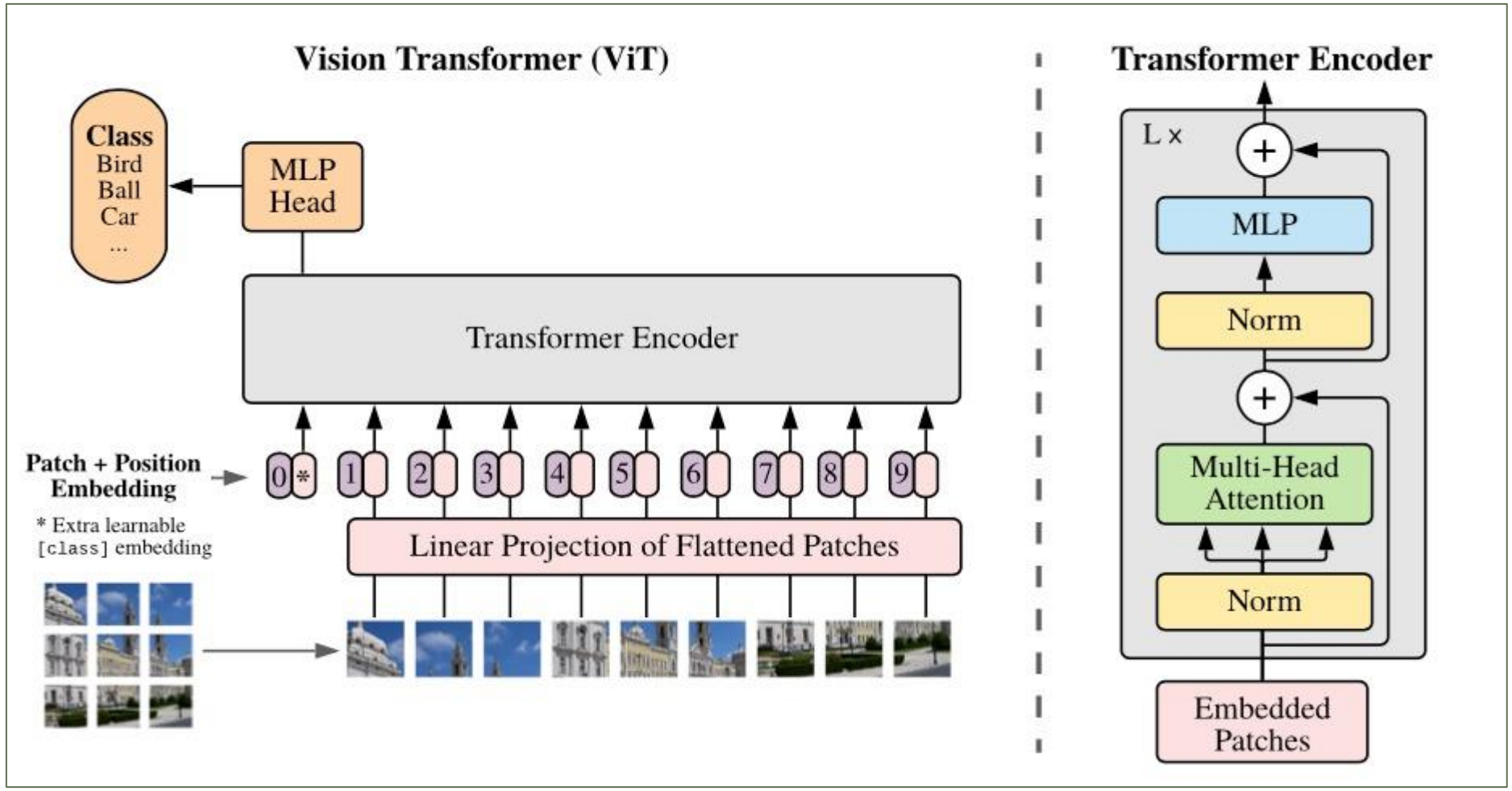
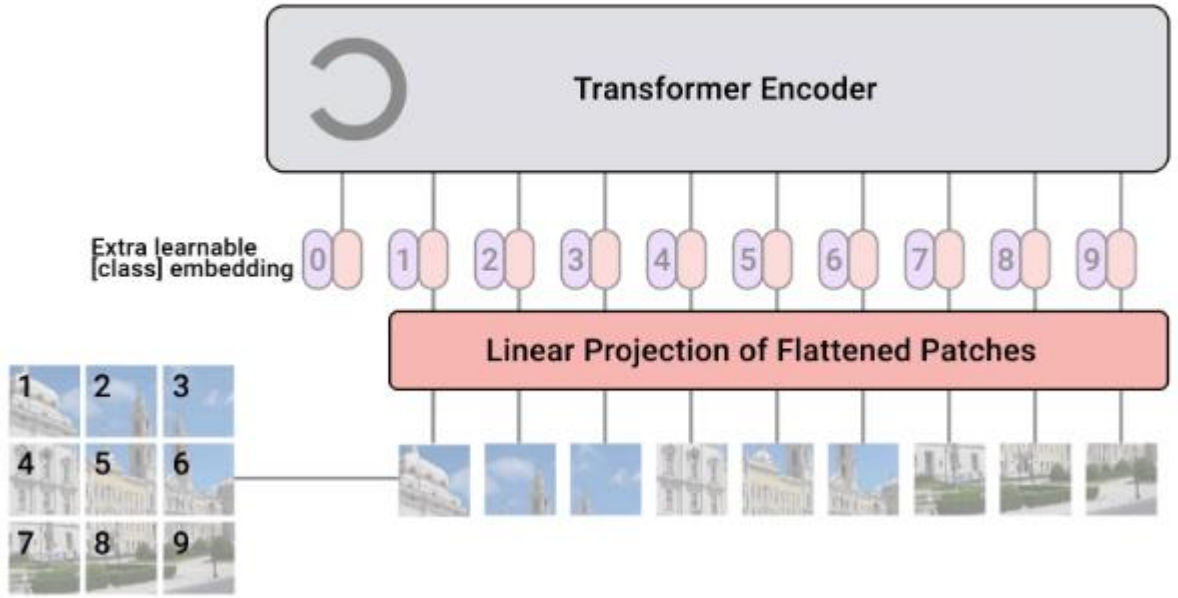
Vision Transfomer

