arXiv:2406.03071v1 [cs.CV] 5 Jun 2024

EXPLOITING LMM-BASED KNOWLEDGE FOR IMAGE CLASSIFICATION TASKS [*][†]

Maria Tzelepi and Vasileios Mezaris
Information Technologies Institute (ITI)
Centre of Research and Technology Hellas (CERTH)
Thessaloniki, Greece
`{mtzelepi,bmezaris}@iti.gr`

ABSTRACT

In this paper we address image classification tasks leveraging knowledge encoded in Large Multimodal Models (LMMs). More specifically, we use the MiniGPT-4 model to extract semantic descriptions for the images, in a multimodal prompting fashion. In the current literature, vision language models such as CLIP, among other approaches, are utilized as feature extractors, using only the image encoder, for solving image classification tasks. In this paper, we propose to additionally use the text encoder to obtain the text embeddings corresponding to the MiniGPT-4-generated semantic descriptions. Thus, we use both the image and text embeddings for solving the image classification task. The experimental evaluation on three datasets validates the improved classification performance achieved by exploiting LMM-based knowledge.

***K*eywords** Large multimodal models · MiniGPT-4 · Vision language models · CLIP · Embeddings · Representations · Image classification.

1 Introduction

Large Language Models (LLMs) [1, 2], such as GPT-3 [3] and GPT-4 [4], trained on vast amounts of data, have demonstrated exceptional performance in several downstream tasks over the recent few years, placing them squarely at the center of the research activity on Natural Language Processing (NLP) [5] and computer vision [6]. Considering vision recognition downstream tasks, in particular, the emergence of Vision-Language Models (VLMs), such as BLIP-2 [7] and CLIP [8], allowed for connecting image-based vision models with LLMs, forming Large Multimodal Models (LMMs) (also known as Multimodal Large Language Models) [9, 10]. For example, MiniGPT-4 [11] aligns a frozen visual encoder with a frozen LLM using a single projection layer. Besides, it is noteworthy that CLIP, which apart from achieving remarkable zero-shot performance on various downstream tasks through prompting, is a powerful feature extractor, has been extensively used in the recent literature for various applications [12, 13, 14].

In this work, our goal is to address image classification tasks (i.e., tasks of assigning a class label to an image based on its visual content), harnessing the emerging technology of LLMs/LMMs, in order to achieve improved performance in terms of classification accuracy. To achieve this goal, we pursue the direction of utilizing CLIP, proposing to further incorporate knowledge encoded in powerful foundation models such as MiniGPT-4. More specifically, CLIP is commonly utilized as a feature extractor for fitting a linear classifier on the extracted image embeddings and evaluating the performance on various datasets. In this paper, we propose to use MiniGPT-4 for obtaining semantic textual descriptions for each sample of the considered dataset, and then to use the extracted descriptions for feeding them to the textual encoder of CLIP and obtaining the corresponding text embeddings. Thus, instead of using only the image encoder of CLIP to obtain image embeddings, we also use its text encoder to obtain text embeddings that

---

[*] *Citation*: **M. Tzelepi, V. Mezaris, "Exploiting LMM-based knowledge for image classification tasks", Proc. 25th Int. Conf. on Engineering Applications of Neural Networks (EANN/EAAAI 2024), Corfu, Greece, June 2024.**

[†] This preprint has not undergone post-submission improvements or corrections. The Version of Record of this contribution is published in the EANN 2024 Proceedings, CCIS vol. 2141, and is available online at https://doi.org/10.1007/978-3-031-62495-7_13.

capture LLM-extracted knowledge. As we experimentally show, the extracted LMM knowledge is indeed useful for the downstream image classification task, providing increased classification performance.

In the recent literature, as discussed in the subsequent section, there have been some efforts for introducing LLM-based knowledge in image and video classification tasks, using the CLIP vision-language model [8]. However, to the best of our knowledge, this is the first work that utilizes GPT-4 (MiniGPT-4 in particular) in a multimodal fashion for downstream image classification tasks. More specifically, relevant approaches utilize either the unimodal GPT-3 model for textual prompting (e.g., [15]) or the multimodal GPT-4 model again for textual prompting (e.g., [16]). On the contrary, in this work we prompt the MiniGPT-4 model both with text and images. Furthermore, in the literature approaches the LLM is prompted to extract information about the classes of the problem (e.g., [15]). On the contrary, in this paper, we obtain sample-specific information from the LMM.

The rest of the paper is organized as follows. Section 2 briefly presents related works. Section 3 presents in detail the proposed methodology for image classification tasks. Subsequently, Section 4 provides the experimental evaluation of the proposed methodology. Finally, conclusions are drawn in Section 5.

