MMM-CLIP: Multi-label-image-classification with Multi-method CLIP

Yitong Chen[1*] Zhipeng Qiu[1*] Xingsong Ye[1*]

Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University

{yitongchen20, zpqiu20, xsye20}@fudan.edu.cn

Abstract

The concerns surrounding single-label image classification have been extensively explored, yet the multi-label image classification problem, often as an adjunct task of object detection, has not been fully studied separately. The conventional approach to solving this task involves using some pre-trained models as the backbone, followed by a fully connected layer to form the classification model, which is then trained on a large scale dataset. While this method is versatile, it is not well-suited for few-shot scenarios and fundamentally incapable of handling zero-shot situations. In our study, we present multiple novel universal multi-label image classification schemes based on the CLIP model including a text prompt-based scheme and a scheme involving modifications to the CLIP model by adding adapters. We have tested the effectiveness of these schemes on multiple datasets. In particular, our CLIP-MLD-Adapater Method has obtained SOTA outcomes in traditional learning, few-shot learning, and zero-shot learning. Our code is available at https://github.com/CV-Magician/MMM-CLIP.

1. Introduction

Image classification has a long history in the Computer Vision (CV) field. After the ImageNet [2] project and the introduction of AlexNet [9], the Convolutional Neural Network (CNN) [6, 9, 10, 22, 24] proved to be a sufficient approach for CV tasks. However, the original AlexNet was introduced for the single-label image classification task. In the real world, it is very common for an image to contain multiple objects. Therefore, multi-label image classification is more realistic, which is the task that we are concerned with in our paper.

Unlike the old ImageNet period, nowadays we have multiple choices to solve this problem. The big success of Transformer in the Natural Language Processing (NLP) field proves the power of pre-trained models [25]. For a long time, the CV field has known that pre-trained models' parts can serve as a feature extractor for downstream tasks. However, most models in the CV field are trained through supervised learning, which means they need a large amount of annotated data to train. Therefore, how to efficiently train a pre-trained CV model using images without annotations remains a big problem. By cutting images into patches served as image token embeddings, images can be considered as sequences and used for training a Vision Transformer (ViT) [4]. Therefore, nowadays the ViT is frequently used for image feature extraction and used in many downstream CV tasks. By adding some additional layer after the pre-trained ViT as a new model and fine-tuning it in a small dataset, it can achieve SOTA performance compared with other elaborately designed models.

In the field of multi-label image classification, due to issues such as uneven distribution and poor data quality in the target dataset, we hope to pre-train the model on a larger and richer dataset to learn some general knowledge, and

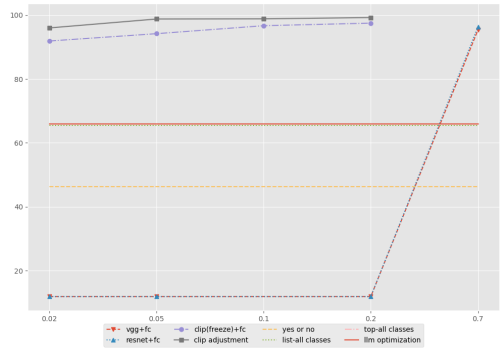

Figure 1. Evaluation of baselines and our multiple methods using Kaggle Dataset Benchmark in different learning scenarios (adjusting the train-test ratio). The metric for evaluation is the mAP (in %), where a higher score indicates a greater performance.

*Equally contributed & Co-Corresponding author.

then fine-tune in zero-shot or few-shot manner on the target dataset to prevent the model from being misled. Therefore, there are many powerful CV models which have strong zero-shot or few-shot capacity, especially CLIP [18]. It is a model that both contains an image feature extractor and a text sequence feature extractor, which means it is a multimodal model. Since it is trained through the alignment of given text and image pairs, it can even do great performance in zero-shot situations. This kind of model is called Vision-Language Pre-training (VLP) [11, 12, 16, 18], they learn and sense the correspondence between the text and the image.

To utilize the CLIP in the multi-label image classification task, there are two main approaches. The first is to design proper prompts and it can even perform well in zero-shot. The second is to redesign the CLIP structure, such as simply adding a classification head. The latter is much more complex and cannot always ensure the redesigned structure can perform better than the original one.

