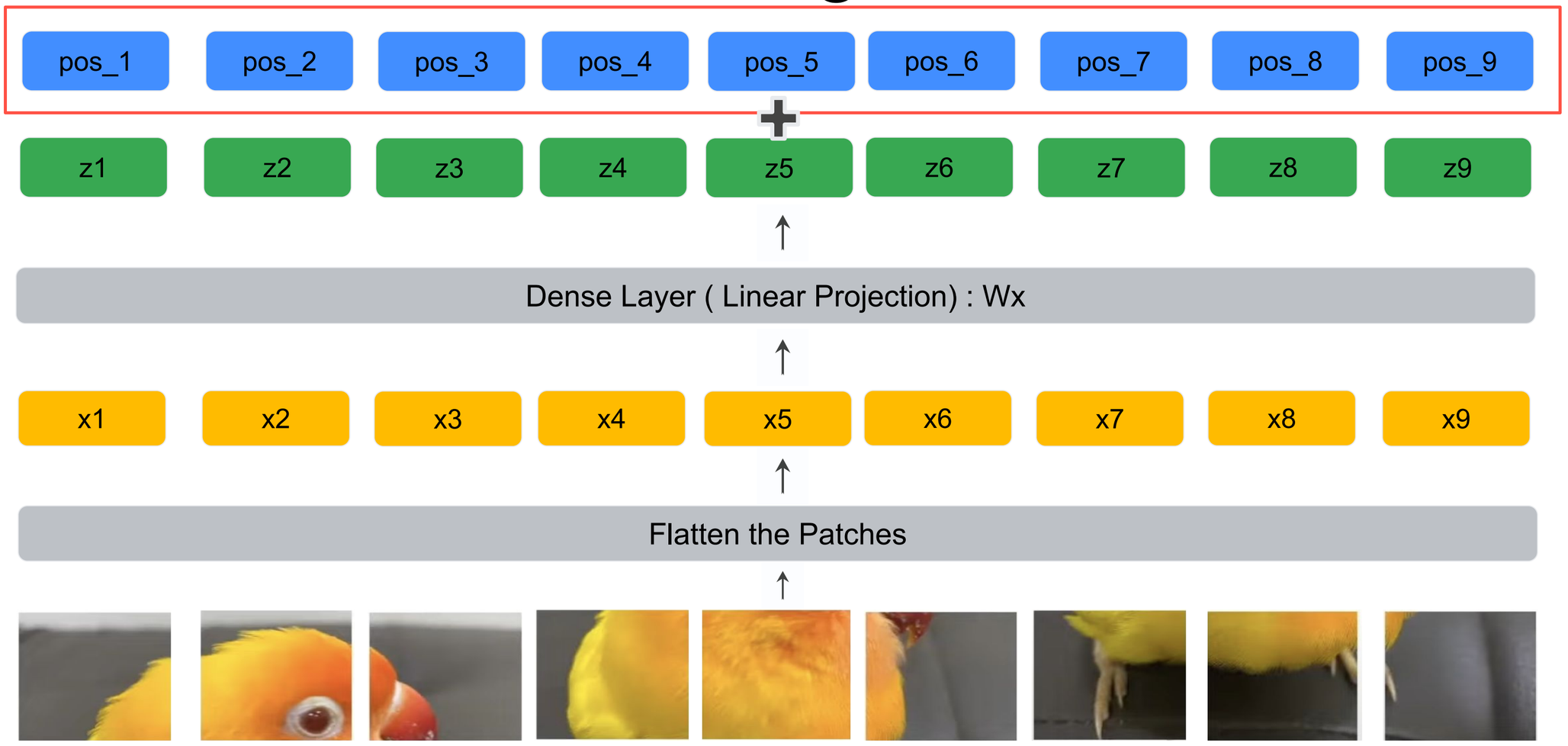
$W \in R^{(P*P*C) \times D}$

$x(i) \in R^{1 \times (P * P * C)}$
where **i** from **1 to 9**

Image Patches of size P * P * C

Section 3

Positional embeddings



pos_(i) Є $R^{1 \times D}$
where **i** from **1 to 9**

z(i) Є $R^{1 \times D}$
where **i** from **1 to 9**

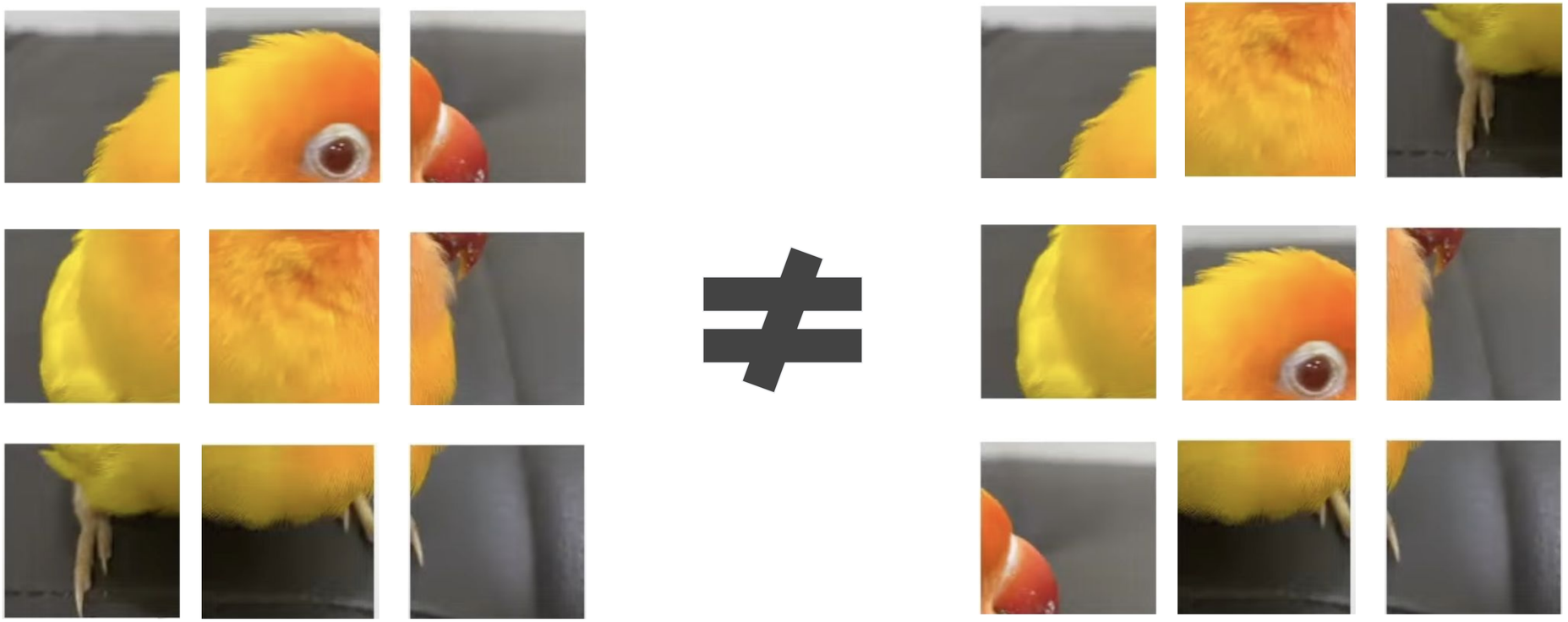
W Є $R^{(P*P*C) \times D}$

x(i) Є $R^{1 \times (P * P * C)}$
where **i** from **1 to 9**

Image Patches of size P * P * C

Section 3

Why Positional Embeddings ?