2 Prior Work

In this section we present previous relevant works, involving CLIP and LLMs / LMMs for classification tasks. First, in [17] the authors address a challenge associated with the deployment of VLMs, such as CLIP, that is, identifying the right prompt. To do so, they propose a method, called Context Optimization (CoOp), for adapting such models for downstream image recognition tasks, exploiting recent advances in prompt learning in NLP. Specifically, they model the context words of a prompt with learnable vectors, keeping the entire pretrained parameters frozen. Next, in [18] the authors aim to improve the generalization ability of the aforementioned learned context, by proposing a method called Conditional Context Optimization (CoCoOp). More specifically, they extend CoOp by introducing a lightweight neural network so as to produce an input-conditional token for each image, which is combined with the learnable context vectors.

In another recent work [15], the authors propose a method called Language Guided Bottlenecks (LaBo) building upon concept bottleneck models [19], aiming to develop interpretable-by-design classifiers. To do so, they combine the GPT-3 language model [3] and the CLIP vision-language model. More specifically, they prompt GPT-3 to produce candidate concepts describing each of the problem's classes. A submodular optimization is used then to select a subset of discriminating concepts, which are then aligned to images using the CLIP model. A linear layer is applied on the similarity scores of concepts and images in order to learn weight matrix that represents the importance of each concept in the classification task.

Finally, in [16] the authors utilize the GPT-4 language model [4] for generating text that is visually descriptive. Then they use this information in order to adapt the CLIP model to downstream tasks. In this way, they can achieve considerable improvements in zero-shot CLIP accuracy, compared to the default CLIP prompt. Furthermore, they propose a self-attention based few-shot adapter, called CLIP-A-self, for exploiting this additional information, providing improved performance over the aforementioned CoCoOp method [18]. More specifically, CLIP-A-self learns to select and aggregate the most relevant subset of the visually descriptive text in order to produce more generalizable classifiers.

3 Proposed Method

In this section we present the proposed pipeline, briefly presenting first the two main components involved in the pipeline, i.e., CLIP and MiniGPT-4.

