Indian Journal of Natural Sciences *www.tnsroindia.org.in ©IJONS*

Vol.13 / Issue 73 / August / 2022 *International Bimonthly (Print)* *ISSN: 0976 – 0997*

RESEARCH ARTICLE

Vision-and-Language Transformer using Patch Projection

Guttikonda Partha Sai*, G. Keerthi Reddy, S.V.Sai Kiran and T. Venkat Narayana Rao

Department of Computer Science and Engg., Sreenidhi Institute of Science and Tech. Yamnampet, Hyderabad, Telangana, India

Received: 17 June 2022 Revised: 05 July 2022 Accepted: 25 July 2022

***Address for Correspondence**
Guttikonda Partha Sai,
Department of Computer Science and Engg.,
Sreenidhi Institute of Science and Tech. Yamnampet,
Hyderabad, Telangana, India.
Email: guttikondaparthasai@gmail.com

ABSTRACT

The performance of Vision-and-Language Pre-training (VLP) on several coupled vision-and-language downstream tasks has been demonstrated. VLP techniques currently rely largely on picture feature extraction processes, the majority of which require region supervision (e.g., object detection) and the convolutional architecture (e.g., ResNet). Although it has been dismissed in the literature, we believe it is problematic in terms of (1) efficiency/speed, as simply extracting input features requires significantly more computation than the multimodal interaction steps; and (2) expressive power, as it is upper bounded by the visual embedder's expressive power and its predefined visual vocabulary.In this study, we introduce the Vision-and-Language Transformer using Patch Projection (VLTPP), a monolithic VLP model in which the processing of visual inputs is dramatically condensed to merely the processing of visual inputs.

Keywords: performance, Vision-and-Language Pre-training (VLP), visual embedder's, Projection (VLTPP),

INTRODUCTION

The Vision-and-Language Pre-training (VLP) belongs to a joint domain of vision and language. There are few popular VLP models like Vi LBERT and Pixel-BERT. In Vi LBERT images will be undergoing CNN Backbone and Region Operation which makes the runtime around 900ms.In Pixel-BERT also image needs to undergo CNNwhich a costly operation makes the runtime around 60ms. So, We thought about removing Convolution or Region Supervision to make modals faster. We are using Patch Projection where we do not perform any costly operations like Convolution Neural Networks (CNN)[4], Regions with Convolutional Neural Networks (RCNN)[13]. While some studies include other aims and data structures, practically every VLP model has these two goals.

Figure 1 . A visual comparison of traditional VLP structures and the modal we propose. Convolutional neural networks were completely removed from the VLP pipeline without affecting performance on downstream tasks. For multimodal interactions, VLTPP is the first VLP model in which the modal-specific components require less processing than the transformer component. These models are fine-tuned on vision-and-language downstream tasks where the inputs involve two modalities, and are pre-trained with image text matching and masked language modelling objectives1 on images and their matched descriptions.

A visual comparison of traditional VLP structures and the VLTPP we propose. Convolutional neural networks were completely removed from the VLP pipeline without affecting performance on downstream tasks. For multimodal interactions, VLTPP is the first VLP modelin which the modal-specific components require less processing than the transformer component. Most VLP research has so far concentrated on boosting the power of visual embedders in order to improve performance. Because area features are often cached in advance during training time to minimize the effort of feature extraction, the drawbacks of having a high visual embedder are often overlooked in academic research. The limitations, however, are still visible in real-world applications, as queries in the wild must go through a laborious extraction process.

To that aim, we'll focus on making visual inputs lightweight and fast to incorporate. Recent work (Dosovitskiy et al., 2020; Touvron et al., 2020) shown that embedding pixels before feeding them into transformers[5] can be accomplished via a simple linear projection of a patch. Transformers (Vaswani et al., 2017)are only recently being employed for images, despite being the established mainstream for text (Devlin et al., 2019). We believe that the transformer module, which is utilised in VLP models for modality interaction, may alsoprocess visual information in place of a convolutional visual embedder, in the same way that it processes textual data. The Vision-and-Language Transformer (VLTPP) is a system that handles two modalities in a single unified manner, according to this study. Its shallow, convolution-free embedding of pixel-level inputs sets it apart from earlier VLP models. By design, removing deep embedders dedicated entirely to visual input reduces model size and run time dramatically. Figure 1 indicates that our parameter-efficient model is tens of times quicker than VLP models with region features and at least four times faster than those using grid features, while also performing similarly or better on downstream vision and language tasks.

