A Picture is Worth a Thousand Words: Evaluating the Vision-Only Performance of Multi-modal Models

Team: Convoluted Thinkers

Balaji Chidambaram **Dongwon Jung** **Maheshvar Chandrasekar** **Nishant Tharani**

Abstract

The study examines the performance of multi-modal models in vision-only tasks, which has not been previously explored. The hypothesis is that vision-language (VL) models possess a more profound semantic understanding of the world, enabling them to perform better in vision-only tasks. The evaluation of various multi-modal models is conducted on an image classification task, where a fixed prompt is used to replace the missing text input. The effectiveness of different adaptation methods is compared, along with the performance of VL models versus vision models that have undergone training in image classification. The findings suggest that the quality of pre-training on image classification tasks is more critical than the adaptation method used to replace the missing text modality.

1 Introduction

Multi-modal models are deep neural networks trained to expect input of more than one modality. These models are designed to perform multi-modal tasks involving both input types, such as visual question answering. Previous work has investigated whether multi-modal learning improves performance on text-only tasks. For example, Jin et al (Jin et al., 2022) are motivated by the theory that textual training may contain insufficient information related to the visual properties of objects. Hagström and Johansson (Hagström and Johansson, 2022) directly evaluate the performance of multi-modal models on text-only tasks, testing different strategies to adapt the models to make up for the absence of visual information.

Along with these lines of work, we seek to evaluate the performance of multi-modal models on uni-modal *vision-only* task. We are motivated by the possibility that training on images alone may contain insufficient information related to the semantic properties of objects, and models trained for multi-modal networks might acquire an enhanced comprehension of the world than those trained on images alone. This might help them perform better on vision-only tasks.

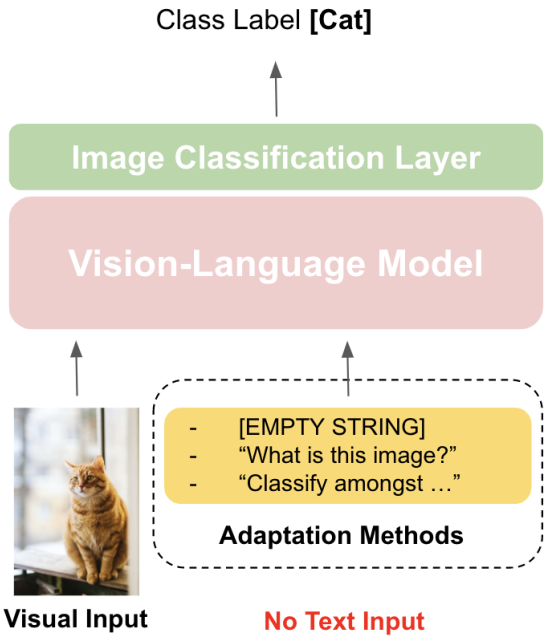

Figure 1: Overview of vision-language model architecture for image classification task. In order to fill in missing text input, we explore various adaptation methods that can potentially help the classification of the given image.

We explore various methods to fill the missing text modality of the VL models. Our objective is to identify the best adaptation method that enables VL models to leverage their semantic understanding capabilities and achieve performance comparable to vision models pre-trained on image classification tasks. Furthermore, we analyze the performance of VL models on image classification task.

2 Related Work

Prior relevant work can roughly be grouped into three categories: researches that 1) have applied a multi-modal framework only to text-only tasks,

2) as above but to vision-only tasks, and 3) are concerned with multi-modal tasks but with some element of uni-modality.

Text-Only Task. (Jin et al., 2022) aim to improve the performance of language models on text-only tasks that need an understanding of visual properties. They theorize that textual training data might not include sufficient visual knowledge, since humans assume that this is common knowledge acquired by everyone through the visual senses. Rather than using a VL model pre-trained on a multi-modal task, they take a pre-trained text encoder and perform further intermediate pre-training designed to transfer visual knowledge into it. Pre-training is done both on a text-only task on data that may contain enriched visual knowledge (image captions), and on VL tasks. This improves performance, particularly in a low-resource environment when only a few training examples are available.