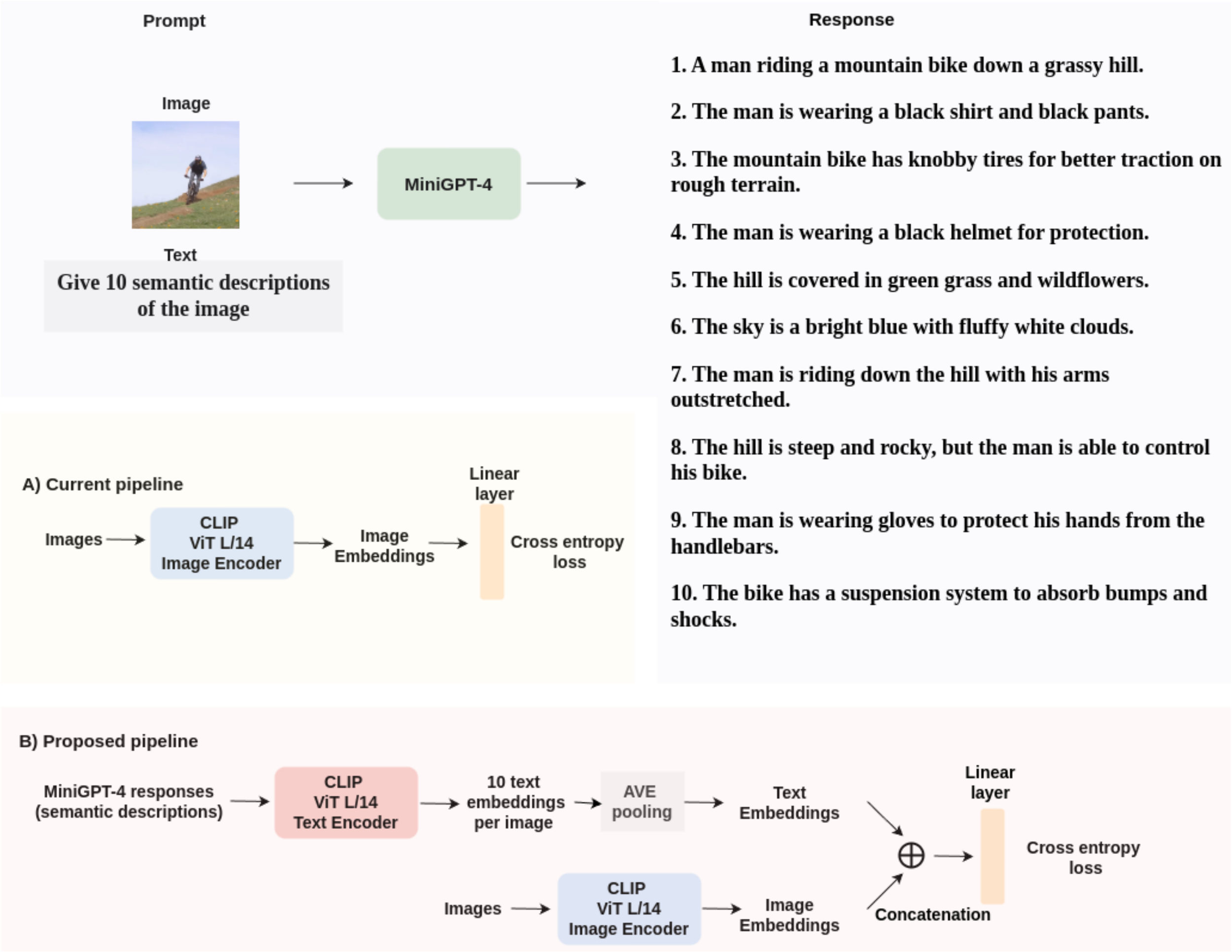
3.1 MiniGPT-4

Following the evolution of LLMs, GPT-4 is the first model to accept both text and image input, producing text output. With the technical details behind GPT-4 remaining undisclosed, MiniGPT-4 was subsequently proposed. MiniGPT-4 aligns a frozen visual encoder with a frozen LLM, utilizing a projection layer. Specifically, MiniGPT-4 uses the Vicuna LLM, while for the visual perception it uses a ViT-G/14 from EVA-CLIP and a Q-Former network. MiniGPT-4 has been utilized in the recent literature for several applications, e.g., [20, 21, 22].

3.2 CLIP

Even though VLMs have been emerged since 2015 [23], CLIP is the first, most popular VLM, trained with natural language supervision at large scale. CLIP comprises of an image encoder and a text encoder, and it is trained with (image, text) pairs for predicting which of the possible (image, text) pairings actually occurred. To do so, it learns a multimodal embedding space by jointly training the image and text encoders to maximize the cosine similarity of the corresponding image and text embeddings of the real pairs in the batch, while minimizing at the same time the cosine similarity of the embeddings of the incorrect pairings. CLIP provides outstanding zero-shot classification performance, using the class labels of the dataset of interest in the text encoder. Another approach to exploit CLIP for classification tasks is to use the CLIP image encoder for extracting the corresponding image representations and use them with a linear classifier (linear probe) achieving exceptional classification performance. Other approaches focus on fine-tuning on a dataset of interest.

3.3 Combining image embeddings and MiniGPT-4 descriptions



Give 10 semantic descriptions of the image

1. A man riding a mountain bike down a grassy hill.
2. The man is wearing a black shirt and black pants.
3. The mountain bike has knobby tires for better traction on rough terrain.
4. The man is wearing a black helmet for protection.
5. The hill is covered in green grass and wildflowers.
6. The sky is a bright blue with fluffy white clouds.
7. The man is riding down the hill with his arms outstretched.
8. The hill is steep and rocky, but the man is able to control his bike.
9. The man is wearing gloves to protect his hands from the handlebars.
10. The bike has a suspension system to absorb bumps and shocks.

Figure 1: Proposed pipeline for image classification tasks. We first prompt the MiniGPT-4 model for obtaining 10 semantic descriptions of each image of the dataset. While the current pipeline (A) uses the image encoder of CLIP to extract the image embeddings, we propose to exploit the knowledge extracted from the MiniGPT-4 (B). To do so, we also extract the text embeddings of the MiniGPT-4 semantic descriptions using the text encoder of CLIP, followed by the average pooling operation. Finally, the two embeddings for each image are concatenated and propagated to a linear layer for performing the classification task.

In this paper, we aim to exploit knowledge encoded in the MiniGPT-4 model for achieving advanced classification performance in terms of accuracy. To do so, as illustrated in Fig. 1, we first use the MiniGPT-4 for obtaining semantic descriptions for each image of the dataset. That is, we prompt the model with the image along with the text "*Give 10 semantic descriptions of the image*". Subsequently, we introduce the acquired responses to the CLIP's textual encoder (i.e., 10 semantic descriptions for each image of the dataset) and obtain the text embeddings at the final layer of the text encoder. Next, a pooling operation is applied on the aforementioned text embeddings, resulting to a unique embedding for each image. We finally use these embeddings along with the corresponding image embeddings, obtained from the image encoder of CLIP, to feed a single linear layer with a cross entropy loss for performing classification. That is, instead of simply using the image embeddings from the CLIP's image encoder to perform classification, we additionally use the CLIP's text embeddings that capture LMM-based knowledge.