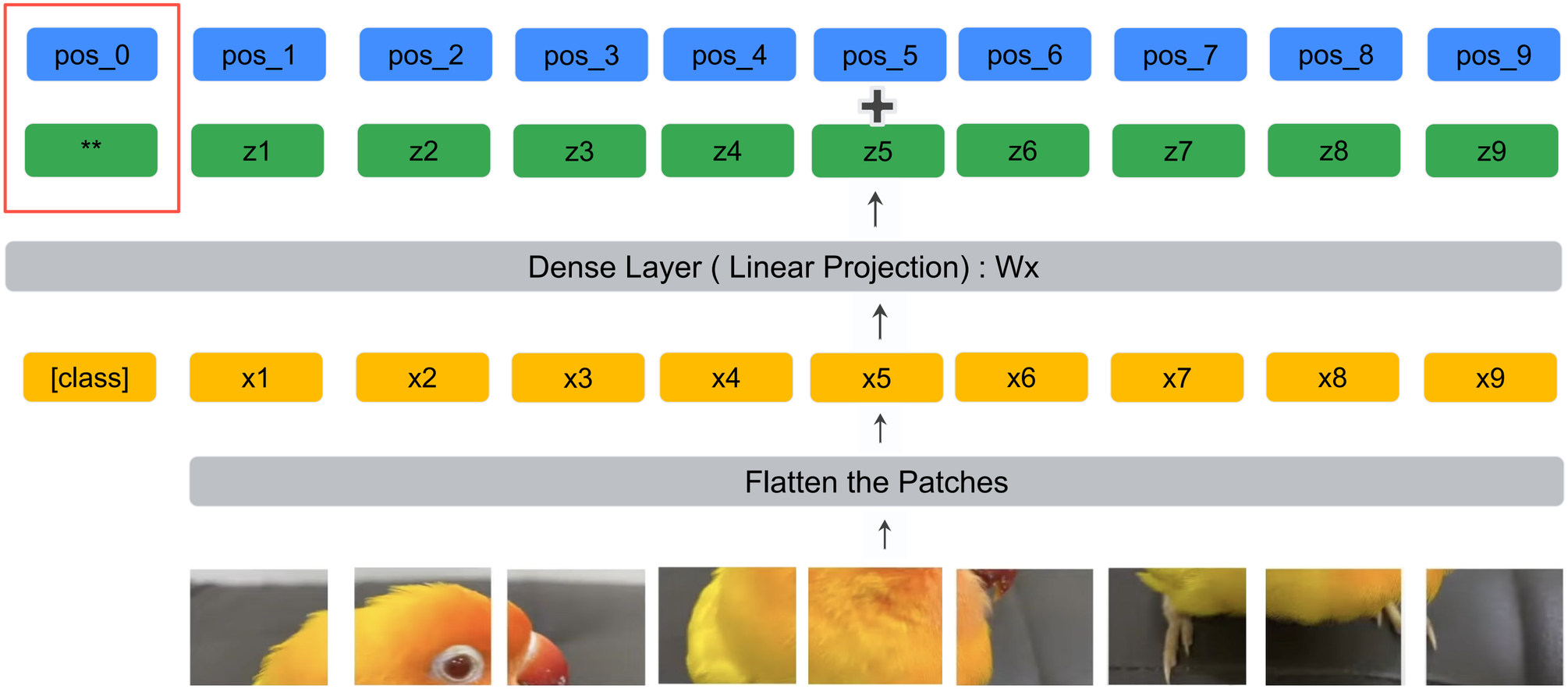


Allows the model to learn the relative position of each patch in the image

3% drop in accuracy observed without positional embeddings

Section 3

[class] embedding



pos_(i) Є $R^{1 \times D}$
where **i** from **1 to 9**

z(i) Є $R^{1 \times D}$
where **i** from **1 to 9**

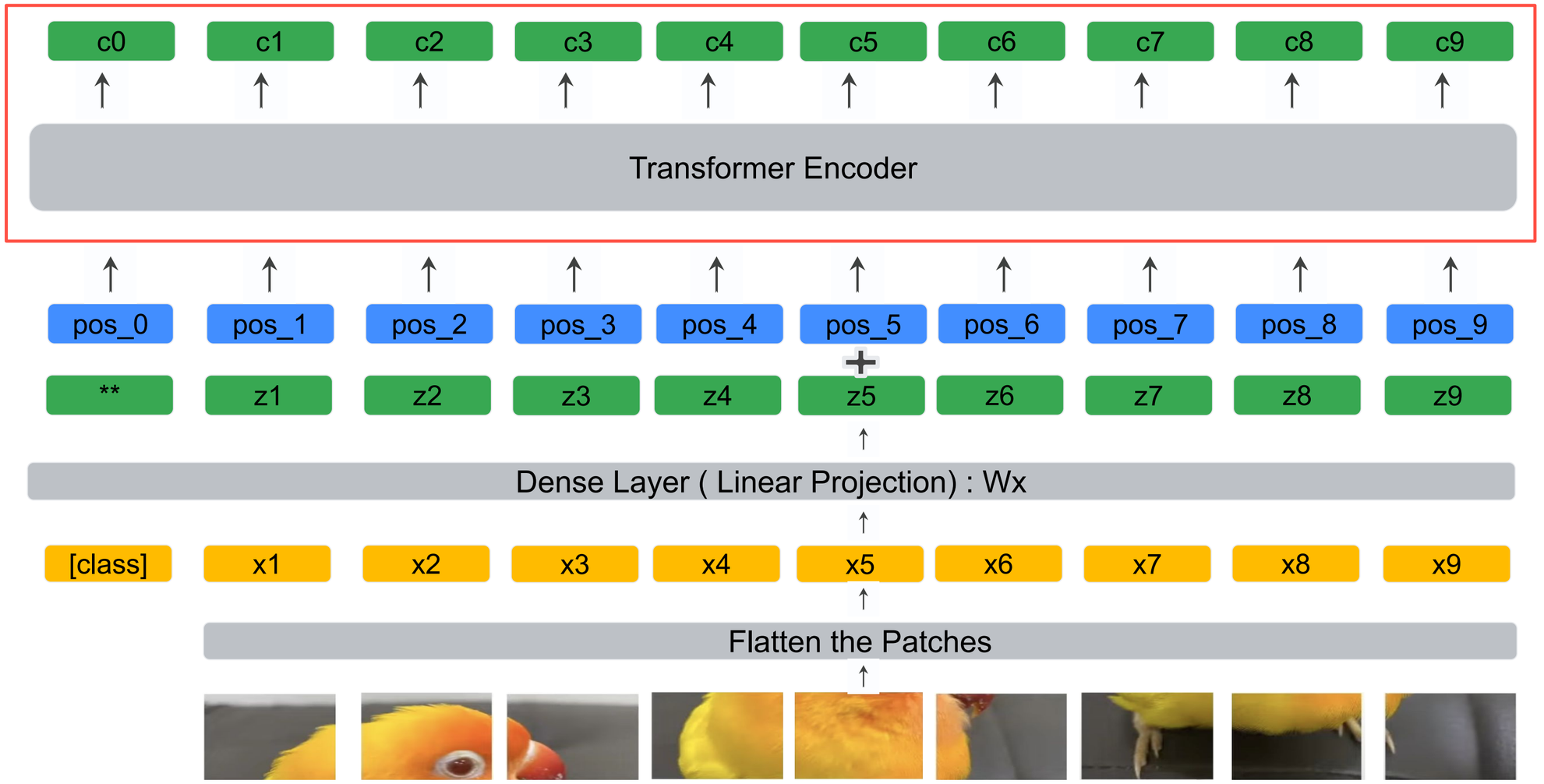
W Є $R^{(P*P*C) \times D}$

x(i) Є $R^{1 \times (P * P * C)}$
where **i** from **1 to 9**

Image Patches of size P * P * C

Section 3

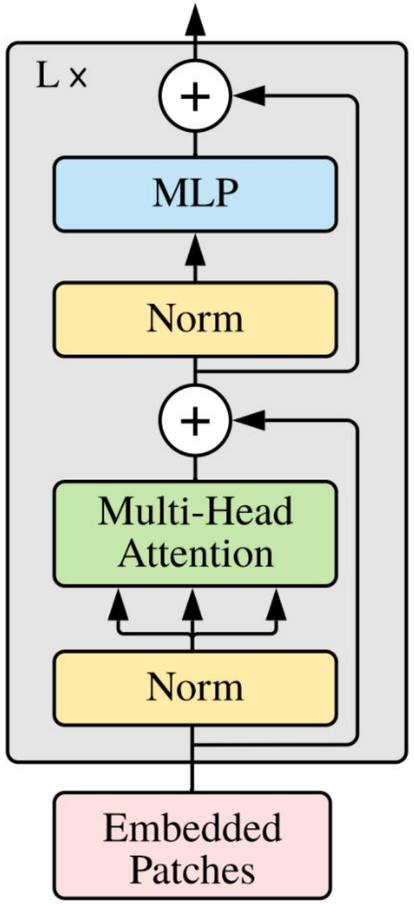
Transformer Encoder



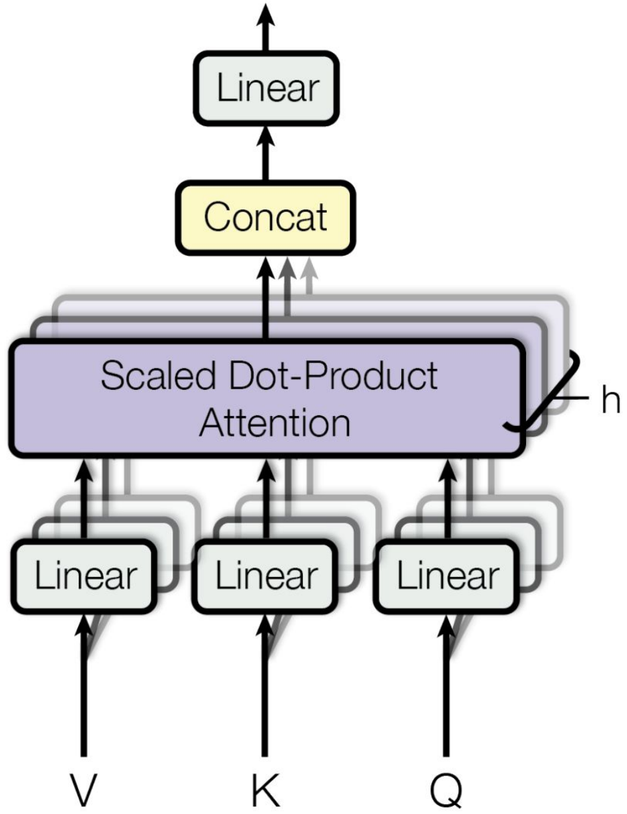
Contextual Embeddings

Section 3

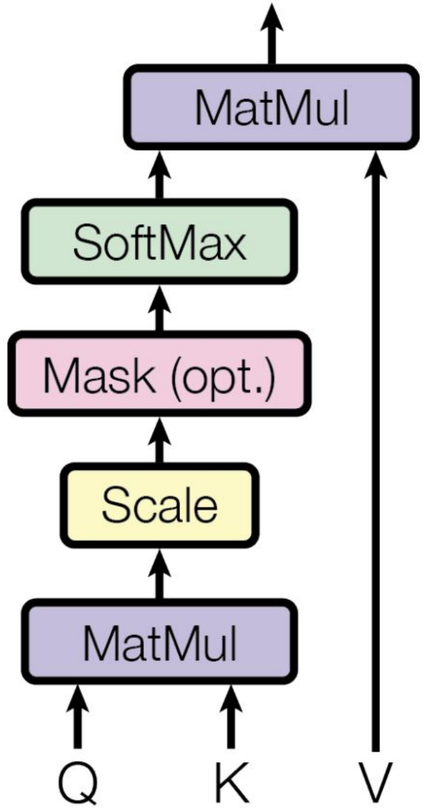