Our aim is to transformer module extracts and processes visual characteristics instead of a separate deep visual embedder, making it the simplest design for a vision-and-language model. This architecture produces substantial runtime and parameter efficiency by default. We have also designed a web app to demo VLTPP. Where we can pass the image path and a question related to it. In response we will be sending the answer for the question. For User Interface VUE version 3[10], For Backend Python Flask app[1].

System requirements for the training and testing of the Data Sets needs more GPU and RAM to make the process faster, So it's better to do it in google colab by adding a few v RAM and v GPU. We can download the modal to the local system and use it in our flask app.

BACKGROUND KNOWLEDGE

Vision-and-Language Modals

The working of vision-and-language modals based on two points: (1) whether or whether the two modes are equally expressive in terms of specialised parameters and/or computation and (2) In a deep network, whether the two modalities interact. A combination of these two points leads to four archetypes based on runtime of Modality Interaction(MI), Textual Embed(TE) and Visual Embed(VE). Four different archetypes like VE>TE>MI(type 1), VE=TE>MI(type 2),
VE>MI>TE(type 3) and MI>VE=TE (type 4).

Type 1- In this approach image embed is costly than the remaining two Textual embed and Modality interaction. Modals like VSE++ and SCAN belong to this archetype , here they use separate embedders for image and text, then they represent the similarity of the embedded features from the twoembedders with simple dot products or shallow attention layers. Type 2- In this approach both image and textual are equally costly and same like Type 1 Dot product or shallow of two vectors. Despite CLIP's remarkable zero-shot performance on image-to-text retrieval, we could not observe the same level ofper for- mance on other vision-and-language downstream tasks. This finding supports our hypothesis that simple output fusion,even from high-performing unimodal embedders, may not be enough to master complicated vision and language tasks, emphasising the necessity for a more stringent inter-modal interaction scheme.

Type 3- Here it is the most recent VLP model use a deep transformer to model the interaction of image and text features. This involves convolutional networks to extract and embed image features, Which makes the runtime of image embed takes longer than remaining. Also Modulation-based vision-and-language models also comes under type 3, with their visual CNN[4] stems corresponding to visual embedder, RNN[14] producing the modulation parameters to textual embedder and modulated CNNs to modality interaction. Type 4- Our Modal comes under this type where runtime for text and image embeddings are equal where As with word tokens, the embedding layers of raw images are shallow and computationally light. As a result, the majority of the computation is focused on modelling modality interactions.

Modality Interaction Schema

At this stage we will have a transformer which takes visual and textual embedding sequences as input, model inter-modal and optionally intra-modal interactions throughout layers, then output a contextualized feature sequence. Classifies interaction schema into two categories (1) single-stream approaches(e,g., Visual-BERT,UNITER concentrates on image and text inputs); and (2) dual-stream approaches (e.g., Vi LBERT, LXMERT) where the two modalities are not concatenation of image and text inputs. We follow the single-stream for our modality interaction because the dual approach introduces additional parameters.

Visual Embedding Schema

While all perform ant VLP models use the same textual embedder– a tokenizer from pre-trained BERT, as well as word and position embeddings similar to BERT– the visual embedders are different. Visual embedding is still the bottleneck in most (if not all) extant VLP models. Instead of employing region or grid features, which need expensive extraction modules, we focus on cutting cuts on this phase by adding patch projection.

Region Feature[7]

This VLP model dominantly utilize region features. which are obtained from an off-the-shelf object detection like Faster R-CNN. The following is a general pipeline of generating region features. First, a region proposal network (RPN) suggests regions of interest (RoI) based on grid features aggregated from the CNN backbone. The number of RoIs is then reduced to a few thousand by non-maximum suppression (NMS). RoIsare routed through RoI heads and become region features after being pooled by operations such as RoI Align (He et al., 2017). NMS is then applied to each class, reducing the number of features to under a hundred. Object detectors, no matter how light they are, are less likely to be faster than the backbone or a single-layer convolution. Freezing the visual backbone and caching region features in advance only helps during training and not during inference, and it may hinder performance.

Grid Feature [6]

In addition to detector heads, the output feature grid of convolutional neural networks like Res Nets can beused as visual features for pre-training vision and language. VQA-specific models were the first to suggest direct use of grid features (Jiang et al., 2020; Nguyen et al., 2020), primarily to avoid requiring extremely slow region selection operations. The grid aspects of X-LXMERT (Cho et al., 2020) were revisited by fixing region proposals to grids rather than those from region proposal networks. However, the caching of features prevented the backbone from being fine-tuned any more.