More specifically, let $\mathcal{X} = \{\mathbf{X}_i \in \Re^{h \times w \times c} | i = 1, \dots, N\}$ be a set of N images, where h, w, c denote the height, width, and channels of the image respectively. Each image $\mathbf{X}_i$ is linked with a class label l_i. Considering a classification task, a common approach is to use the CLIP image encoder f in order to obtain for each image $\mathbf{X}_i$ the corresponding image embedding, $\mathbf{y}_i = f(\mathbf{X}_i) \in \Re^d$, where d corresponds to the dimension of the embedding. Then, the image embeddings are used to fit a linear classifier.

Instead, we use MiniGPT-4 for obtaining semantic descriptions, that is we consider for each image $\mathbf{X}_i$ the MiniGPT-4-generated responses, $d_i^j, j = 1, \dots, 10$, and then we extract the text embeddings using the CLIP text encoder g. That is, $\{\mathbf{t}_i^j = g(d_i^j) \in \Re^d | j = 1, \dots, 10\}$. Subsequently, the average pooling operation is applied and a unique text embedding $\mathbf{t}_i^M \in \Re^d$ is obtained for each image $\mathbf{X}_i$. For combining the image embeddings with the corresponding text embeddings and realizing the classification task, we generally use concatenation, that is $[\mathbf{y}_i^\intercal, \mathbf{t}_i^{M\intercal}]^\intercal \in \mathbb{R}^{d+d}$. For example, in our experiments we use CLIP with ViT-L-14, were the dimensions of both the image and text embeddings is 768, i.e., $d = 768$. The mean embedding, $\mathbf{m}_i \in \Re^{768}$, instead of the concatenated, is also explored. Finally, a learnable linear layer processes the concatenated embeddings to predict the class labels, using the cross entropy loss.

4 Experiments

In this section we present the experiments conducted in order to evaluate the proposed methodology for image classification. First, the utilized datasets are presented followed by the evaluation metrics. Subsequently, the implementation details are presented followed by the experimental results.

4.1 Datasets

We use three datasets to evaluate the effectiveness of the proposed methodology, i.e., UCF-101 [24], Event Recognition in Aerial videos (ERA) [25], and Biased Action Recognition (BAR) [26]. UCF-101 is an action recognition dataset, containing 101 classes. We derive the middle frame from each video and we follow the same protocol as in [24] for forming the training and test sets of the 13,320 extracted frames. ERA is an event recognition dataset in unconstrained aerial videos. It consists of 1,473 training images and 1,391 test images, divided into 25 classes. BAR is a real-world image dataset with six action classes which are biased to distinct places, consisting of of 1,941 training images and 654 test images.

4.2 Evaluation Metrics

We use test accuracy to evaluate the performance of combining LMM-based features with image features for image classification tasks. Qualitative results of the MiniGPT-4 responses are also provided.

4.3 Implementation Details

As previously described, MiniGPT-4 is used as the source of LLM-based knowledge. We use MiniGPT-4 with Vicuna-13B locally. For image and feature extraction we use the ViT-L-14 CLIP version. For the classification task, a single linear layer is used, with output equal to the number of classes of each dataset. Models are trained for 500 epochs, and the learning rate is set to 0.001. Experiments are conducted using the Pytorch framework on an NVIDIA GeForce RTX 3090 with 24 GB of GPU memory.

4.4 Experimental Results

In this section we present the experimental results of the proposed methodology for addressing image classification tasks. We first provide the evaluation results of the proposed methodology of using both CLIP image embeddings and the CLIP text embeddings that reflect the MiniGPT-4 descriptions for the classification task, against the baseline approach of merely using the CLIP image embeddings, in Table 1. Best performance is printed in bold. As demonstrated, combining LMM knowledge with image embeddings provides considerably advanced classification performance in all the utilized datasets. Correspondingly, in Fig.s 2-4 we provide the test accuracy throughout the training epochs for the two compared approaches on the three utilized datasets, where the steadily advanced performance of the proposed approach is validated.

Table 1: Test accuracy.

Method	**UCF-101**	**ERA**	**BAR**
CLIP (image) [8]	89.981	84.472	94.801
CLIP (image & MiniGPT-4 descriptions)	**91.753**	**85.909**	**95.566**

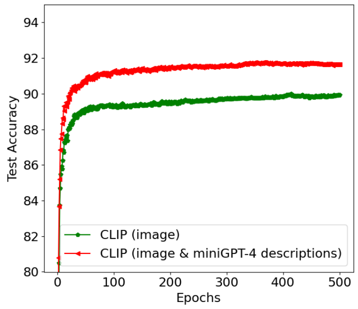


Figure 2: UCF dataset: Test accuracy throughout the training epochs.

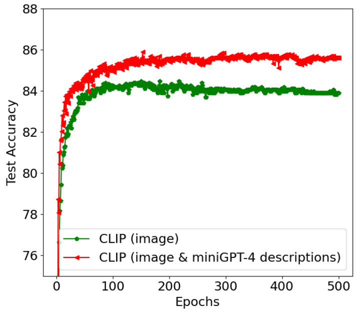


Figure 3: ERA dataset: Test accuracy throughout the training epochs.

