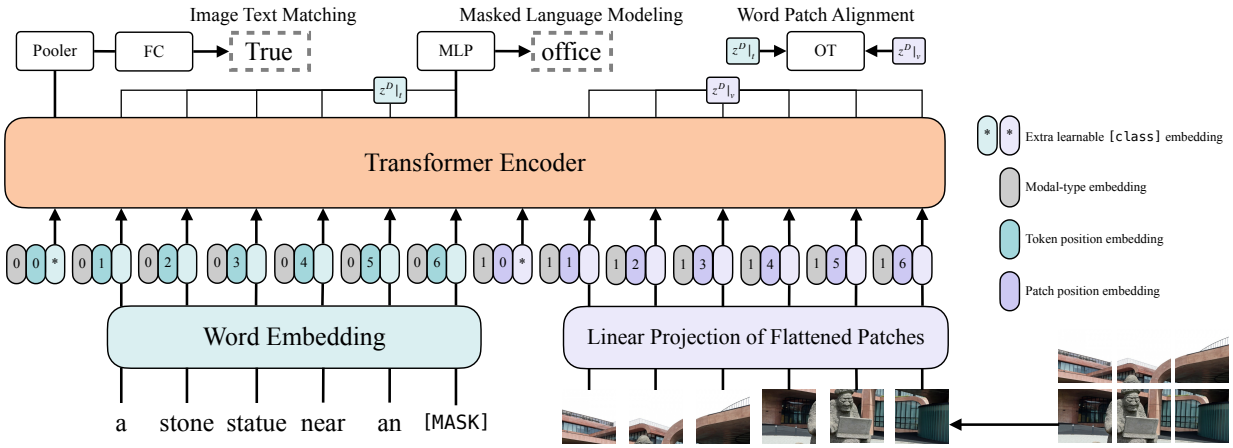

Figure 3. Model overview. Illustration inspired by Dosovitskiy et al. (2020).

inference, not to mention that it could hold performance back.

Grid Feature. Besides detector heads, the output feature grid of convolutional neural networks such as ResNets can also be used as visual features for vision-and-language pre-training. Direct use of grid features was first proposed by VQA-specific models (Jiang et al., 2020; Nguyen et al., 2020), mainly to avoid using severely slow region selection operations.

X-LXMERT (Cho et al., 2020) revisited grid features by fixing the region proposals to grids instead of those from the region proposal networks. However, their caching of features excluded further tuning of the backbone.

Pixel-BERT is the only VLP model that replaces the VG-pre-trained object detector with a ResNet variant backbone pre-trained with ImageNet classification. Unlike frozen detectors in region-feature-based VLP models, the backbone of Pixel-BERT is tuned during vision-and-language pre-training. The downstream performance of Pixel-BERT with ResNet-50 falls below region-feature-based VLP models, but it matches that of other competitors with the use of a much heavier ResNeXt-152.

We claim that grid features are not the go-to option, however, since deep CNNs are still expensive that they account for a large portion of the whole computation as in Figure 1.

Patch Projection. To minimize overhead, we adopt the simplest visual embedding scheme: *linear projection* that operates on image patches. The patch projection embedding was introduced by ViT (Dosovitskiy et al., 2020) for image classification tasks. Patch projection drastically simplifies the visual embedding step to the level of textual embedding, which also consists of simple projection (lookup) operations. We use a 32×32 patch projection which only requires 2.4M parameters. This is in sharp contrast to complex ResNe(X)t backbones[3] and detection components. Its running time is also ignorable as shown in Figure 1. We make a detailed runtime analysis in Section 4.6.

3. Vision-and-Language Transformer

3.1. Model Overview

ViLT has a succinct architecture as a VLP model with a minimal visual embedding pipeline and following the single-stream approach.

We deviate from the literature that we initialize the interaction transformer weights from pre-trained ViT instead of BERT. Such initialization exploits the power of the interaction layers to process visual features while lacking a separate deep visual embedder. [4]

$$\bar{t} = [t_{\text{class}}; t_1 T; \cdots; t_L T] + T^{\text{pos}} \quad (1)$$

$$\bar{v} = [v_{\text{class}}; v_1 V; \cdots; v_N V] + V^{\text{pos}} \quad (2)$$

$$z^0 = [\bar{t} + t^{\text{type}}; \bar{v} + v^{\text{type}}] \quad (3)$$

$$\hat{z}^d = \text{MSA}(\text{LN}(z^{d-1})) + z^{d-1}, \qquad d = 1 \dots D \quad (4)$$

$$z^d = \text{MLP}(\text{LN}(\hat{z}^d)) + \hat{z}^d, \qquad d = 1 \dots D \quad (5)$$

$$p = \tanh(z_0^D W_{\text{pool}}) \quad (6)$$

ViT consists of stacked blocks that include a multiheaded self-attention (MSA) layer and an MLP layer. The position of layer normalization (LN) in ViT is the only difference from BERT: LN comes after MSA and MLP in BERT ("post-norm") and before in ViT ("pre-norm"). The input

[3] Parameters for R50 is 25M, R101 is 44M, and X152 is 60M.

[4] We also experimented with initializing the layers from BERT weights and using the pre-trained patch projection from ViT, but it did not work.