Transformer Encoder continued ..



Single Encoder Block



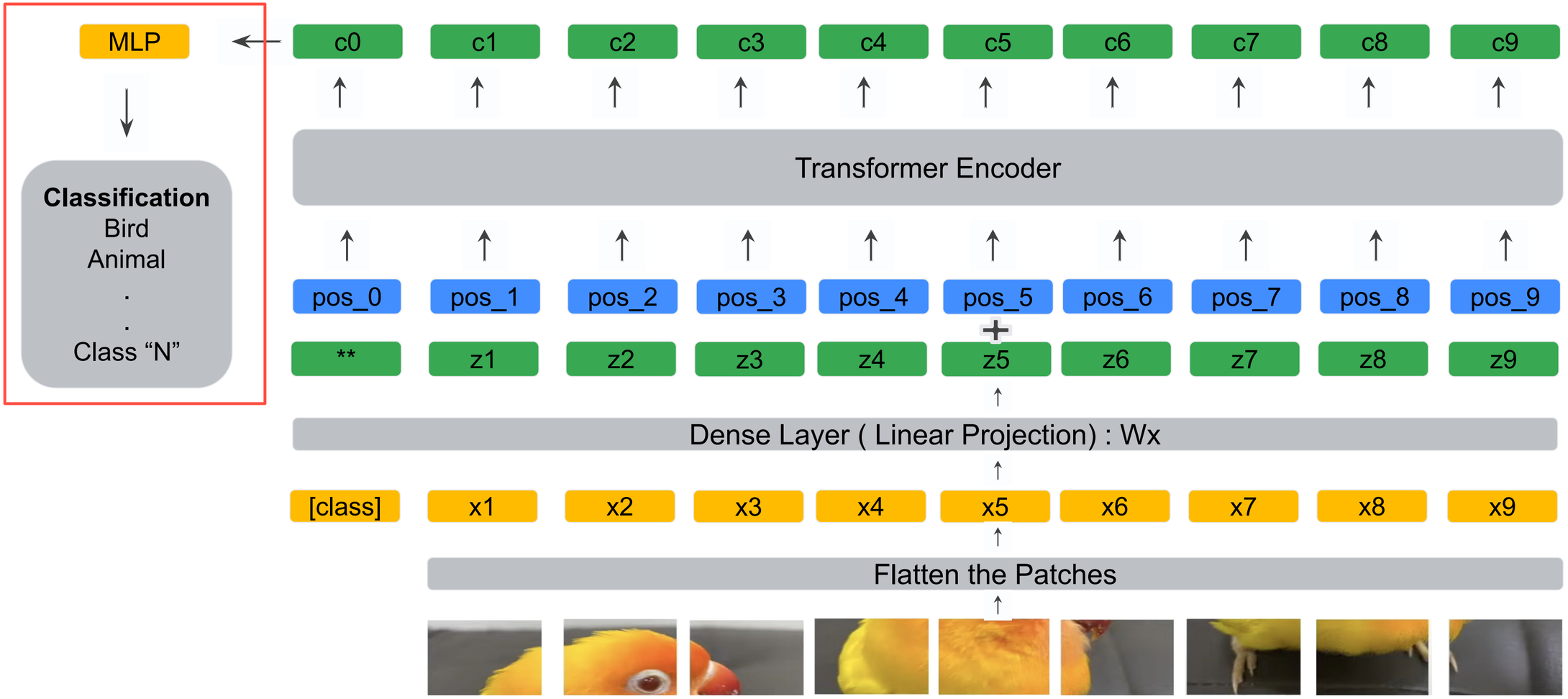
Multi head attention



Scaled Dot Product Attention

Section 3

ViT: MLP head and classification



Section 04

Keras Implementation for Vision Transformer(ViT)

https://github.com/rajesh-bhat/vision_transformers_googleio_extended

Transformer Encoder Keras NLP: code snippet

```
keras_nlp.layers.TransformerEncoder(
    intermediate_dim,
    num_heads,
    dropout=0,
    activation="relu",
    layer_norm_epsilon=1e-05,
    kernel_initializer="glorot_uniform",
    bias_initializer="zeros",
    normalize_first=False,
    name=None,
    **kwargs
)
```

