By google.

VISION TRANSFORMER. ← (introduced in a Research paper in 2020 "An image worth 16x16 words : Transformers for image recog at scale.)

A deep learning model designed to process images using the self-attention mechanism; originally developed for 'Natural language processing'.

Unlike CNNs which rely on receptive fields, ViTs model global relationship b/w all pixels of an image.

Working of a ViT:-

Instead of convolutions, ViT process image by.

(i) Dividing an image into patches (e.g 16x16 pixels)
(ii) Flattening the patches into 1D vectors.
(iii) Encoding positional information to retain spatial struct.
(iv) Feeding the encoded patches into standard Transformer Model (like in NLP).
(v) Using a classification head for classifing/final predictions.

Important Vit Specific Steps:-

(A) Patch Embedding Since transformers work with sequential data, we need to convert the 2D images into a sequence of 1D patch embeddings.

step1
↳ spitting the input image (HxWxC) into N fixed patches

step2
↳ Flattening each patch into 1D vector.

step3
↳ Projecting each 1D vector into a D-dimensional embedding space using a learnable linear projection.
↳ (fully connected layer).

* Thus the image is transformed into a sequence of N patch embeddings

(B) Positional Encoding.

Transformers donot have built in understanding of spatial relationship. To preserve the order and structure of patches we add a positional encoding to each patch embedding (Similar to generic structure of Transformer).

(C) Transformer Encoder

Patches are passed here to be processed in parallel.

This block of code/Architecture is similar to generic Transformer architecture, consisting of the following layers

a. Multi head Self Attention (MSA).

b. Layer Normalization.

c. Feed Forward Network.

d. Skip Connection / Residual Connections.

(D) CLASS TOKEN

(questionable statement btw) usually.
a unique component that is not present in the generic Transformer.

It is added as an input/prepended to patch embedding b/f ↑ encoder la

Now our image is basically divided into different/numerous patches but we still need one final output that represents the entire image.

↳ The class token solves this problem as it acts as a special placeholder that collects information from all patches; serving as the global representation of the image.

How generic Transformer like BERT had this idea?
In NLP Transformers Like BERT there was a similar token used called as the '[CLS]' which basically used to contain the summary of the whole sentence, and then used for tasks.

Example: The weather is nice today [SEP]
↓
[CLS]

- Sentiment analysis (e.g positive or negative).
- sentence classification. (News or sports).

ViT borrowed the same idea and introduced a (x_cls) class token that serves the same purpose.

→ gathers information from all the patches through self attention.

→ at the end represents the entire image for classification

Working of Classtoken.

- starts off as a random no/vector
- interacts with patch embeddings
- collects and condenses info from all patches.
- after multiple Transformer layers it now represent the entire image.
- Instead of taking avg of each patch we use class token directly to interact with the image. classify

CNN does not use this technique rather relies upon GAP to summarize the image which is less flexible (not allowing to treat key regions imp)

Feature.	ClassToken (ViT)	Pooling (CNN)
Learnable?	Yes, learns representations.	No, just a fixed operation.
Global Context	Yes, attends to all patches.	Limited, only aggregates value.
Task-Specific.	Can adapt per dataset	fixed mathematical operation.

(E) MLP HEAD

- The final output of class token is passed through MLP head.
- It predicts the final class's label.

Transformer Architecture.