(Hagström and Johansson, 2022) shares the motivation of (Jin et al., 2022), with a methodology that is closer to our work. They evaluate the performance of multiple pre-trained VL models on two text-only tasks: GLUE, a collection of nine sentence classification tasks, and VPN, a task that explicitly focuses on visual properties and concepts. To achieve this, they fine-tune the models to perform these tasks and test different adaptation methods for filling in the visual part of the input expected by the VL model. Adaptation methods include fine-tuning the model to expect only text, using averaged visual features, and using visual features from a black image. They find that the multi-modal models are not superior to a baseline text-only model (BERT-base) on either of the task groups and that adaptation methods matter more for performance on VPN.

The clear difference between our project and these papers is that they consider the effect of additional visual knowledge on text-only task performance, whereas we consider the effect of additional textual knowledge on vision-only task performance. (Jin et al., 2022) share our general idea that human performance in tasks involving one modality benefits from human knowledge in other modalities, so the same might be true for models. (Hagström and Johansson, 2022) 's work can be regarded as the methodological equivalent of our project for text-only tasks.

Vision-Only Task. (Srinivasan et al., 2022) presents a benchmark to explore continual learning in VL tasks that can be extended to vision-only tasks. Relevant to our study, they use a Vision-Language Transformer (ViLT) model and transfer it downstream to few-shot uni-modal tasks. To replace the missing text input, they use the phrase "This is an image". Although the work experiments with multi-modal models on vision-only tasks, the main focus is on continual learning methods in a multi-modal setting rather than testing the capabilities of multi-modal models on vision-only tasks. Also, the work uses a single adaptation method for missing language modality, while we will compare several adaptation methods.

(Sariyildiz et al., 2020; Desai and Johnson, 2021) demonstrate that it is effective to use natural language as a form of guidance for developing visual features which can be transferred to visual-only task. They utilize two uni-modal models to encode visual and textual features to pre-train image-caption pairs. They remove the textual framework and transfer the visual component to downstream visual-only tasks. They have shown that multi-modal pre-training is efficient and effective on downstream visual tasks, but they use two separate uni-modal models. On the other hand, our work is more focused on keeping the 'multi-modal' part of the model architecture and exploring approaches to fill in the text placeholder.

(Frank et al., 2021) compares the performance of different multi-modal models on uni-modal tasks, with the motivation of devising a diagnostic test to see how well models predict masked data when ablated inputs in the other modality are given. They pre-train the models on a vision task named Masked Region Classification (MRC) that predicts the target object class of the masked visual region of the image, given the visual context and the varying amount (all, partial, none) of text captions. Their objective differs from ours since they test whether each modality can aid each when either is ablated. Our objective is to evaluate the capability of multi-modal models in performing vision-only tasks where language input is absent. If none of the text captions are given, the task becomes a vision-only task where only given the visual surrounding of the target object. However, the task is different from the traditional object classification task since the region we want to predict is removed, and only visual surrounding is given, making the task less

meaningful. Also, they do not consider different adaptation methods to fill in the text expected by the multi-modal models.

Multi-modal Tasks with Missing Modalities (Ma et al., 2022) investigates the robustness of multi-modal transformer models to missing modalities in input data. The authors train VL models on data with one modality missing and test them on data where the text is either 100% or 30% present. The authors find that the network's performance degrades drastically with missing modalities and that the text modality appears to be dominant in multi-modal transformer learning. However, the study is limited to exploring only multi-modal datasets, which may limit its applicability to classification datasets. Instead of focusing on missing modalities during training, our work evaluates fully trained VL models on uni-modal vision-only tasks.

(Salin et al., 2022) work on uni-modal tasks with 2 pairs of multi-modal data to investigate the amount of information captured from different modalities in a multi-modal task. The authors conclude that fine-tuning a vision-language embedding on multi-modal tasks does not necessarily improve its multi-modal ability. They further find that the textual modality is more important than the visual modality for model decisions.

(Ma et al., 2021) proposes a new method SMIL which uses Bayesian Meta-Learning in dealing with missing modalities regarding flexibility and efficiency. The paper discusses using the latent feature space so that embeddings of one modality can be used to approximate the ones of the whole modality. The proposed method implements a meta-learning framework that learns feature reconstruction and feature regularization networks to approximate modality. Although the Bayesian Meta Learning framework does a great job of approximating missing modalities during training, there is a higher chance that the predicted posterior distribution may be different from real-world modality, and there is a high probability that the meta-learning framework has poor performance on data where the posterior distribution modality is uncertain.