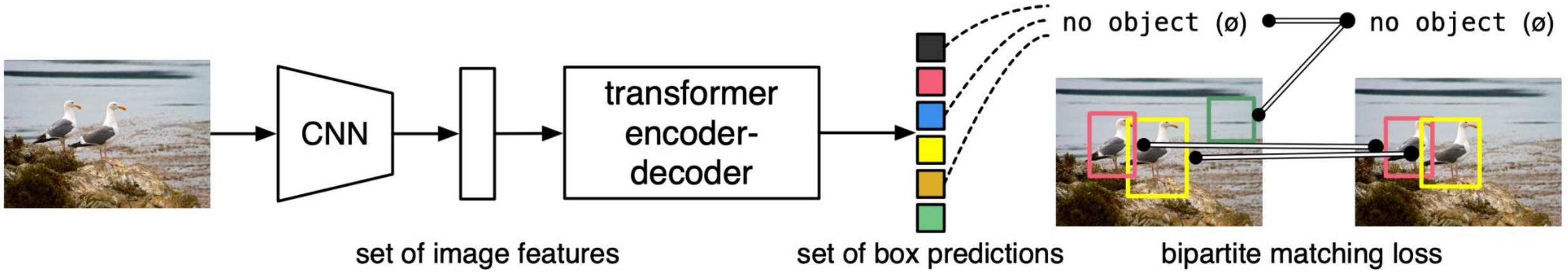
Section 05

Transformers for other Vision tasks

Object Detection, segmentation ..

Section 5

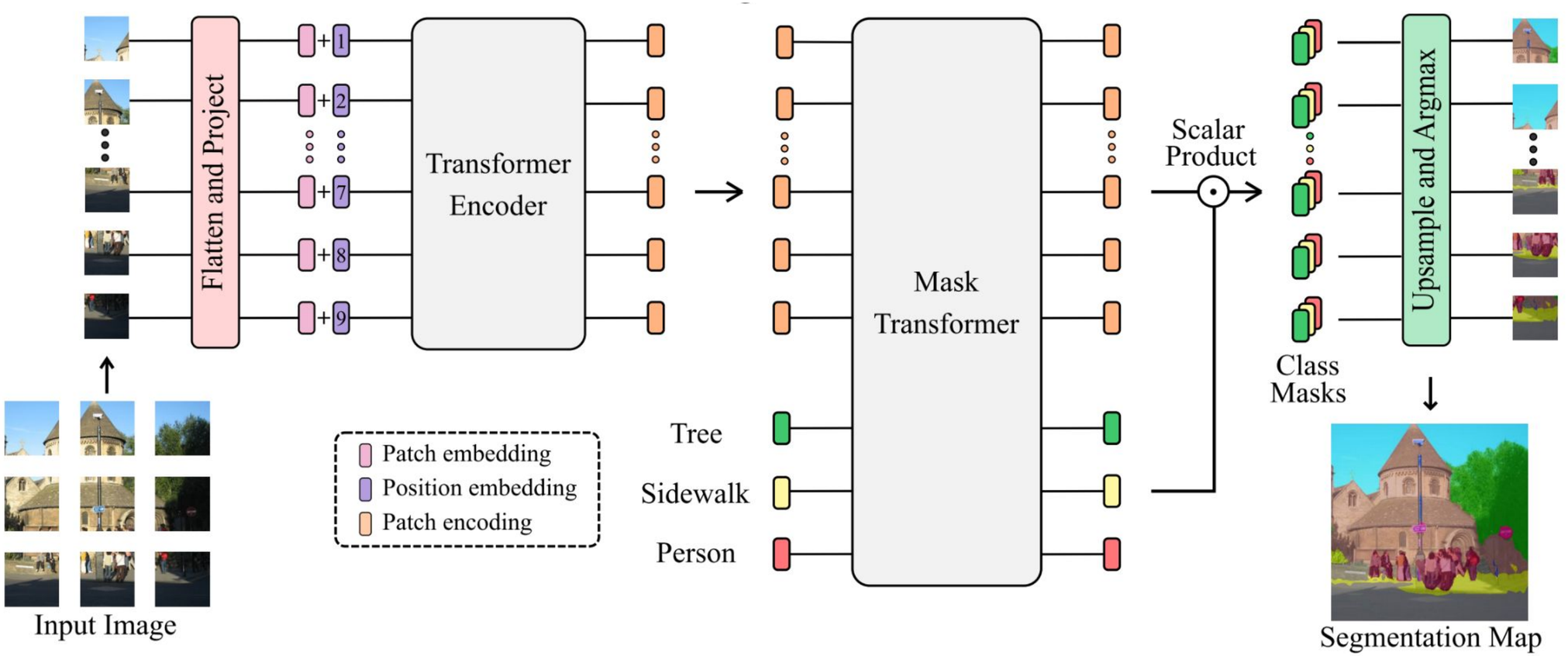
DETR: End to End Object detection with Transformers



DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture

Section 5

Segmenter



Overview of Segmenter model

Summary

- Intro to Computer Vision and various tasks
- Evolution of Transformer models
- Vision Transformer(ViT) architecture in detail
  - Images to Patches
  - Patch Embeddings
  - Positional Embeddings
  - Transformer Encoder
  - MLP head and classification
- Vision Transformers in Keras
- Transformers for Object Detection(DETR) and Segmentation tasks(Segmenter)