Pixel-BERT[15] is the only VLP model that uses a ResNet version backbone pre-trained with Image Net classification instead of a VG-pre-trained object detector. The backbone of Pixel-BERT[15] is modified during vision and language pre-training, unlike frozen detectors in region-feature-based VLP models. Pixel-downstream BERT's performance withResNet-50 is inferior to region-feature-based VLP models, but it is comparable to other competitors when using a significantly heavier ResNeXt-152.

Patch Projection[8]

To save time, we use the simplest visual embedding approach possible: linear projection on image patches. ViT (Dosovitskiy et al., 2020) developed patch projection embedding for image classification applications. Patch projection reduces the visual embedding phase to that of textual embedding, which likewise involves basic projection (lookup) procedures.

We utilize a 32X32 patch projection, which requires only 2.4 million parameters. Complicated Res Ne(X)t backbones3 and detection components, on the other hand, are extremely complex.

Vision-and-Language Transformer

Model Overview

VLTPP has a straightforward architecture as a VLP model, with a minimal visual embedding pipeline and a single-stream approach. We deviate from the literature in that we use pre-trained ViT instead of BERT to in itialise the inter-action transformer weights. Such initialization makes use of the interaction layers' ability to process visual features in the absence of a separate deep visual embedder.

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

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

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

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

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

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

Vit consists of stacked blocks that includes a multi-headed self-attention (MSA) layer and MLP layer. The position of layer normalization (LN) in Vit is the only difference from BERT:LN comes after MSA and MLP in BERT and before in Vit. the text t ∈ RL×|V | is embedded to t ¯∈ RL×H with a word embedding matrix T ∈ R|V |×H and a position embedding matrix Tpos ∈R(L+1)×H.

The input image I ∈ RC ×H ×W is patched and flattened to v ∈ RN×(P2·C), where (P,P) is the patch resolution and N = HW/P2. v is embedded into v RNH after lin- ear projection v ∈ RN×(P2·C) and position embedding V pos ∈ R(N+1)×H The text and image embeddings are added together with their responding modal-type embedding vectors t^type, v^type R^H, and then concatenated into a combined sequence z^0. The contextualised vector z is updated iteratively through D-depth transformer layers until it reaches the final contextualised sequence z^D. p is a pooled representation of the entire multimodal input, obtained by performing linear projection Wpool R^HxH and hyperbolic tangent on the first index of sequence z^D.

The text and image embeddings are concatenated into a composite sequence z0 after being summed with their cor- responding modal-type embedding vectors type ,type ◉ℝ .Up until the final contextualised sequence zD, the contextualised vector z is iteratively updated using D-depth transformer layers. The first index of sequence zD is used to generate p, which is a pooled representation of the entire multimodal input generated by applying linear

projection Wpool RHH and hyperbolic tangent. We employ weights from ViT-B/32 that have been pre-trained on Image Net for all tests, hence the term VLTPP-B/32.5. The number of attention heads is 12, the hidden size H is 768, the layer depth D is 12, the patch size P is 32, the MLP size is 3,072, and the hidden size H is 768.

Pre-training Objectives

VLTPP is trained using two popular VLP model training objectives: image text matching (ITM) and masked language modelling (MLM).

Image Text Matching. With a chance of 0.5, we replace the aligned image with a different image at random. We compute negative log-likelihood loss as our ITM loss using a single linear layer ITM head that projects the pooled output feature p to logits over binary class. Plus, inspired by the word region alignment objective in Chen et al. (2019), we design word patch alignment (WPA) that computes the alignment score between two subsets of zD : zD it (textual subset) and zD |v (visual subset), and add the approxwas serstein weight multiplied by 0.1 to the ITM loss.

Complete Word Masking

This is a masking technique that masks all consecutive sub word takens that compose a complete word. It is also applied in BERT.

We believe that whole word masking is especially important for VLP to fully utilise information from the other modality. The pre-trained bert-base-uncased tokenizer, for example, tokenizes the word "giraffe" into three word piece tokens["gi", "##raf", "##fe"]. If not all tokens are masked, such as ["gi", "[MASK]", "##fe"], the model may instead depend on the nearby two language tokens["gi", "##fe"] to anticipate the masked "##raf."

Image Augmentation

This improves the generalization power of vision models that builds on ViT experimented with various augmentation techniques and found them beneficial for ViT training. Caching visual features restraining region-features-based VLP models from using image augmentation. Notwithstanding its applicability, neither did Pixel-BERT study its effects. During fine-tuning, we use Rand Augment (Cubuk et al., 2020)to do this. All of the original policies are used, with the exception of two: colour inversion, which is used since texts frequently contain colour information, and cutoff, which is used to remove small but crucial elements that are spread across the image. The hyper parameters are N = 2 and M = 9.