3 Problem Definition

Among various vision-only tasks, we focus specifically on image classification. Image classification is a computer vision task that involves assigning a label or a category to an image based on its content. Its applications span a wide range of fields, such as object recognition, face detection, medical diagnosis, and autonomous driving.

Figure 1 illustrates the overview of the VL model architecture. To make VL models perform image classification, we add a linear classifier module at the end, which takes the hidden state output of the base model as input. We fine-tune the linear classifier on our image classification dataset. All VL models use pre-trained weights trained on Vision Question Answering (VQA) task. Additionally, we freeze all model layers except the added classifier layers and the pooling layer preceding the classifier layers.

4 Models

We explore five VL models that are pre-trained on Visual Question Answering (VQA) task. We further discuss how we preprocess images to feed into the VL models.

4.1 VisualBert

VisualBERT (Li et al., 2019) is a single-stream model where the visual input embeddings are extracted by passing the image through an object detector model, e.g., Faster R-CNN.

For the experiment, we emulate the object detection pipeline from the model's respository, which uses Meta's Detectron2 library and a Mask R-CNN Instance Segmentation model with a ResNet-101 backbone. The emulated pipeline produces a (36,2048) shaped visual embedding for each image, corresponding to information about 36 bounding boxes with a hidden dimension of 2048.

4.2 LXMERT

Learning Cross-Modality Representations from Transformers (LXMERT) (Tan and Bansal, 2019) is a framework for learning Vision-and-language reasoning. LXMERT proposes understanding the alignment and relationships between visual concepts and language semantics with a large-scale Transformer consisting of three encoders: an object relationship encoder, a language encoder, and a cross-modality encoder.

The visual inputs to LXMERT must be object-level image embeddings that have features of detected objects as image embeddings rather than using feature map output by a convolutional neural network. We generate the input visual embeddings

by passing the input images through an object detection model, such as Generalized RCNN.

4.3 ViLT

Vision-and-Language Transformer (ViLT) (Kim et al., 2021) is a single-stream VL model that uses convolution-free embedding for pixel-level image input that was introduced by ViT (Dosovitskiy et al., 2020), instead of visual features from object detection models such as Faster R-CNN. The input images are sliced into fixed-size patches on which linear projection can be performed. The result patch projection embeddings are concatenated with text embedding to be processed into transformer layers. For our experiment, we directly feed the patch projection embedding of the images into the ViT-B/32 transformer layer.

4.4 FLAVA

FLAVA (Singh et al., 2022) is based on the idea that a reliable foundational VL model should demonstrate proficiency in both visual and language-based tasks, respectively, as well as the ability to effectively combine and apply both modalities in cross-modal and multi-modal scenarios.

In this work, we experiment with two types of FLAVA models: 1) FLAVA-vision and 2) multi-modal FLAVA model. FLAVA-vision is the uni-modal FLAVA model that uses only the image encoder, while the multi-modal FLAVA model uses both a uni-modal encoder and a multi-modal encoder. For both the image and text encoder of FLAVA, we use a ViT-B/32 transformer.

4.5 BLIP

BLIP (Li et al., 2022) is a framework for vision-language understanding and generation that consists of a Multi-modal mixture of Encoder-Decoder (MED) model architecture and a Captioning and Filtering (CapFilt) dataset boostrapping method. The model employs a vision transformer (ViT) as the image encoder and BERT as the text encoder. For our experiment, we use ViT-B/16 and BERT-base as the uni-modal encoders.

5 Adaptations to Vision-Only Input

We compare the performance of different ways of replacing the missing text modality, expected by the multi-modal models.

5.1 Empty String

We tokenize an empty string and pass the result as the text input for a VL model. This simple method serves as a baseline.

5.2 Question

We use the fixed text input "What is this image?" as the text input for VL models. We theorize that this input may help harness the Visual Question Answering (VQA) capabilities of our models, all of which were trained on this objective.

5.3 Class Names

We use a text input that lists all the possible classes the image could belong to, in the format "The image belongs to one of the following classes: 'airplane,'..., 'truck.'". For CIFAR-10, our list contains 10 classes, and for CIFAR-100, our list contains 100 classes. Here, we theorize that this may further harness the VQA capabilities of our models by priming them to choose an answer from one of the listed classes, which may feed into better training a classifier.