Google I/O Extended

Thank You

Rajesh Shreedhar Bhat

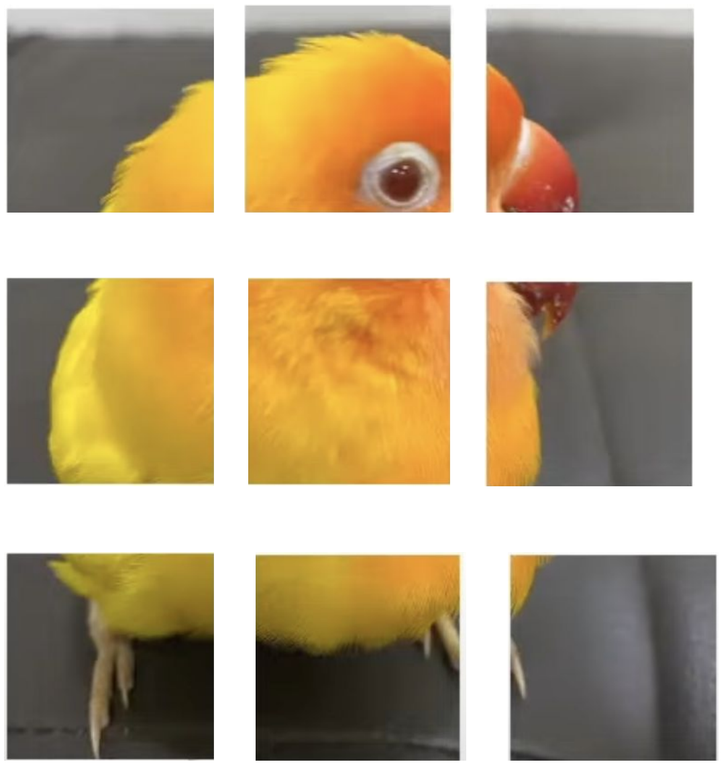
Sr. Data Scientist, Walmart & GDE ML

Section 06

Appendix

Section #

Patch slide

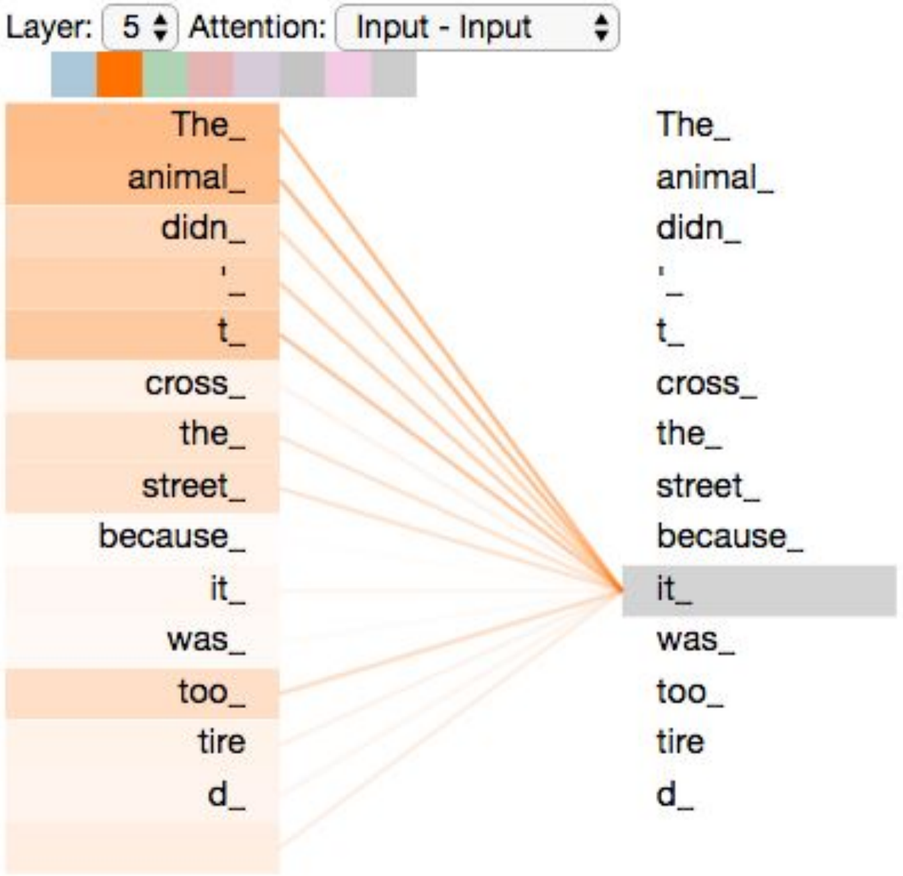


Ref: An image is worth 16 x 16 words: Transformers for Image Recognition at Scale

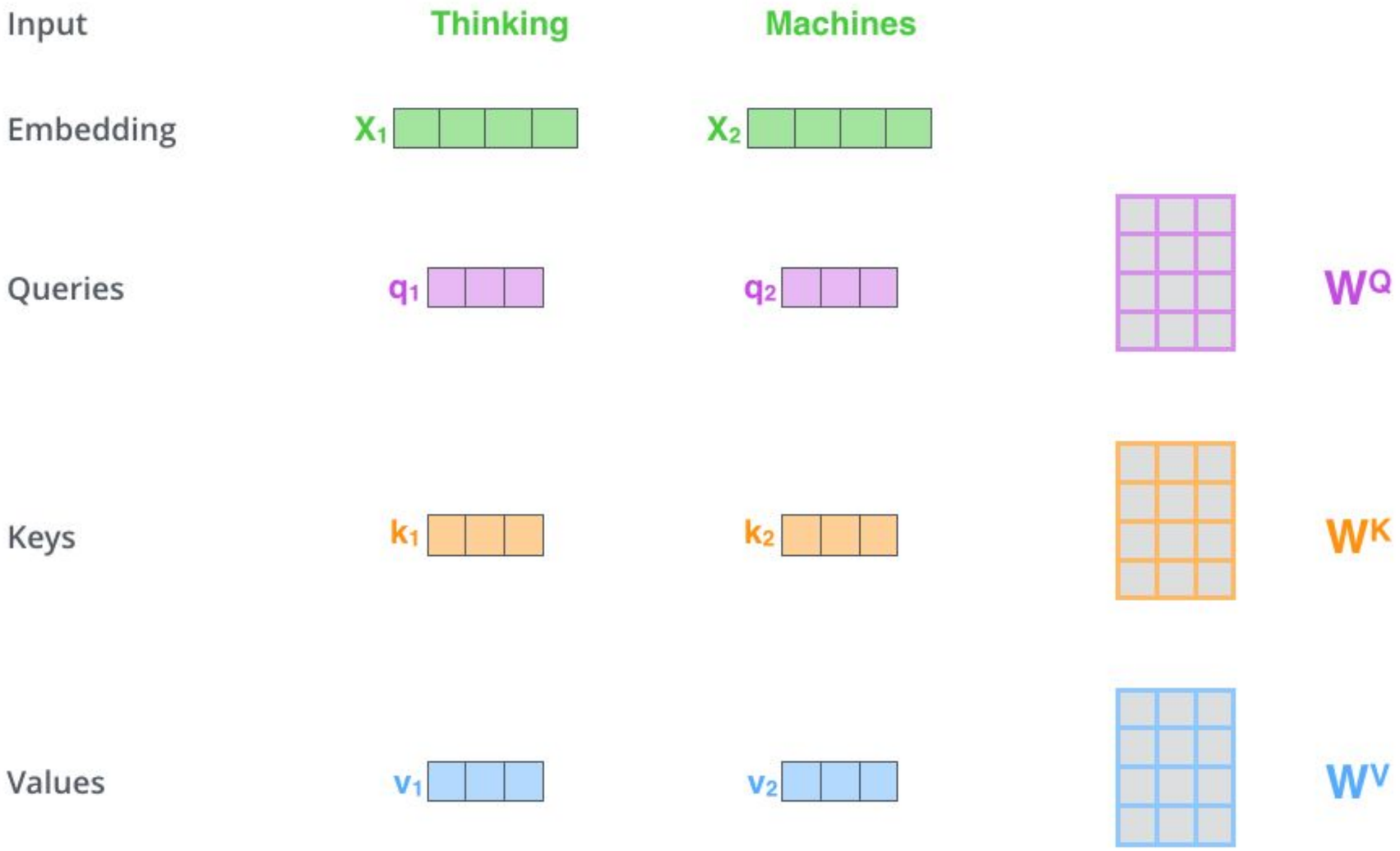
Self Attention

Translate : “The animal didn't cross the street because it was too tired”.