It is easy to design a CLIP prompt for the single-label image classification, for example, a prompt like "A photo of *CLASS*." is sufficient. However, different prompt templates may influence the classification performance. If we try negative prompts like "A photo of no *CLASS*.", it may give us additional information about the image. Since the CLIP is trained with positive prompts, which means it may have a weak ability to process negative prompts. Therefore, researchers have pointed out that we need to train an additional negative text encoder to fit this requirement [26]. In the meantime, the permanent prompt template may not be the best prompt to lead the right classification. With the Large Language Model (LLM) [1, 3, 17, 19] prompt optimization, these prompts can be polished and processed better to be fed into the text encoder in the CLIP, which may lead to a better classification result [28].

Another approach is to design a proper classification head for this task, which is called Adapter. Since the CLIP is trained by comparison, which means it has a strong ability to do the single-label image classification in zero-shot. To the best of our knowledge, earlier research proves that a well-designed classification head can lead to a better performance [20] and the current SOTA model also uses a CLIP as a backbone. However, the classification head is elaborately designed for the multi-label image classification task [7] and it's harder to replicate.

Considering the multi-label image classification task, the dataset can be found in [21]. It contains 7843 images, which are annotated with 10 classes. It can be solved directly by using the pre-trained model and method mentioned above in conventional learning. However, it is difficult to solve this problem in the few-shot even zero-shot situation, which is more concern and takes the main energy to solve for our group. Specifically, our contributions are as follows:

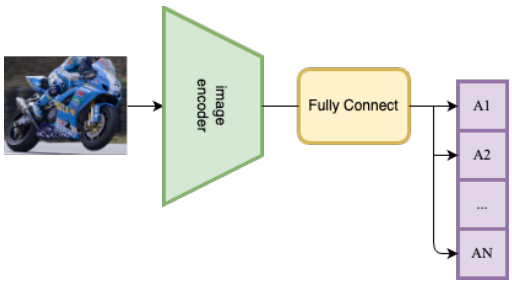

Figure 2. The baseline model structure.

- We introduce multiple methods to prompt and fine-tune the CLIP, proving its powerful capabilities
- One of our methods work perfectly in few-shot scenarios, delivering SOTA performance with minimal training.
- The approaches that we introduce are transferable to other datasets.
- These adapters are simple in structure and can be quickly replicated and verified.

2. Baseline

In the absence of a predefined benchmark score for evaluating our approach on the Kaggle dataset dedicated to the multi-label image classification task, we seek to establish a baseline performance using an existing methodology. To address this need, we turn to the work presented in the repository at hellowangqian's Github [8, 27], which employs pre-trained visual models, specifically VGG [23] and ResNet101 [6], followed by a fully connected layer for multi-class classification tasks. Specifically, the model structure is illustrated in Figure 2.

2.1. Pre-Trained Model

The choice of pre-trained models is rooted in their ability to capture intricate features from large-scale image datasets. VGG and ResNet101, being well-established architecture, have demonstrated success in various computer vision tasks. Leveraging the pre-trained weights of these models allows us to benefit from the knowledge acquired during their training on extensive datasets, providing a robust starting point for our multi-label image classification task.

2.2. Fully Connected Layer

The fully connected layer is used to transform the extracted features from the pre-trained models into predictions for the specific classes present in the dataset. In the context of multi-label image classification, it's crucial for accommodating the simultaneous prediction of multiple labels associated with each input sample. The customized layer adjusts the output dimensions to align with the number of classes in

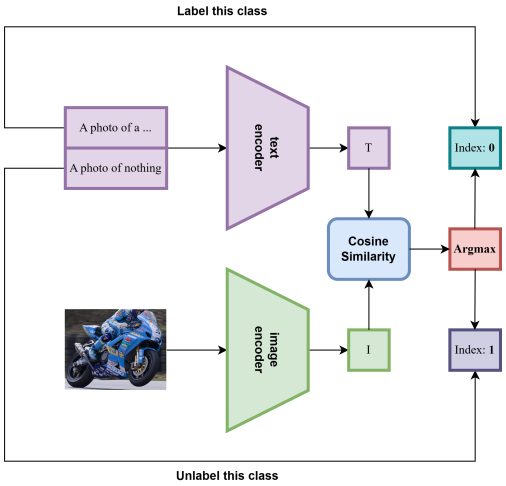

Figure 3. The inference pipeline of **Yes or No** aligns with CLIP's, except that the input takes our prompt template.

datasets, enabling the model to capture the inherent features of the class labels.