IMPLEMENTATION AND RESULTS

We can run the code as per this paper idea and the modal is ranonly on MSCOCO Data Set[11] and the results are accessible at [16].Now will download the modal to the local system , with which we can start work on User Interface and Flask app. Now, We wrote an api to receive the input and send the answeras output . In the coming sections we will compare our results with other models. screenshots are provided at the results section.

Overview

We use Microsoft COCO[11] (MSCOCO) dataset which has 113000 Images, 567000 Captions and Caption Length of 11.81± 2.81. MSCOCO and Flickr30K (F30K) (Plummer et al., 2015) are used, which have been re-split by Karpathy & Fei-Fei (2015). We fine-tune the head and data or- dering three times with various initialization seeds for the classification tasks, then present the mean results. Table 5 shows the standard deviation as well as ablation studies. We just fine-tune the retrieval tasks once. Figure 4 shows the Run Time vs test scores on two VQA Data Sets (VQAv2[16], NLVR2[17]). As discussed Linear model takes very less time approx 15 ms. VLTPP-B/32 was pre-trained on 64 NVIDIA V100 GPUs with a batch size of 4,096 for 100K or 200K steps. We train for 10 epochs for all downstream tasks, with a batch size of 256 for VQAv2/retrieval tasks and 128 for NLVR2[17].

Retrieval Tasks

On the Karpathy & Fei-Fei (2015) split of MSCOCO and F30K, we fine-tune VLTPP-B/32. We evaluate both zero-shot and fine-tuned performance in image-to-text and text-to-image retrieval8. We use the pre-trained ITM head to in itialise the similarity score head, notably the component that computes true-pair logits. We use 15 random texts as negative samples and use cross-entropy loss to tweak the model so that the scores on positive pairs are maximised.

Table 3 shows the zero shot retrieval results, whereas Table 4 shows the fine-tuned results. Despite Image BERT's pre-training on a bigger (14M) dataset, VLTPP-B/32[9] performs better in general at zero-shot retrieval. VLTPP-B/32 recalls are significantly higher than the second quickest modelwhen retrieval is fine-tuned (Pixel-BERT-R50).

User Interface

We like to provide a User Interface to the users to give inputs and witness the magic of our model. So we built a User interface in Vue version 3 and backend python Flask. Here users can give an image address in http protocol and a question related to the image after that we will send a responseas the answer for the given question.

RESULT

Model Results are available at the [16] . Figure 5 shows the User Interface screenshot.

CONCLUSION

The Vision-and-Language Transformer, a simple VLP architecture, is presented in this work (VLTPP). VLTPP distinguishes itself from competitors who rely primarily on convolutional visual embedding networks (e.g., Faster R-CNN and Res Nets). We request that future work on VLP concentrate on the modality interactions within the transformer module rather than an arms race that just increasesthe power of unimodal embedders. VLTPP-B/32 is more of a proof of concept, demonstrating thatefficient VLP[3] models without convolution and region supervision can still be competent. Finally, we'll mention afew factors that could contribute to the ViLT family.

REFERENCES

1. python FLASK Doc's available at <https://flask.palletsprojects.com/en/2.1.x/
2. UI framework Vue version 3 documentation available at <https://vuejs.org/guide/introduction.html />
3. Wonjae Kim, Bokyung Son, and Ildoo Kim "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision " 2021 arXiv:2120.03334v2 DOI: 10 jun 2021
4. jiuxiang Gu and Zhenhua Wang " Recent Advances in Convolutional Neural Networks " 2017 arXiv:1512.07108v6,DOI: 19 Oct 2017
5. Ashish Vaswani, Noam Shazeer, Jakob Uszkoreit and Niki Parma "Attention Is All You Need" 2017 arXiv:1706.03762v5, DOI : 6 Dec 2017
6. Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller and Xinlei Chen " In Defense of Grid Features for Visual Question Answering " 2020 arXiv:2001.03615v2,DOI:2 Apr 2020
7. Jiayuan Gu1 , Han Hu , Liwei Wang , Yichen Wei and Jifeng Dai " Learning Region Features for Object Detection "2018 arXiv:1803.07066v1 DOI: 19 Mar 2018
8. Yehui Tang1,2, Kai Han2 , Yunhe Wang2*, Chang Xu3 , Jianyuan Guo2,3, Chao Xu1 , Dacheng Tao4 " Patch Slimming for Efficient Vision Transformers " 2022 arXiv:2106.02852v2 [cs.CV] DOI: 4 Apr 2021
9. Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh " VQA: Visual Question Answering " 2016 arXiv:1505.00468v7 [cs.CL], DOI: 27 Oct 2016
10. Test pictures in User Interface are available at <https://unsplash.com/s/photos/splash />
11. MSCOCO Dataset is available at <https://cocodataset.org/#home />