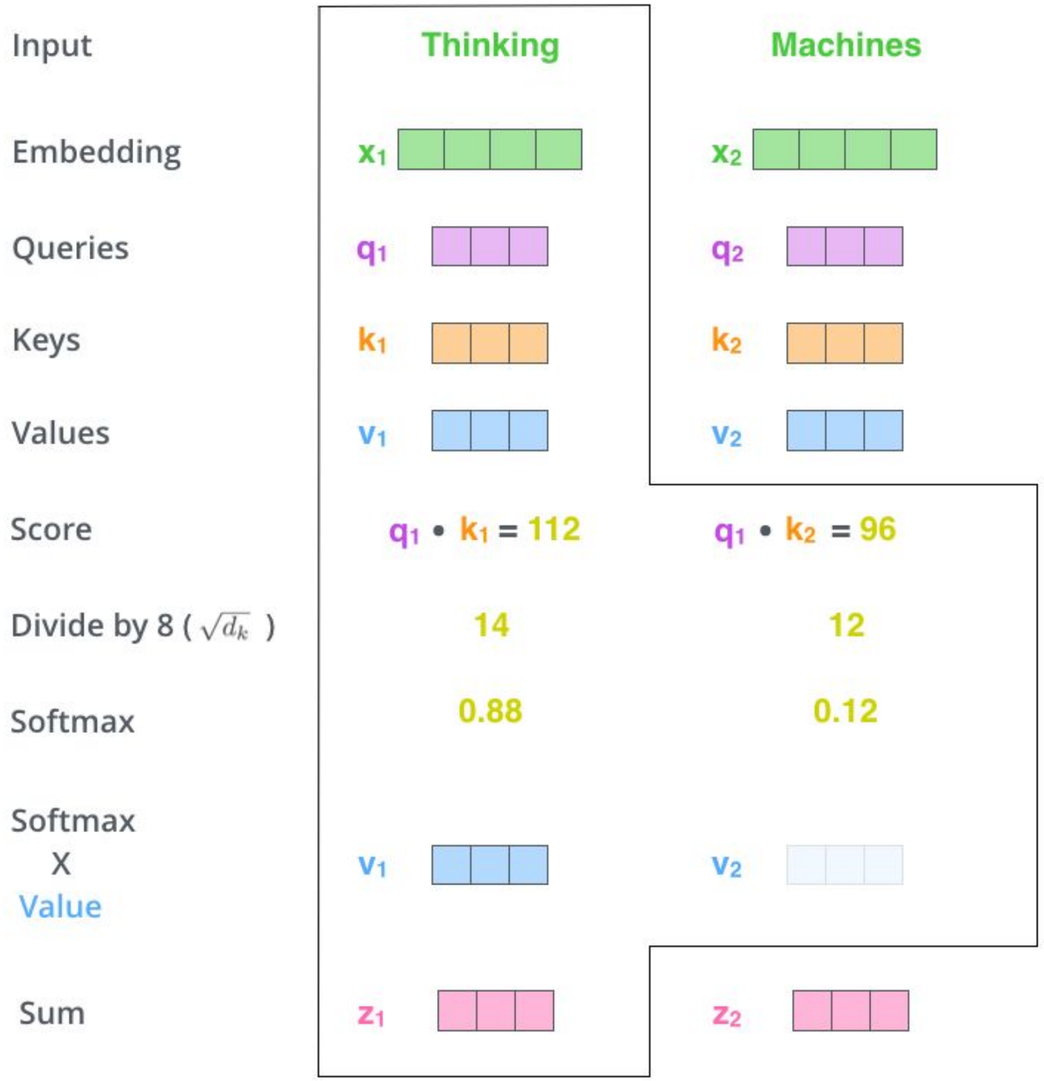
What does “it” in the above sentence refer to? Animal or Street?