3. Multi-method CLIP

3.1. Multi-method to Prompt CLIP

The inherent power of CLIP in handling a wide array of tasks is well-established. Inspired by *"Prompt: the fourth paradigm in NLP"* [14], we explore strategies to adapt our frozen CLIP model for downstream tasks with zero-shot. Our objective is to transform the task into a format that CLIP can seamlessly address, effectively converting it into a more tractable single-label problem. To achieve this, we attempted four methods: Yes or No, List-all Classes, List-all Classes and LLM Optimization. The first method can be regarded as a separate idea, while the subsequent ones should be viewed as incremental refinements building upon the second method, and their performance is shown at Table 4.

(1) Yes or No In this method, we perform binary classification for each class of each image, determining whether it contains the target object. For "existence" state, we utilize the foundational CLIP template *"A photo of a..."*. To mitigate CLIP's dependence on label content, the negative case, indicating "non-existence", is formulated as *"A photo of nothing"* instead of *"A photo without a..."*. The cosine similarity results between these two label constructions and the image, obtained through the CLIP text and image encoders, are then assigned a corresponding label of 1 or 0 based on the similarity. For example in Figure 3, if we ask about the "Motorcycle" in this picture, it will select 0-index and mark the motorcycle label as 1; but for "Trunk", the negative is selected and the trunk label is marked as 0.

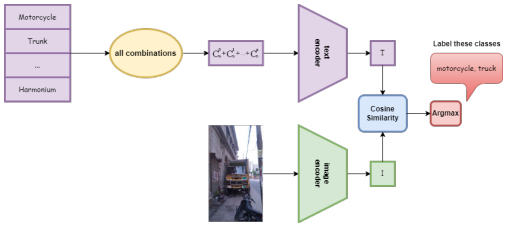

Figure 4. A brief illustration of the **List-all Classes** framework. It just gets all the text combinations for input, and then gets the answer combinations for output.

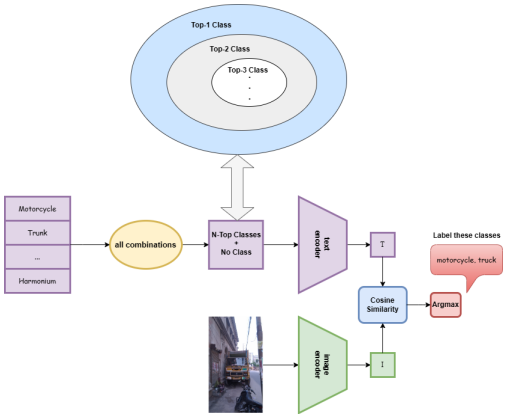

Figure 5. The detailed pipeline of **Top-all Classes** first sorts the text through CLIP and obtains the reduced text combination result for input, then follows the **List-all Classes** to get its answers.

(2) List-all Classes For every image, we exhaustively enumerate all possible multi-classification scenarios. The result is given by the sum of combinations:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

The prompt content involves a simple combination of classes without using templates or embellishments. For the optimal results provided by CLIP, all labels corresponding to the identified combinations are marked as 1, others are labeled 0. See Figure 4 for details.

(3) Top-all Classes Enumerating all possible combinations in the "List-all" approach becomes increasingly challenging as the number of categories grows exponentially. Even with the capability to handle Kaggle's 10-class dataset, when faced with the VOC-2007 dataset, issues of *"CUDA out of memory"* arise. The comparative results from CLIP not only provide information about the highest values but also incorporate ranking information derived from large-scale text-image pair pretraining, exhibiting a certain degree of accuracy. To address the complexity arising from the exhaustive enumeration in List-all, we propose the "Top-all" strategy. Initially, CLIP is employed to

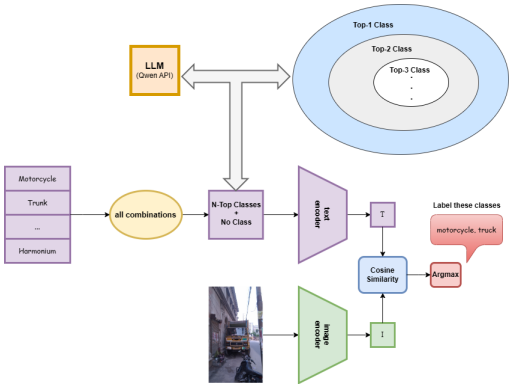

Figure 6. **LLM Optimization** replaces the text input of the previous method with the optimization result of the Qwen LLM.