5.4 Task Description

We modify the Class Names adaptation to include a description of the task and source dataset in the format "This is a classification problem from the [dataset]. Classify the images amongst the classes 'airplane,'..., 'truck.'". Again, we list all ten classes for the CIFAR-10 dataset and all 100 classes for the CIFAR-100 dataset. We hypothesize that describing the task as a classification problem may help the model better utilize its VQA capabilities by directing it to perform classification, potentially producing intermediate outputs better suited for training a classifier.

6 Experiment Setup

6.1 Train and Evaluation

To assess the model's ability on the vision-only task, we train the models on the image classification task, where the models are expected to output precisely one class label. We train the models to minimize cross entropy loss between the predicted and true class labels. We use accuracy as our evaluation metric. We have included a link to our code repository in Appendix B.

	CIFAR-10				CIFAR-100			
	Empty	Question	Classes	Task	Empty	Question	Classes	Task
ViLT	94.62	94.61	94.89	93.81	78.03	75.99	74.51	77.36
FLAVA	93.88	93.92	94.17	93.94	69.30	71.76	74.62	72.56
BLIP-exact	95.07	95.07	94.44	94.05	67.65	59.65	60.96	47.23
BLIP-gpt	-	-	-	-	78.9	79.1	76.6	78.1
VisualBERT	60.32	58.79	58.73	57.48	35.51	31.64	31.43	30.85
LXMERT	71.36	69.61	67.04	65.18	44.25	43.01	39.63	38.88
ViT-B/16	95.56				**80.88**			
FLAVA-vision	**95.72**				79.72			

Table 1: Performance comparison of adapted VL models on the two image classification datasets. We use the accuracy metric for comparison. "Empty," "Question," "Classes," and "Task" indicate adaptation methods "Empty String," "Question," "Class Names," and "Task Description." BLIP-exact and BLIP-gpt represent the performance result evaluated on two methods: exact matching and GPT prompting. The performance of BLIP refers to the performance of BLIP-gpt in our discussion.

6.1.1 BLIP Evaluation

BLIP is a generative model, while our other models are discriminative. This means our evaluation method has to be different for BLIP. BLIP generates text output in the form of tokens. The first evaluation method we try is 'exact matching': we consider the output correct if its text exactly matches the class label on a character level. Results using this method are labeled 'BLIP-exact' in Table 1.

The downside of this method is that BLIP may generate output text that does not exactly match the class label but is still reasonably correct (for example, it could output "a cat" when the label was "cat"). In particular, we observed that BLIP often produced mangled output such as "televis" when the label was "television", or "caterllar" when the label was "caterpillar". A reasonable human would match these outputs to the appropriate label, but our evaluation method marks them as incorrect.

To address this issue, we make use of gpt-3.5-turbo-0301 (henceforth "GPT"), an LLM (Large Language Model) trained by OpenAI for an interactive chat. We design a 'prompt' that presents the model's output and candidate class labels. Then, we ask GPT to choose which class label each output is synonymous with. We leave GPT the option to choose none of the class labels if it feels that none of them is a good match. We make use of OpenAI's best practices for prompt design by using """ to separate instructions and context and providing example input and output (few-shot prompting). The full prompt is presented in Appendix C.

We face the following tradeoff when choosing how many outputs to ask GPT to evaluate at once. Suppose we present just one output at a time. In that case, the evaluation process is slow and costly (because the OpenAI API rate limits the number of API calls that can be made in a given time duration, and the explanatory prompt has to be sent in its entirety each time). However, if we present too many outputs, GPT tends to lose track and skip over some of them, rendering the output useless. Presenting ten outputs at a time avoids this issue while achieving acceptable speed and cost.

We refer to this process as 'translation.' The translated outputs provided by GPT are then evaluated against the class labels using the 'exact matching' method described earlier. Results using this method are labeled 'BLIP-gpt' in Table 1.

6.2 Datasets

We utilize two widely used image datasets in our experiment: (1) CIFAR-10 (Krizhevsky et al., a) contains 60,000 32x32 pixel images divided into 10 classes, each containing 6000 images. (2) CIFAR-100 (Krizhevsky et al., b) consists of 100 classes, each of which contains 600 32x32 pixel images. For each class, 500 images are used for training, and 100 images are used for testing. We use this split for training and testing, respectively.

6.3 Baseline

We compare our adapted VL models with state-of-the-art uni-modal vision models, Vision Transformer (ViT) (Dosovitskiy et al., 2020) and FLAVA-vision.