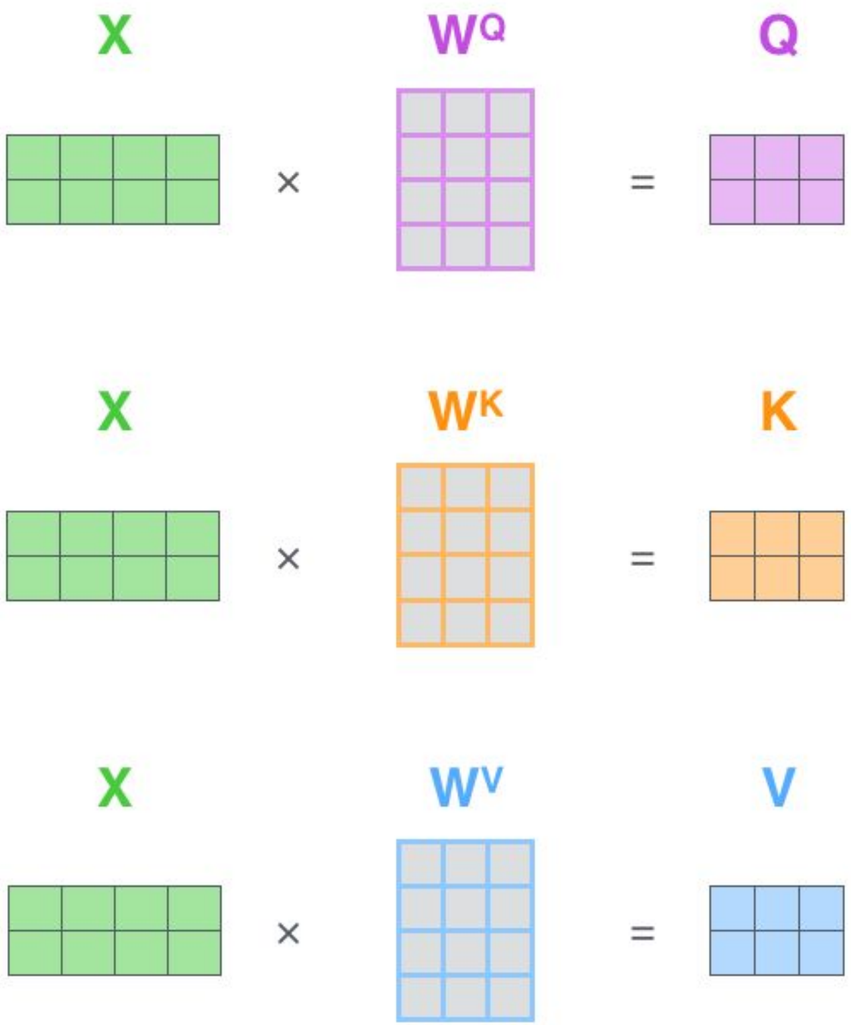
Self Attention in detail



Self-Attention: output



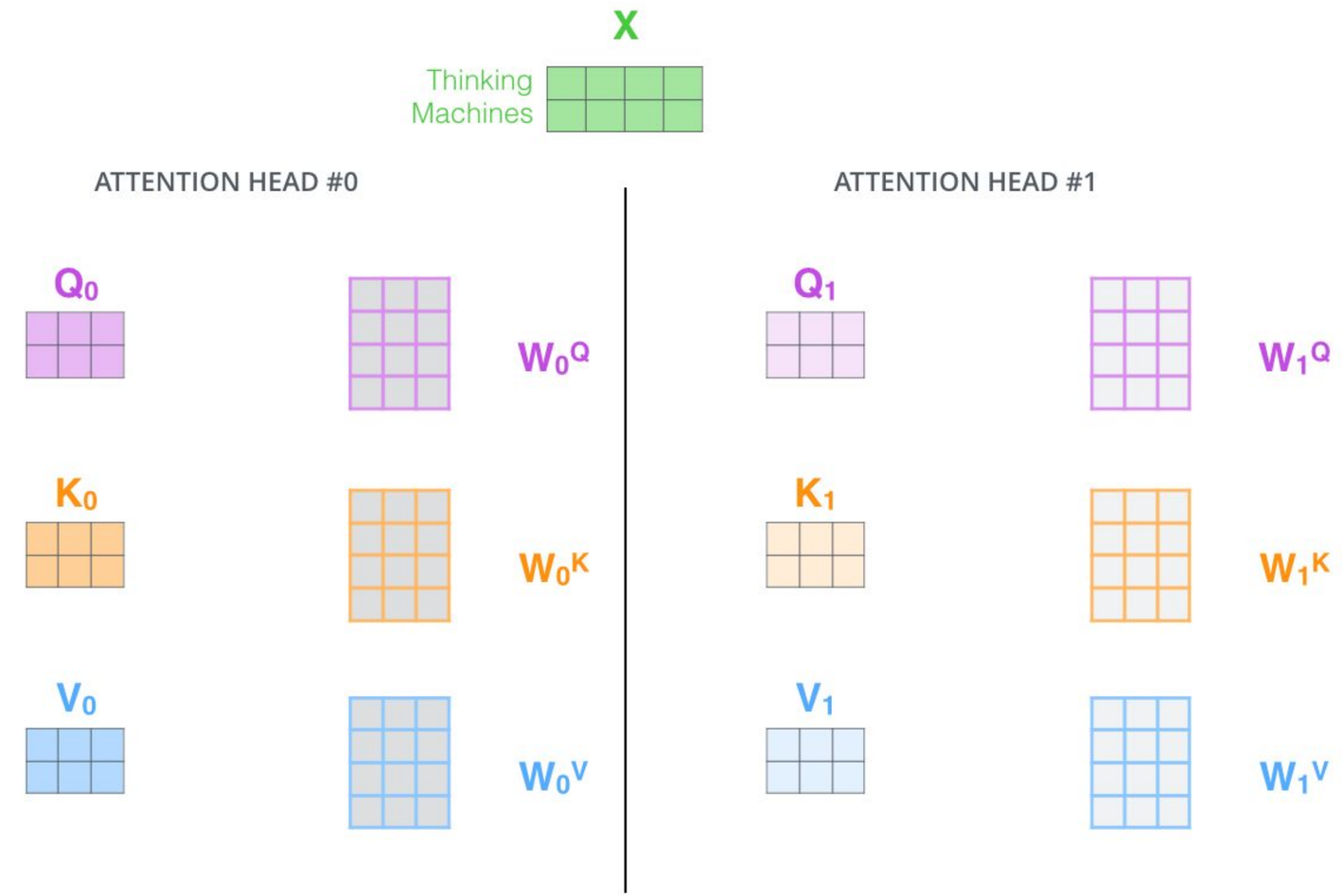
Matrix Calculation of Self-Attention



$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

Multi-head attention

- Model ability to focus on different positions.
- Input embeddings/ vectors from lower encoders/decoders into a different representation subspace.



Multi-head attention ..

1) This is our input sentence*

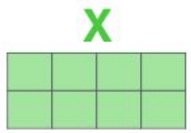
2) We embed each word*

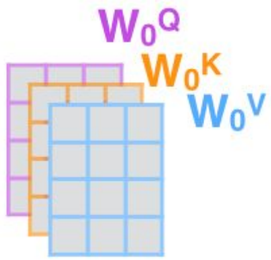
3) Split into 8 heads. We multiply X or R with weight matrices

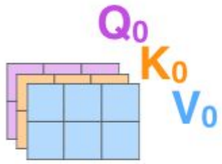
4) Calculate attention using the resulting Q/K/V matrices

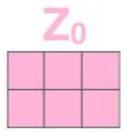
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