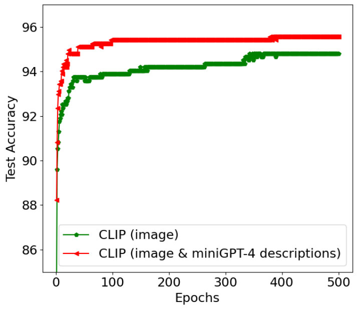


Figure 4: BAR dataset: Test accuracy throughout the training epochs.

Subsequently, an ablation study is performed on the UCF-101 dataset. We present in Table 2 the evaluation results. More specifically, we first evaluate the performance of each of the components individually. That is, the classification performance using only the CLIP image embeddings is firstly presented, which serves as the main comparison approach. Evidently, this approach provides exceptional performance. Next, we present the classification performance using only the text embeddings. That is, we only use the MiniGPT-4 knowledge to represent the images of the dataset. This approach performs, as expected, worse, however it achieved remarkable performance. Subsequently, we use both the image and text CLIP embeddings, concatenated. As it is shown, using the proposed methodology we can accomplish significant improvement over the approach of using only the image embeddings. Besides, as it was previously mentioned, we also explore the mean embedding of image and text embeddings instead of the concatenated, which as it is demonstrated also considerably improves the baseline performance, being however slightly worse as compared to the concatenation approach. The aforementioned remarks are validated through Fig. 5, where the test accuracy throughout the training epochs is provided for all the compared approaches.

Table 2: UCF-101: Ablation study.

Method	**Test Accuracy**
CLIP (image) [8]	89.981
CLIP (MiniGPT-4 descriptions)	80.333
CLIP (image & MiniGPT-4 descriptions - CONCAT)	**91.753**
CLIP (image & MiniGPT-4 descriptions - MEAN)	**91.277**

Table 3: UCF-101: Comparisons with CLIP-based approaches.

Method	Test Accuracy
CLIP-zero shot [8]	76.20
CLIP-A-self [16]	84.1
CoOp [17]	84.69
CoCoOp [18]	82.33
LaBo [15]	90.67
CLIP - Linear Probe [15]	90.11
CLIP (image) [8]	89.981
CLIP (image & MiniGPT-4 descriptions)	**91.753**

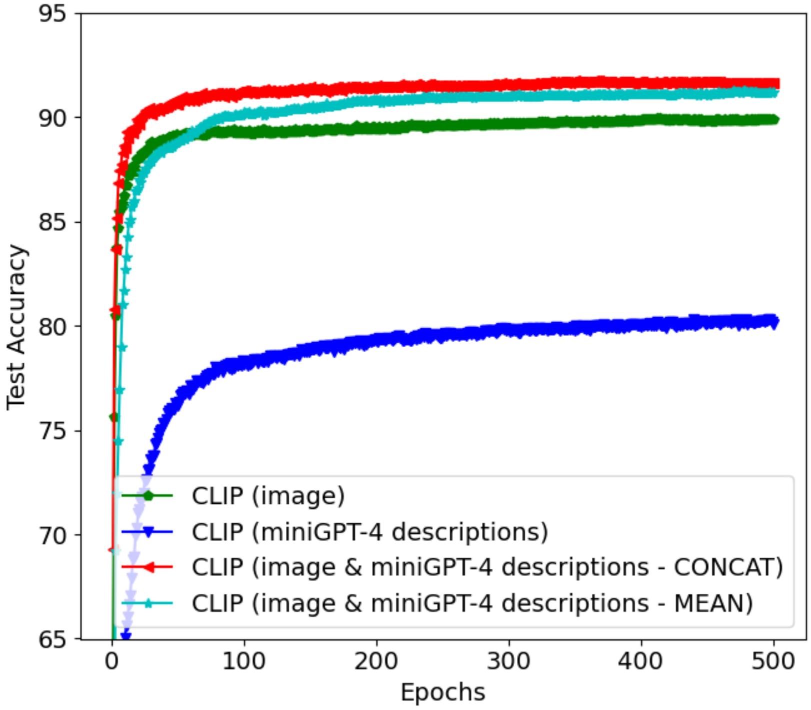
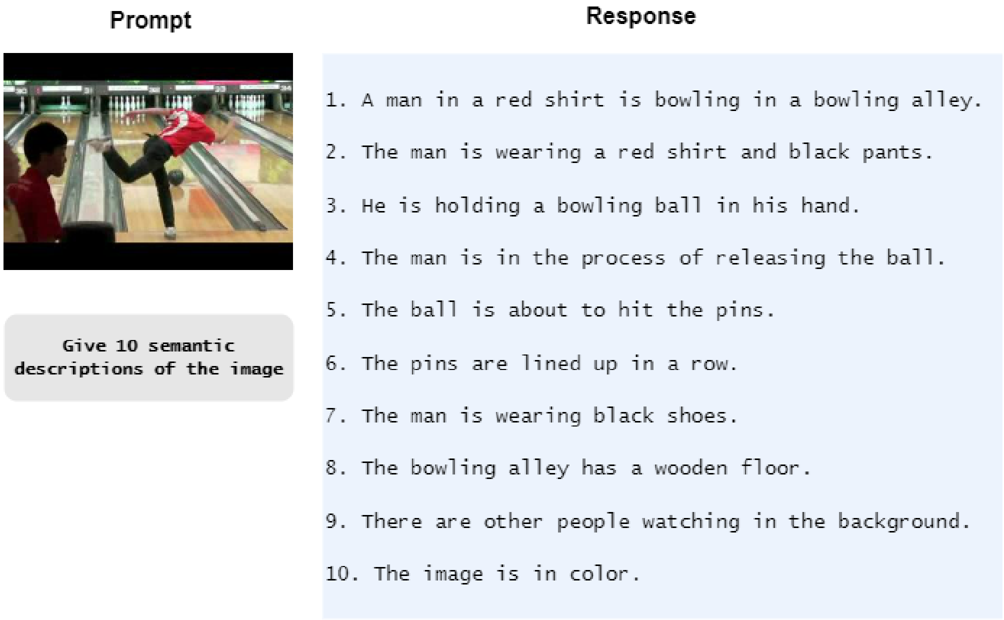