12. VQA v2 Dataset is available for download at <https://visualqa.org/download.html />
13. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik " Rich feature hierarchies for accurate object detection and semantic segmentation " arXiv:1311.2524v5 [cs.CV] DOI: 22 Oct 2014
14. Alex Sherstinsky " Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network " arXiv:1808.03314v9 [cs.LG], DOI: 31 Jan 2021
15. Zhicheng Huang , Zhao yang Zeng , Bei Liu, Dongmei Fu1 , and Jianlong Fu " Pixel-BERT: Aligning Image Pixels withText by Deep Multi-Modal Transformers " arXiv:2004.00849v2 [cs.CV] DOI: 22 Jun 2020
16. results are available at the git repo md file <https://github.com/propardhu/4-2Project/blob/main/EVAL.md />
17. nlvr 2 data set is available to at <https://paperswithcode.com/dataset/visual-question-answering-v2-0 />
18. Patch Projection working image from following address <https://www.analyticsvidhya.com/blog/2021/03/an-image-is-worth-16x16-words-transformers-for-image-recognition-at-sca le-vision-transformers/ />

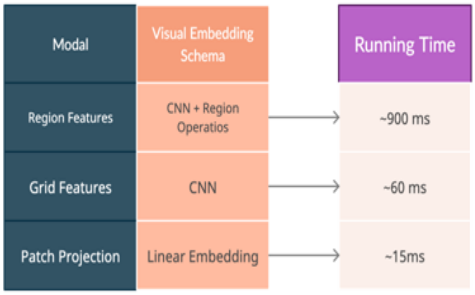

Figure 1 . A visual comparison of traditional VLP structures and the modal we propose.

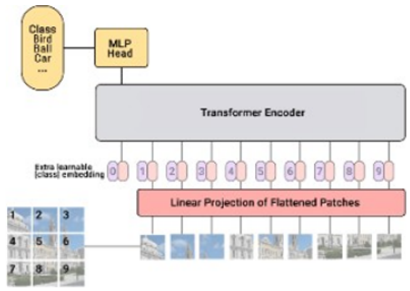

Figure 2. Patch Projection working image source [18]

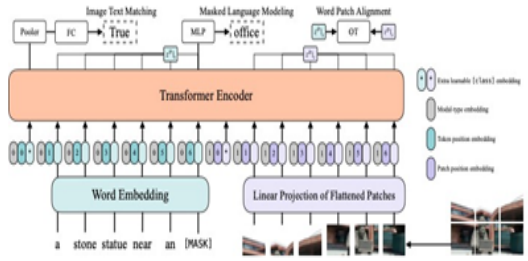

Figure 3. Model Overview. Illustration inspired by Dosovitskiy et al. (2020).

Visual Embed	Model	Time (ms)	VQAv2 test-dev	NLVR2 dev	NLVR2 test-P
Region	w/o VLP SOTA	~900	70.63	54.80	53.50
	ViLBERT	~920	70.55	-	-
	VisualBERT	~925	70.80	67.40	67.00
	LXMERT	~900	72.42	74.90	74.50
	UNITER-Base	~900	72.70	75.85	75.80
	OSCAR-Base†	~900	73.16	78.07	78.36
	VinVL-Base†‡	~650	75.95	82.05	83.08
Grid	Pixel-BERT-X152	~160	74.45	76.50	77.20
	Pixel-BERT-R50	~60	71.35	71.70	72.40
Linear	ViLT-B/32	~15	70.33	74.41	74.57
	ViLT-B/32ⓐ	~15	70.85	74.91	75.57
	ViLT-B/32ⓐⓟ	~15	71.26	75.70	76.13

Figure 4. Table of Time comparison

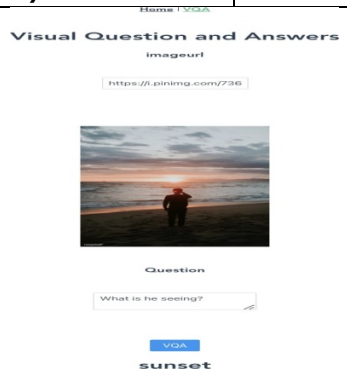

Figure 5. screenshot of User Interface. We passed an image anda question. We got answer as sunset [10]