ViT is a BERT-like transformer encoder model, pre-trained on one of the following datasets; Im-

Models	# Images	Backbone
ViT-B/16	IN-21K (14M)	ViT-B/16
FLAVA-vision	IN-1K (1M)	ViT-B/32
FLAVA	IN-1K (1M)	ViT-B/32
ViLT	IN-21K (14M)	ViT-B/32
BLIP	IN-21K (14M)	ViT-B/16
VisualBERT	-	Faster R-CNN
LXMERT	-	Faster R-CNN

Table 2: Number of images that the visual backbone of each model is pre-trained on with image classification objective. "IN-21K" and "IN-1K" indicate "ImageNet-21K" and "ImageNet-1K", respectively.

ageNet (Russakovsky et al., 2015) or JFT300M (Sun et al., 2017). Among different versions of ViT, we choose the ViT-B/16 model pre-trained on ImageNet-21K as the baseline of the experiment. As the name indicates, ViT-B/16 uses a sequence of 16x16-sized patches as the projection embedding.

FLAVA-vision is the uni-modal model of FLAVA, which uses an image encoder. In our experiment, we use the model that is pre-trained on ImageNet-1K.

6.4 Parameter Settings

We implement our VL models and the baseline model using Huggingface and Pytorch. We optimize the models with AdamW optimizer (Loshchilov and Hutter, 2017) and set the learning rate to 5×10^{-5}. Moreover, we rely on empirical knowledge to determine the number of epochs for each model to achieve optimal performance.

7 Results

In this section, we analyze the experimental results reported in Table 1. We calculate the accuracy of the test set. Note that when discussing the performance of BLIP, we mean the performance of BLIP-gpt. Here are some observations from the results.

- Overall, the uni-modal models, ViT and FLAVA-vision, outperform all the multi-modal models in both datasets. This is an expected result since uni-modal models are already pre-trained on the exact uni-modal task and are thus more suited to the task.

- VL models that require external visual embeddings (VisualBERT and LXMERT) perform relatively poorly on both datasets. We theorize that this could be because the representation of an image as embeddings of individual objects in the image, rather than the image as a whole, is not conducive to classifying the entire image.

- Generally, empty string adaptation is consistently robust in both datasets. Even though an empty string does not contain useful information on the task, the method is straightforward for the models to interpret.

- For BLIP and ViLT, question adaptation and informative prompts (Class Names and Task Description) are not helpful. This is an unexpected result since we theorized that giving more context on the task would help the model perform better. This shows that the models barely need more context in the text and already understand the vision-only task well without any more context.

- FLAVA performs more poorly than ViLT and BLIP on both datasets. This can be explained by the fact that FLAVA is less pre-trained on image classification than ViLT and BLIP. Table 2 shows the number of images the models pre-trained on with the image classification task. ViT vision backbones of BLIP and ViLT are pre-trained on ImageNet-21K with 14 million images, while the vision backbone of FLAVA is pre-trained on ImageNet-1K, with only 1 million images. This proves that the more pre-training on image classification, the higher the performance.

- The performance of BLIP on CIFAR-10 using both evaluation methods (exact matching and GPT prompting) are the same. This indicates that the outputs are the same as the class label string. This would have been possible since all the class labels in CIFAR-10 consist of only one token. This makes the model easier to match the class label string exactly.

- BLIP model evaluated with GPT prompting outperforms the BLIP evaluated using the exact matching and all the other VL models. This shows that the generative model, like BLIP, can be competitive on tasks like classification, even though the model is not built on the specific task. Interestingly, BLIP performs best with empty string adaptation, where there

is no context or task information. This implies that even though the model is not instructed on what to do, it knows exactly what to do and utilizes its semantic understanding of the images to generate text outputs. However, there is no suitable evaluation method for such a generative model, and the performance can vary heavily on the evaluation methods.

8 In-depth Study

8.1 Image Object Embedding

Multi-modal models like Visual-BERT and LXMERT use models like Faster R-CNN to extract visual features from the image and encode spatial information about objects in the image. Faster R-CNN is a widely used object detection model that outputs a set of bounding box embeddings, where each embedding corresponds to a specific bounding box in the image. See Figure 2 for a visualized example of the bounding boxes produced by our object detection model on an example image. In LXMERT and Visual-BERT, these bounding box embeddings are used as inputs to the visual modality of the model. Specifically, each bounding box embedding is processed by a separate linear layer to produce a fixed-length feature vector. These feature vectors are then combined with the textual input modality of the model to perform downstream tasks.