rank classes, forming a sequence of *[Top-1, Top-2,...,Top-n]*. The combination content is then transformed into combinations of Top-k and all preceding results, as explained in Figure 5. A common sense is that when a label exists, labels with higher probabilities are also likely to exist. Finally, by adding a "nothing" prompt to ensure label completeness, the complexity of the previous method is reduced to $O(n + n + 1) = O(n)$.

(4) LLM Optimization In this approach, we harness the capabilities of large language model (LLM) to optimize the classification prompts by receiving the response of quarying the Qwen-14B API with *"Please caption an image that contains ..."*. Other details are the same as the above and see the change in Figure 6.

3.2. Multi-method to Fine-tune CLIP

If we leave CLIP unadjusted and only change the prompt, there exist some problems. The response to a single-label image classification task is based on contrast, and it's possible for the predicted probabilities of the positive and negative labels to be closely aligned. Therefore, we contemplate modifying the architecture of CLIP by adding a learnable adapter to tailor it for our specific task.

(1) FC-Adapter Following the baseline approach, we simply add a fully connected layer to the fused features after the image and text encoders to transform them into the required label information (See their difference in Figure 7). Specifically, the probability of each label appearing in the image. To ensure a probability distribution within the range of *[0, 1]*, we applied a sigmoid activation function to impose constraints. This method, as compared to the baseline, achieved a remarkable *98.4% mAP* on the Kaggle dataset (See more results at Table 2 and Table 3), demonstrating its efficacy in adapting the model without altering the original parameters.

(2) MLD-Adapter Considering the strong correlation between the structure of FC Adapter and the number of categories in the target dataset, which greatly limits the Zero-shot and Few-shot capabilities of CLIP, inspired by [20], we replace the non-transferable classification header FC-layer with a transferable Multi-label decoder. Specifically, we simply remove the self-attention layer from the Transformer decoder structure, and input text embedding as query and image embedding as key and value into the cross-attention layer. We then average-pooling on the token dimension to obtain output logits, which are then activated by sigmoid function to obtain probability values of *[0, 1]*. The experimental results show that just one layer of MLD block is enough to achieve good results on the target dataset after pretraining. See more details in Figure 8.

Dataset	Classes	Training Set Size	Testing Set Size
Kaggle	10	7843 (Dynamically adjust the train-test ratio)	
VOC-2007	20	5,011 (Train-Val)	4,952 (Test)
MS-COCO	80	82,081 (Train)	40,137 (Val)

Table 1. Dataset Details

4. Experiments

4.1. Setups

Datasets. We train and evaluate our model on three diverse multi-label image classification datasets: Kaggle [21], VOC-2007 [5], and MS-COCO [13]. The Kaggle dataset, sourced from the original task, has been selected with a curated set of 10 classes following human inspection In the zero-shot scenario, the entire dataset is utilized, while in the train-test split mode, 70% is allocated for training and 30% for testing. VOC-2007, a benchmark dataset with 20 classes, includes 5,011 images in the train-validation set and 4,952 images in the test set for evaluation. MS-COCO, a vast dataset with 80 classes, includes 82,081 training and 40,137 validation images. The information of the datasets above is listed in Table 1.

Implementation Details. In replicating the baseline, we reproduce the parameters from the official GitHub repository. For the Kaggle dataset, we align the parameter selection with VOC-2007, and the mAP results are recorded from the best-performing epoch. For the FC-Adapter, aiming to showcase its superiority, we employ the AdamW [15] optimizer and conducted training for only 1 or 2 epochs. The learning rate is set to 0.001, and the most fundamental MSELoss is chosen for the loss function. For the MLD-Adapter, we use one decoder block with four-heads-cross-attention layer and 0.4 dropout feed-forward layer, and employ the AdamW optimizer as well with 180 epochs pretraining on MS-COCO. To investigate the few-shot capabilities of our Adapter, we conduct experiments on the Kaggle dataset with varying training set/test set splits: 2/98, 5/95, 10/90, and 20/80.