MUSTAFA HAJIJ

Model Architecture



Vision Transformer Pseudo-Code



tugot17/Vision-Transformer-Presentation: Presentation on An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (github.com)

Vision Transformer Pseudo-Code

Self-attention layer

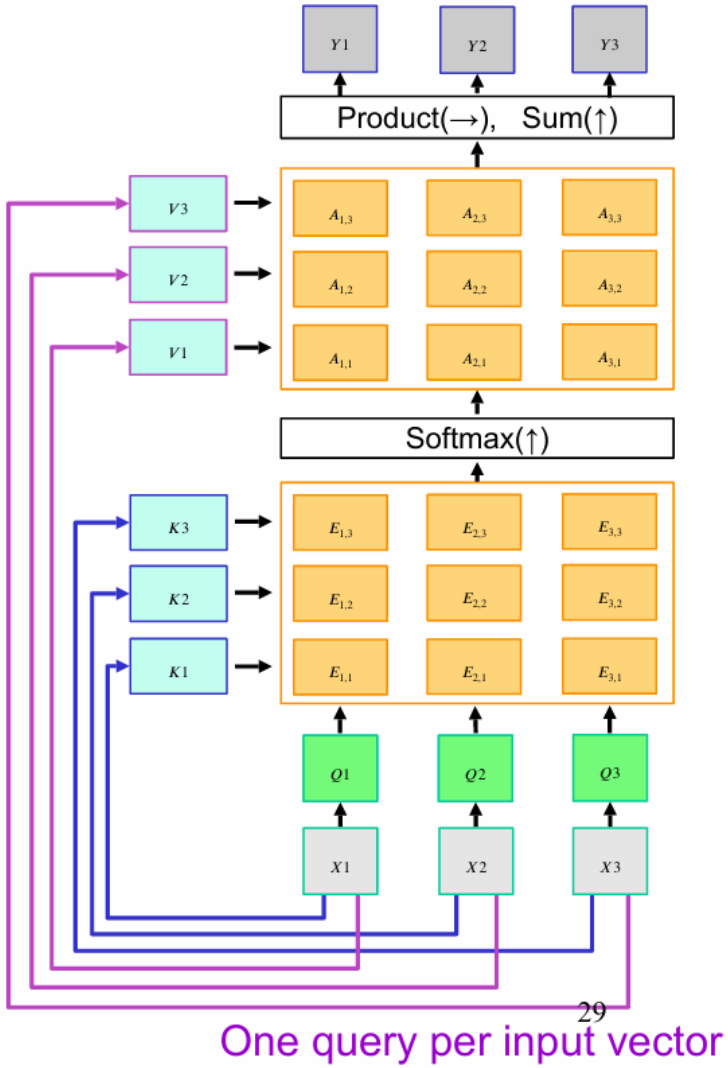
- Query vectors: $Q = XW_Q$
- Key vectors: $K = XW_K$
- Value vectors: $V = XW_V$
- Similarities: *scaled dot-product attention*

$$E_{i,j} = \frac{(Q_i \cdot Kj)}{\sqrt{D}} \text{ or } E = QK^T/\sqrt{D}$$

(D is the dimensionality of the keys)

- Attn. weights: $A = \text{softmax}(E, \text{dim} = 1)$
- Output vectors:

$$Y_i = \sum_j A_{i,j} V_j \text{ or } Y = AV$$



One query per input vector

Adapted from J. Johnson and S. Lazebnik.

Vision Transformer Pseudo-Code

```
def ViT (input):
    patches = Create_Patches(input)
    patch_embed = Patch_Embedding(patches)
    sequence = Concat(class_token, patch_embed) + Position_embedding
    hidden_states = Transformer(sequence)
    class_output = Classification_Head(hidden_states[0])

    return class_output
```

This is the pseudo code for the sequence of operations on a image to classify it using the Vision Transformer Model