However, these bounding box embeddings are intended for tasks that require reasoning between visual and text modalities. The bounding box embeddings are designed to capture information about the spatial layout and arrangement of objects in the image for models that require spatial reasoning. However, the focus of the image classification task is on recognizing the overall content of the image rather than the specific locations of objects in it. By passing bounding box embeddings, we introduce unnecessary noise into the model, which will be forced to process information about specific locations that are not relevant for image classification. Furthermore, introducing the embeddings will increase the computational complexity of the model, potentially slowing down training and inference times. This can be seen by the fact that the training time required for VisualBERT and LXMERT is much higher than other models used in our experiments.

In conclusion, introducing the bounding box embeddings increases the noise and computational complexity of the model, forcing it not to focus on the image classification task at hand.

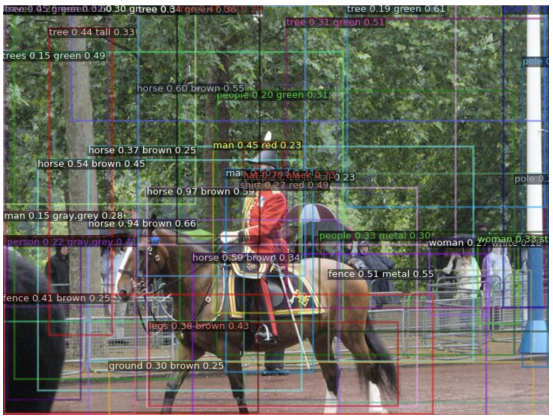

Figure 2: Example of bounding boxes produced by an object detection model on a candidate image. A set of each embedding (one for each bounding box) are used by LXMERT and VisualBERT as a representation of the image input.

8.2 Glove Embedding for BLIP evaluation

In order to alleviate the inevitably low accuracy using exact matching evaluation, one approach that was attempted was to match the generated outputs of BLIP to Glove Embeddings ((Pennington et al., 2014)). For instance, if BLIP generates the output "cat," we map this string into the Glove embedding, which represents cat. For the generated outputs with more than one word, we converted each word into vectors and then summed them as a vector addition. We then computed the cosine similarity between the resulting vector and the vectors of all labels. The label with the highest similarity was chosen as the output.

However, this approach did not lead to improved model performance. This may be because many of the words produced by the model either perfectly matched a specific word in the list of labels (resulting in a 100% cosine similarity) or contained spelling errors. In the latter case, the model selected a random word from the label list, which degraded the performance. In addition, the results were heavily reliant on the quality of the embeddings. These issues likely contributed to the need for improvement in the model's performance.

9 Conclusion

We have investigated the potential semantic understanding capabilities of ViLT, FLAVA, BLIP, VisualBERT, and LXMERT on image classification task by employing four different adaptation meth-

ods. We have concluded that the choice of adaptation method utilized to substitute for the absence of textual modality has a comparatively minor impact in comparison to the extent of pre-training undertaken by the models on image classification. These experimental findings demonstrate that (1) vision models outperform all the other VL models, and (2) ViLT and BLIP outperform FLAVA.

Additionally, we have observed that BLIP, a generative model, outperforms all the other VL models. More surprisingly, BLIP performs better when given an empty string or simple question adaptation, where neither contextual information nor task-specific directives are provided. This indicates that despite the absence of explicit instructions, the model can leverage its semantic understanding of the images to generate text outputs. However, the evaluation for this type of generative model is open to debate.

For future research in this area, we suggest further exploring the generative class of multi-modal models and their evaluation methods, as this is where we saw the best performance. Also, we recommend pre-training the multi-modal models on the image classification task for a more fair comparison with vision-only models, which are already pre-trained on the task. This is outside the scope of our project due to limited computation resources.

References

Karan Desai and Justin Johnson. 2021. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers. *arXiv preprint arXiv:2109.04448*.

Lovisa Hagström and Richard Johansson. 2022. How to adapt pre-trained vision-and-language models to a text-only input? *arXiv preprint arXiv:2209.08982*.

Woojeong Jin, Dong-Ho Lee, Chenguang Zhu, Jay Pujara, and Xiang Ren. 2022. Leveraging visual knowledge in language tasks: An empirical study on intermediate pre-training for cross-modal knowledge transfer. *arXiv preprint arXiv:2203.07519*.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. a. Cifar-10 (canadian institute for advanced research).