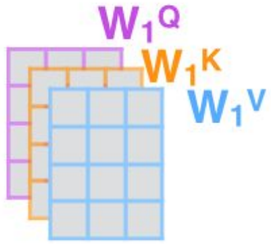
Thinking
Machines

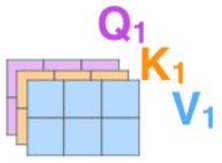


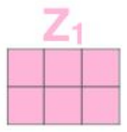




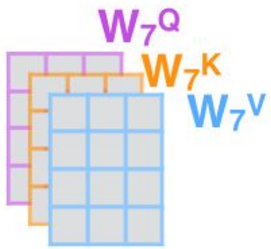


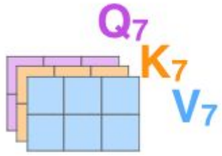


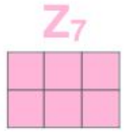


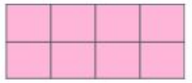


* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one





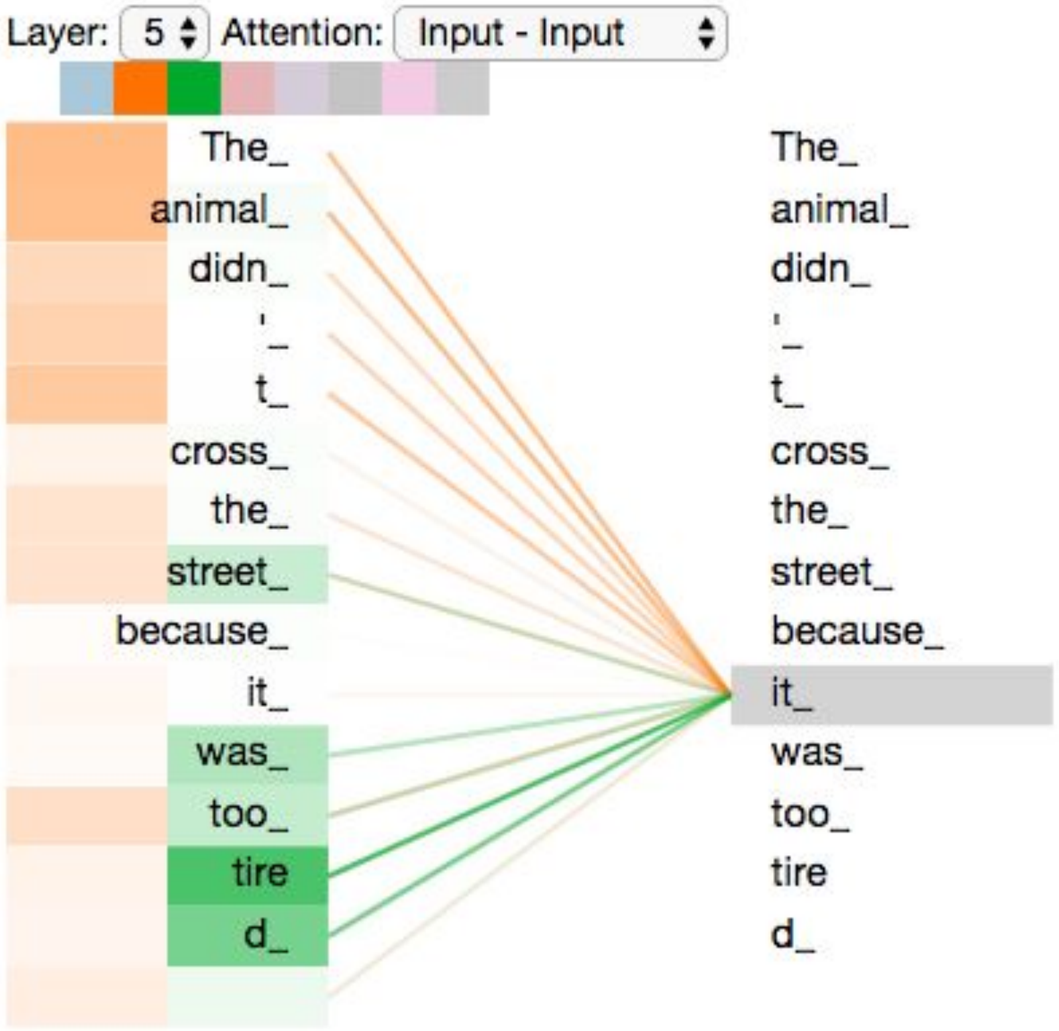




Interpreting - Multi Head Attention

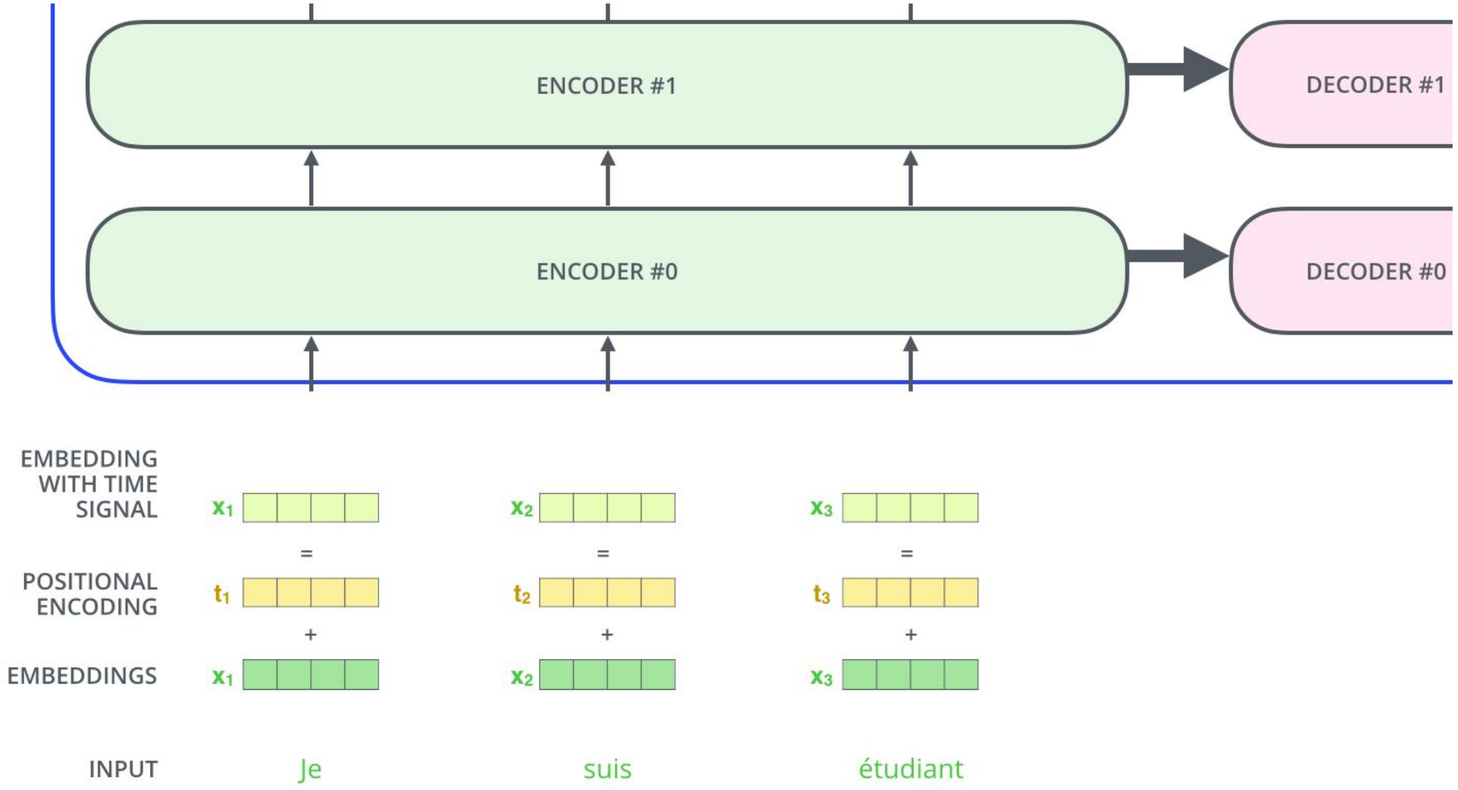
- As we encode the word "it", one attention head is focusing most on "the animal", while another is focusing on "tired".

- In a sense, the model's representation of the word "it" bakes in some of the representation of both "animal" and "tired".



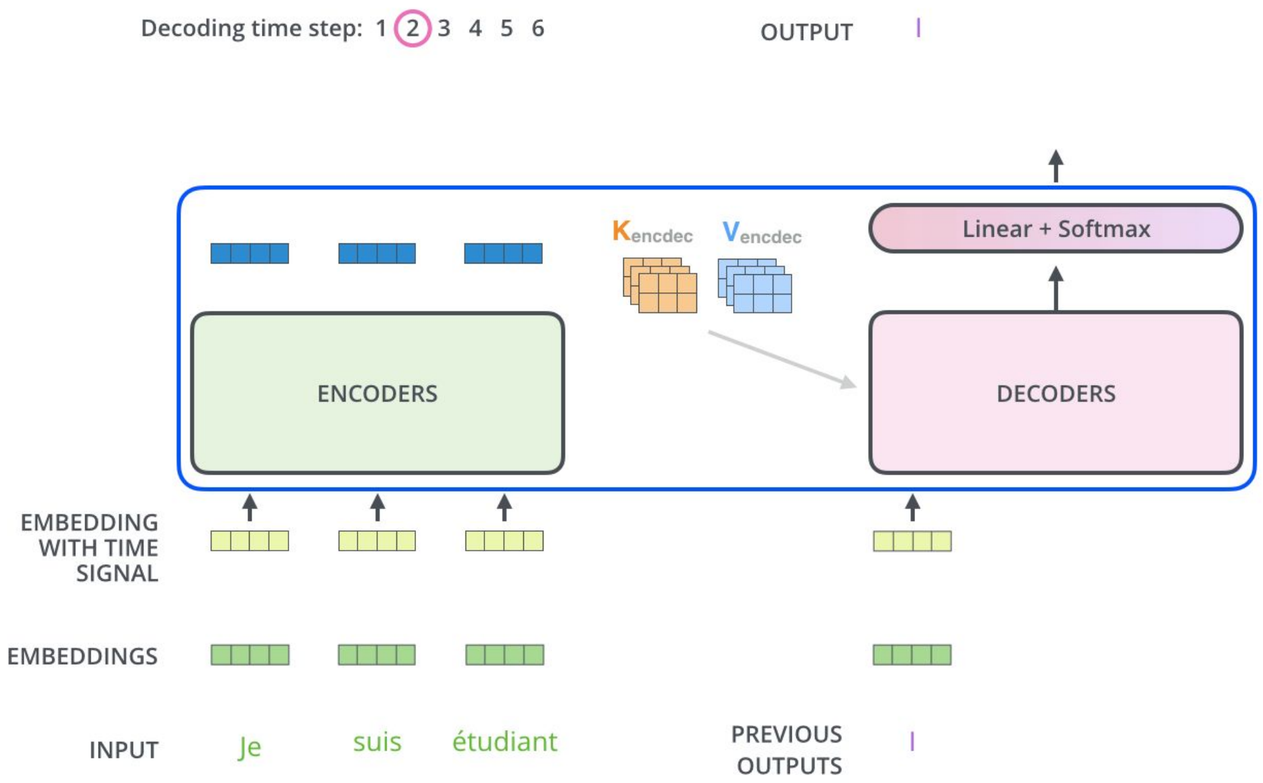
Positional Embeddings

- So far, order of the words in the input sequence is missing.
- **Positional embeddings:** vectors which follow a specific pattern that the model learns, which helps it determine the position of each word, or the distance between different words in the sequence.



Transformer Architecture

The decoder attends on the encoder's output and its own input (self-attention) to predict the next word.



BERT: Bidirectional Encoder Representations from Transformers