Figure 5: Ablation - UCF-101 dataset: Test accuracy throughout the training epochs.

In Table 3 we present the comparisons of the proposed methodology against state-of-the-art methods. More specifically, the most relevant and straightforward comparison is against the CLIP model utilizing only the image emebeddings. However, we also include relevant works that include CLIP and LLMs/LMMs. All the models, as described in the Prior Work Section, include training. Finally, even though we do not proceed in a zero-shot approach, we also report the CLIP zero-shot classification performance for completeness. As it is demonstrated the proposed approach achieves the best performance.

Finally, in Figs. 6-9 we provide some qualitative results of the knowledge encoded in the MiniGPT-4 model. More specifically, we present the responses of the model, prompted with the image and the text "*Give 10 semantic descriptions of the image*". As it can be observed, accurate and meaningful descriptions are generally obtained.



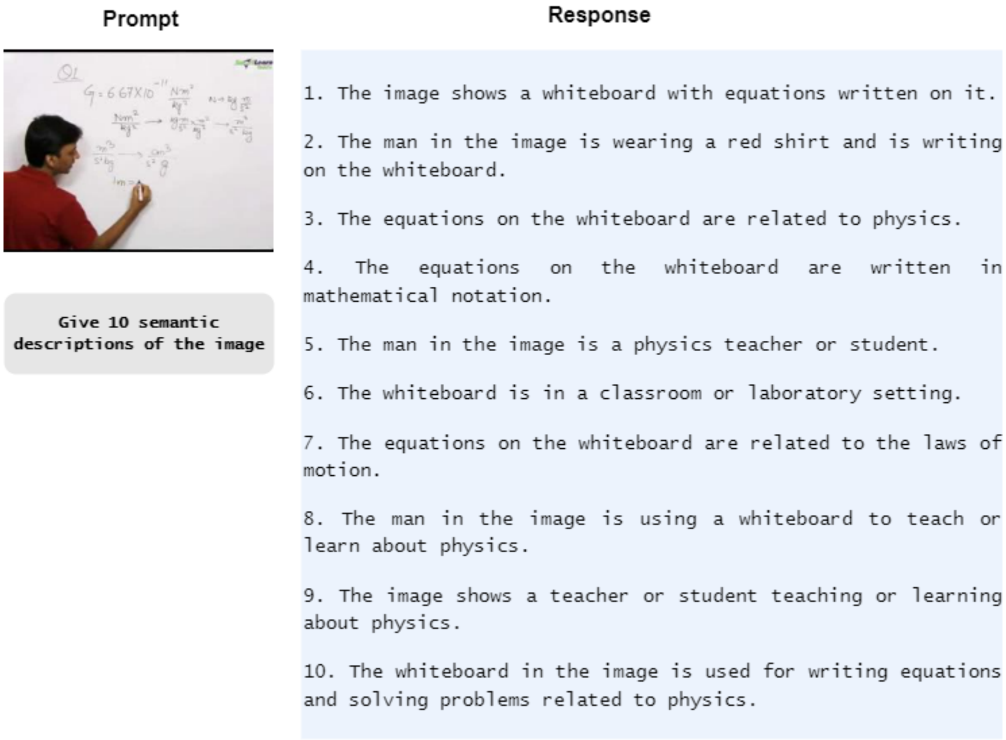
Prompt

Give 10 semantic descriptions of the image

Response

1. A man in a red shirt is bowling in a bowling alley.
2. The man is wearing a red shirt and black pants.
3. He is holding a bowling ball in his hand.
4. The man is in the process of releasing the ball.
5. The ball is about to hit the pins.
6. The pins are lined up in a row.
7. The man is wearing black shoes.
8. The bowling alley has a wooden floor.
9. There are other people watching in the background.
10. The image is in color.