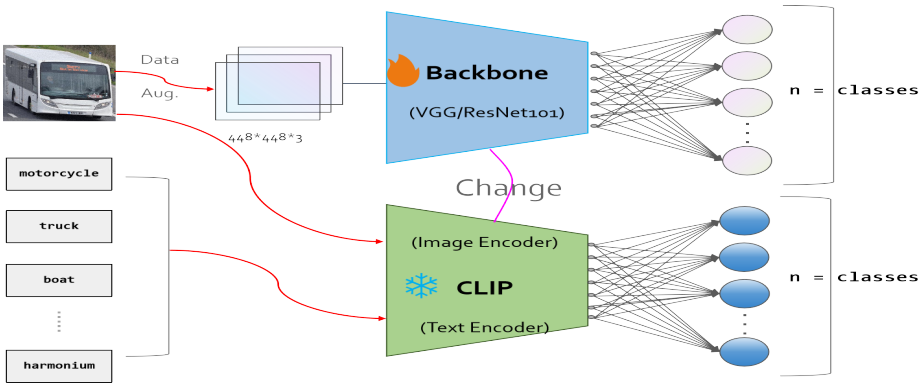

Figure 7. CLIP-FC inherits Baseline's idea, but does not use image preprocessing methods such as data augmentation, and incorporates additional text information.

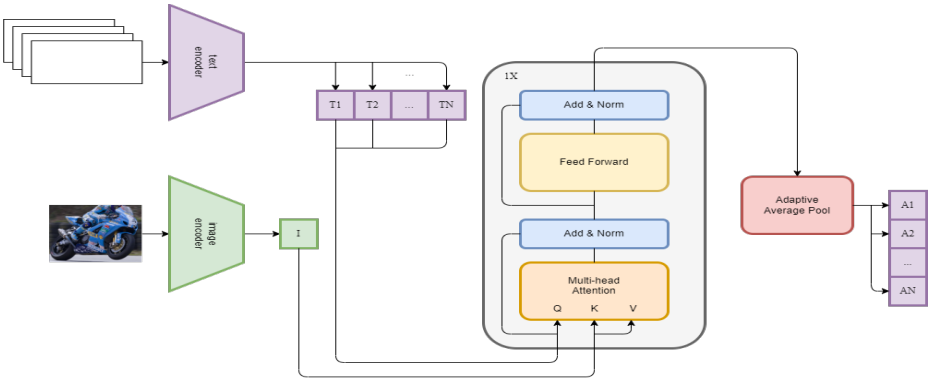

Figure 8. CLIP-MLD simply remove the self-attention layer from the Transformer decoder structure, and input text embedding as query and image embedding as key and value into the cross-attention layer.

Method & Dataset	COCO14-Val	Kaggle-Test (0.3)	VOC2007-Test
VGG	**87.6**	95.4	80.5
ResNet101	85.5	96.4	84.3
CLIP-FC-Adapter	73.2	98.4	85.7
CLIP-MLD-Adapter	—	**99.8**	**92.6**

Table 2. Performance comparison of our Multi-method and baselines in multi-label conventiona learning task (mAP in %) on various datasets

4.2. Multi-label-classification Results

In this section, we conduct experiments to compare various methods proposed in this study against baselines, both internally and externally. Initially, we employ conventional training strategies, utilizing an ample amount of training data distributed across three specified datasets for training and testing. Subsequently, we extend our evaluation to scenarios where CLIP excels, namely in few-shot and zero-shot learning. Notably, the previously mentioned prompt engineering falls within the zero-shot category.

4.2.1 Conventional Learning.

In Table 2, the two adapters we proposed in Kaggle and its similar size of VOC-2007 are better than Baselines' performance. Moreover, CLIP-MLD-Adapter tested in Kaggle dataset can be said to perform surprisingly well, only 0.2% mAP is a full score. Since CLIP-MLD-Adapter will be pre-trained directly on the MS-COCO dataset, it is not appropriate to compare for fairness consideration. And for CLIP-FC-Adapter, we only use a FC-layer, which does not perform well under such large-scale dataset as MS-COCO.

4.2.2 Few-shot Learning.

When it comes to the field of few-shot, we can see that the baselines completely crash in Table 3. But even using a train-test ratio of 2/98, the two adapters easily beat baseline's metrics using the ratio of 70/30. Specially using only a train-test ratio of 20/80, CLIP-MLD-Adapter can also reach more than 99% mAP. The above results show that the adapters we used have excellent performance on the few-shot task.