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. b. Cifar-100 (canadian institute for advanced research).

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. 2022. Are multimodal transformers robust to missing modality? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18177–18186.

Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. 2021. Smil: Multimodal learning with severely missing modality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2302–2310.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. 2022. Are vision-language transformers learning multimodal representations? a probing perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11248–11257.

Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. 2020. Learning visual representations with caption annotations. In *Computer Vision–ECCV 2020:*

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16, pages 153–170. Springer.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650.

Tejas Srinivasan, Ting-Yun Chang, Leticia Leonor Pinto Alva, Georgios Chochlakis, Mohammad Rostami, and Jesse Thomason. 2022. Climb: A continual learning benchmark for vision-and-language tasks. *arXiv preprint arXiv:2206.09059*.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Hao Tan and Mohit Bansal. 2019. LXMERT: learning cross-modality encoder representations from transformers. *CoRR*, abs/1908.07490.

A Individual Contribution

Maheshvar Chandrasekar

- Implemented FLAVA-no-text and FLAVA-question models
- Intergrated the FLAVA hugging face API to the FLAVA Pytorch API for ease of use.
- Wrote section 3.4 and contributed to sections 6 and 7

Balaji Chidambaram

- Integrated Generalized RCNN object detection model to generate visual embeddings for visual inputs to LXMERT.
- Implemented LXMERT-no-text, LXMERT-question, LXMERT-class and LXMERT-task experiments.
- Wrote section 3.2, and contributed to sections 5 and 7.

Dongwon Jung

- Implemented ViLT model with all the adaptations (no-text, question, class, task).
- Implemented BLIP model with all the adaptations (no-text, question, class, task).
- Conducted overall experiments in a unified settings.
- Wrote section 3, 4.3, 4.5, 6, 7, 9 and contributed to all the rest of the sections.
- Created Figure 1, Table 1 in introduction.

Nishant Tharani

- Replicated Faster R-CNN object detection pipeline from VisualBERT demo to create visual embeddings.
- Implemented VisualBERT experiments.
- Worked on BLIP experiments, and implemented the BLIP-GPT evaluation method.
- Contributed mainly to sections 1, 2, 4, 5, 6 of the report, as well as final editing.

B Code Repository

The code that we wrote for this report can be found at the following GitHub repository: https://github.com/JungDongwon/adapt-VL-models-to-vision-only-tasks .

C Prompt Examples

Presented below is the prompt that we fed to GPT to translate the outputs of BLIP to CIFAR-100 class labels. This method was described in section 5.1.1. The strings 'EXAMPLE1', 'EXAMPLE2', etc, at the end of the prompt, were related with 10 actual outputs from the BLIP model, one on each line.

```
I will give you a list of reference words
and phrases, followed by a list of candidate
words and phrases. For each candidate, if
the candidate is synonymous with an item
from the reference list, please print the
item. If not, please print the string
"NONE". Please print just the output, with
no explanation.

Reference list: """
worm
tank
crab
bee
orchid
skunk
woman
hamster
```

```
plate
table
house
possum
lobster
rocket
elephant
oaktree
girl
whale
bridge
leopard
sea
sweetpepper
wardrobe
crocodile
pickup_truck
chimpanzee
bus
clock
castle
bed
pear
snake
squirrel
bear
telephone
forest
trout
seal
flatfish
pinetree
boy
cockroach
apple
aquarium_fish
skyscraper
lamp
porcupine
tulip
butterfly
couch
wolf
man
cattle
bottle
raccoon
dolphin
lizard
train
dinosaur
mushroom
caterpillar
poppy
sunflower
keyboard
bicycle
fox
plain
turtle
mountain
ray
cloud
kangaroo
baby
cup
beaver
shark
mouse
television
snail
spider
shrew
rose
motorcycle
camel
rabbit
otter
willow_tree
orange
streetcar
road
maple_tree
lion
palm_tree
tiger
beetle
lawn_mower
can
chair
bowl
tractor
"""

Example input: """
TV
rooftop
ocean
ice
"""

Example output: """
```

```
television
NONE
sea
NONE
"""

Input: """
EXAMPLE1
EXAMPLE2
EXAMPLE3
EXAMPLE4
EXAMPLE5
EXAMPLE6
EXAMPLE7
EXAMPLE8
EXAMPLE9
EXAMPLE10
"""
```