Figure 6: Example of MiniGPT-4 response for a test image of the class *Bowling*.



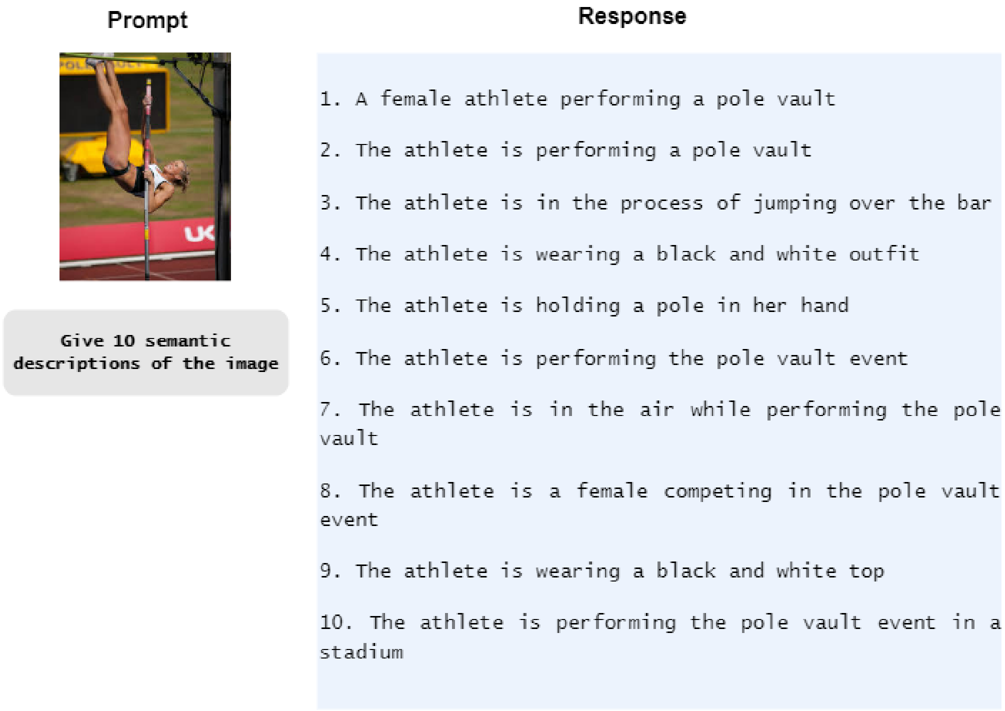
Prompt

Give 10 semantic descriptions of the image

Response

1. The image shows a whiteboard with equations written on it.
2. The man in the image is wearing a red shirt and is writing on the whiteboard.
3. The equations on the whiteboard are related to physics.
4. The equations on the whiteboard are written in mathematical notation.
5. The man in the image is a physics teacher or student.
6. The whiteboard is in a classroom or laboratory setting.
7. The equations on the whiteboard are related to the laws of motion.
8. The man in the image is using a whiteboard to teach or learn about physics.
9. The image shows a teacher or student teaching or learning about physics.
10. The whiteboard in the image is used for writing equations and solving problems related to physics.

Figure 7: Example of MiniGPT-4 response for a test image of the class *Writing on board*.