Method & Train/Test	Kaggle-2/98	Kaggle-5/95	Kaggle-10/90	Kaggle-20/80
VGG-Finetune	11.9	11.9	11.9	11.9
ResNet101-Finetune	11.9	11.9	11.9	11.9
CLIP-FC-Adapter	91.9	94.2	96.7	97.5
CLIP-MLD-Adapter	**96.0**	**98.8**	**98.8**	**99.3**

Table 3. Performance comparison of our Multi-method and baselines in multi-label few-shot learning task (mAP in %) on various datasets. Maybe 11.9 indicates that nothing has been learned.

Method & Dataset	Kaggle	VOC2007-Test
Baseline / CLIP-FC-Adapter	—	—
Yes-or-No	46.3	42.7
List-All Classes	65.5	—
Top-All Classes	65.9	51.2
LLM Optimization	66.0	52.3
CLIP-MLD-Adapter	**75.4**	**90.3**

Table 4. Performance comparison of our Multi-method and baselines in multi-label zero-shot learning task (mAP in %) on various datasets.

4.2.3 Zero-shot Learning.

Our four prompt-based methods show that CLIP has some multi-label image classification capability, and they are gradually improved. But there is still a gap compared to its performance on a single label. CLIP-MLD-Adapter is pre-trained on MS-COCO, so it will have zero-shot capability on Kaggle and VCO-2007, which is a significant performance improvement compared to the previous four methods. It is worth noting that there is not a significant difference in performance between VOC-2007 in few-shot and zero-shot scenarios. However, Kaggle exhibits a noticeable drop, which could be attributed to the distributional differences between Kaggle and MS-COCO.

5. Conclusion

In this paper, we proposed multiple methods to make the CLIP model capable of multi-label image classification, which can be roughly divided into two approaches. The first approach is based on text prompt, while the second involves adding adapters to it. Additionally, we established a basic baseline for conventional learning comparison. In the context of few-shot and zero-shot scenarios, the baseline is clearly incapable of any classification. However, the text prompt-based approach can achieve a mAP of 60% under appropriate trick settings in zero-shot scenarios. In few-shot scenarios, our proposed methods can achieve up to 96% mAP with only 2% of the training set and even over 99% mAP with 20% of the training set. Simultaneously, our approaches have been tested on other datasets, consistently yielding SOTA results. Through our research, it's evident that CLIP stands out as a robust model, showcasing considerable strength in our applications. The introduction of the Adapter should be a pivotal element, demonstrating its paramount role in enhancing CLIP's adaptability and performance across diverse tasks.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2018. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[7] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Xiujun Shu, Bo Ren, and Shu-Tao Xia. Open-vocabulary multi-label classification via multi-modal knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 808–816, 2023. 2

[8] hellowangqian. A baseline for multi-label image classification using ensemble deep cnn. https://github.com/hellowangqian/multi-label-image-classification, 2019. GitHub repository. 2

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[11] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[12] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Uni-modal pre-training for cross-modal tasks. *arXiv preprint arXiv:2002.06353*, 2020. 2

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 4

[14] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. 3

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 4

[16] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 2

[17] Stephen Merity, Bryan McCann, Alex So, Jerry Zheng, Sharan Rajan, Chrisantha Fernando, Jacob Bosboom, Doug Kim, Richard Chen, and Douwe Kiela. Llama: A large language model with abilities in many tasks. *arXiv preprint arXiv:2004.05886*, 2020. 2

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 2

[20] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 32–41, 2023. 2, 4

[21] Meherun Nesa Shraboni. Kaggle multi-label image classification dataset, 2022. 2, 4

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 1

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[26] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1802–1812, 2023. 2

[27] Qian Wang, Ning Jia, and Toby P. Breckon. A baseline for multi-label image classification using an ensemble of deep convolutional neural networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 644–648, 2019. 2

[28] Xuelin Zhu, Jiuxin Cao, Dongqi Tang, Furong Xu, Weijia Liu, Jiawei Ge, Bo Liu, Qingpei Guo, Tianyi Zhang, et al. Text as image: Learning transferable adapter for multi-label classification. *arXiv preprint arXiv:2312.04160*, 2023. 2