Give 10 semantic descriptions of the image

1. A female athlete performing a pole vault

2. The athlete is performing a pole vault

3. The athlete is in the process of jumping over the bar

4. The athlete is wearing a black and white outfit

5. The athlete is holding a pole in her hand

6. The athlete is performing the pole vault event

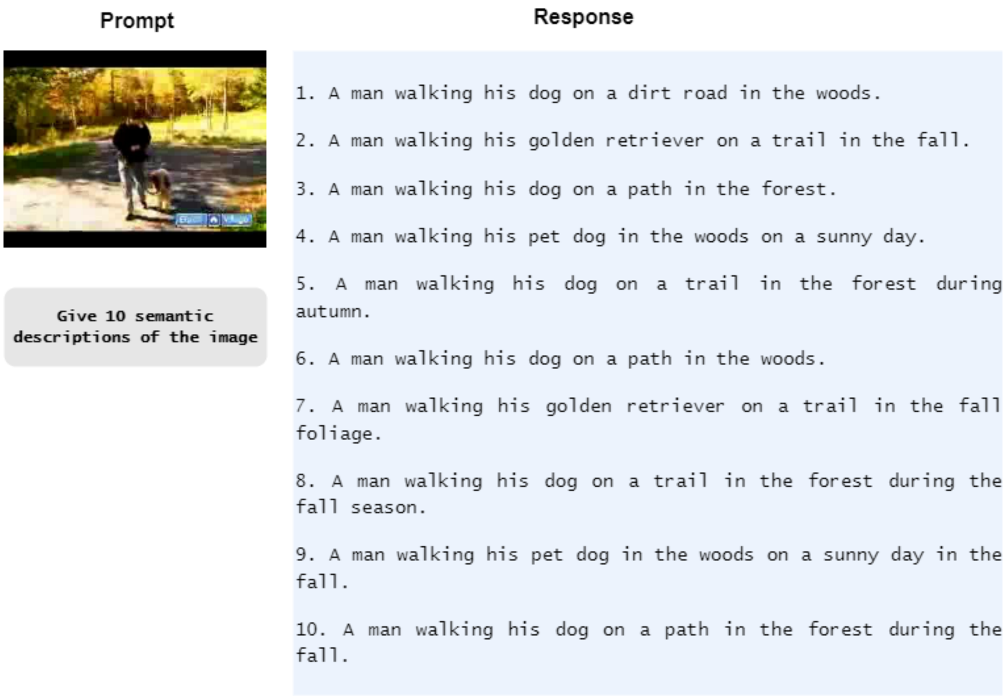
7. The athlete is in the air while performing the pole vault

8. The athlete is a female competing in the pole vault event

9. The athlete is wearing a black and white top

10. The athlete is performing the pole vault event in a stadium

Figure 8: Example of MiniGPT-4 response for a test image of the class *Pole vaulting*.



Give 10 semantic descriptions of the image

1. A man walking his dog on a dirt road in the woods.

2. A man walking his golden retriever on a trail in the fall.

3. A man walking his dog on a path in the forest.

4. A man walking his pet dog in the woods on a sunny day.

5. A man walking his dog on a trail in the forest during autumn.

6. A man walking his dog on a path in the woods.

7. A man walking his golden retriever on a trail in the fall foliage.

8. A man walking his dog on a trail in the forest during the fall season.

9. A man walking his pet dog in the woods on a sunny day in the fall.

10. A man walking his dog on a path in the forest during the fall.

Figure 9: Example of MiniGPT-4 response for a test image of the class *Walking with dog*.

5 Conclusions

In this paper we deal with image classification tasks exploiting knowledge encoded in LMMs. Specifically, we use the MiniGPT-4 model to extract sample-specific semantic descriptions. Subsequently, we use these description for obtaining the text embeddings from the CLIP's text encoder. Then, we use the aforementioned text embeddings along with the corresponding image embeddings obtained from the CLIP's image encoder for the classification task. As it is experimentally validated, incorporating LMM-based knowledge to the image embeddings achieves considerably improved classification performance.

Acknowledgements

This work was supported by the EU Horizon Europe program, under grant agreement 101070109 (TransMIXR).

References

[1] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[2] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[5] Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Xing Wang, Shuming Shi, and Zhaopeng Tu. Parrot: Translating during chat using large language models. *arXiv preprint arXiv:2304.02426*, 2023.

[6] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 910–919, 2024.

[7] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[8] 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.

[9] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

[10] Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024.

[11] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[12] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6978–6988, 2023.

[13] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567. IEEE, 2022.

[14] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *arXiv preprint arXiv:2303.14613*, 2023.

[15] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023.

[16] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 262–271, 2023.

[17] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[18] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[19] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

[20] Dana Aubakirova, Kim Gerdes, and Lufei Liu. Patfig: Generating short and long captions for patent figures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2843–2849, 2023.

[21] Juexiao Zhou, Xiaonan He, Liyuan Sun, Jiannan Xu, Xiuying Chen, Yuetan Chu, Longxi Zhou, Xingyu Liao, Bin Zhang, and Xin Gao. Skingpt-4: An interactive dermatology diagnostic system with visual large language model. 2023.

[22] Zhengqing Yuan, Huiwen Xue, Xinyi Wang, Yongming Liu, Zhuanzhe Zhao, and Kun Wang. Artgpt-4: Artistic vision-language understanding with adapter-enhanced minigpt-4. *arXiv preprint arXiv:2305.07490*, 2023.

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[25] L. Mou, Y. Hua, P. Jin, and X. X. Zhu. ERA: A dataset and deep learning benchmark for event recognition in aerial videos. *IEEE Geoscience and Remote Sensing Magazine*, in press.

[26] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.