A Novel Framework for Automated Explain Vision Model Using Vision-Language Models

Anonymous ACL submission

Abstract

The development of many vision models mainly focuses on improving their performance using metrics such as accuracy, IoU, and mAP, with less attention to explainability due to the complexity of applying xAI methods to provide a meaningful explanation of trained models. Although many existing xAI methods aim to explain vision models sample-by-sample, methods explaining the general behavior of vision models, which can only be captured after running on a large dataset, are still underexplored. Furthermore, some other xAI methods are complex and require expert interpretation, limiting their use in causal vision model development despite the importance of explainability. With the application of Vision-Language Models, this paper proposes a pipeline to explain vision models for both sample and dataset levels. The proposed pipeline can be applied to discover failure cases and understand vision models without much effort, thus it can integrate vision models' development and xAI analysis to advance the development of image analysis.

1 Introduction

Understanding how vision models make decisions is important to improve the reliability and trustworthiness of AI systems. Although there are many established methods, benchmarks for evaluating the overall performance of vision models on large datasets, methods focusing on analyzing how models understand images, especially on large image datasets, are still limited despite the importance of explainability in providing information about how and why the model fails in some scenarios. Consequently, a scalable pipeline to explain vision models in one sample or a large vision dataset would be important for image processing development.

xAI methods such as CAM, GradCAM, LIME, and TCAV are introduced to explain vision models. While concept-based methods like TCAV, ACE (Ghorbani et al., 2019), and CRAFT (FEL et al., 2022) explain a model on a dataset, they only explain on a dataset and require more knowledge about xAI, limiting their application. Meanwhile, CAM-based methods are quick and simple; applying those methods to a dataset would require manual summarization to get useful information. An existing framework, LangXAI, uses VLM to describe vision models' attention. Although the process is automatic, the pipeline has similar limitations on large datasets, and the pipeline's explanation might not be more useful than saliency images.

This work includes three main contributions. First, we propose a scalable pipeline that combines CAM-based methods with VLMs to explain the behavior of the vision model. Second, we propose masked CAM images, which show the benefit of understanding the attended regions of vision models in this study's scope. Lastly, we introduce a confusion matrix used in the pipeline, which helps summarize models' behavior on a large dataset, providing a general understanding of the models.

2 Related Work

Although many frameworks focus on evaluating vision model performance with metrics like accuracy, IoU, ensuring transparency and interpretability through explainable AI (xAI) is also crucial (Gunning and Aha, 2019; Zhao et al., 2015). xAI includes a variety of techniques to make machine learning models more interpretable and is generally classified as model-agnostic and model-specific methods (Lundberg and Lee, 2017). Model-agnostic approaches, applicable to any model, often assess the importance of features, while model-specific methods leverage internal model structures for explanation (Bach et al., 2015). For vision tasks, popular techniques such as LIME (Ribeiro et al., 2016), TCAV (Kim et al., 2018), and CAM-based methods, including CAM (Zhou et al., 2016), Grad-CAM (Sel-

varaju et al., 2017), Grad-CAM++(Chattopadhyay et al., 2017), LayerCAM (Jiang et al., 2021), Score-CAM (Wang et al., 2020a), EigenCAM (Muhammad and Yeasin, 2020), and XGradCAM (Oquab et al., 2015; Wang et al., 2020b) highlight regions important for predictions (Itti et al., 1998; Kümmerer et al., 2014; Zhao et al., 2015). These tools are especially valuable in fields like healthcare (Borys et al., 2023; Kakogeorgiou and Karantzalos, 2021; Kim and Joe, 2022), although many still require expert interpretation, which poses challenges to integration into development workflows.

The development of Vision-Language Models (VLMs) expands the capabilities of LLMs such as Qwen (Bai et al., 2023), Llama (Touvron et al., 2023), and Phi (Li et al., 2023b) by enabling them to process images and text simultaneously (Ranasinghe et al., 2024; Liu et al., 2023). VLMs use vision models such as CLIP (Radford et al., 2021) to excel in multimodal tasks. Prominent examples include Flamingo (Alayrac et al., 2022), BLIP (Li et al., 2022) integrates a visual encoder with an LLM via a querying transformer (Li et al., 2023a), and different VLMs such as GPT-4o, Qwen-VL (Bai et al., 2023), and Llama Vision (Chu et al., 2024), show strong ability to understand visual data. Consequently, they are widely used in many applications, including evaluating existing vision models (Chen et al., 2024).

Despite the importance of xAI and the significant advancement in VLMs in recent years, the applications to analyze interpretive visualizations, such as Grad-CAM, in visual models remain underexplored. To fill this gap, LangXAI Nguyen et al. (2024) explored the potential of using VLMs to generate explanations for visual recognition based on the intensity of colors extracted from CAM methods. However, the framework generates a description for one sample at a time without summarizing, evaluating, and comparing the general interpretability of models on a set of images, making it difficult to understand their general underlying features and behaviors, as we cannot just read many descriptions for each model. To further bridge this gap, we developed a scalable pipeline that utilizes VLMs to evaluate predictions from vision models, scoring them, providing detailed explanations, and summarizing the model's attention with a confusion matrix on a larger dataset. This method overcomes previous work by providing quantitative results on a larger dataset, helping to generalize the use of xAI and better connect training to understanding.

3 Methodology

We introduce a novel pipeline to explain vision models automatically. This pipeline combines CAM methods to visualize the model's attention and uses vision-language models to generate descriptions, evaluations, scores, and a confusion matrix. The entire proposed pipeline to explain and score vision models is illustrated in Figure 1.

3.1 Masked CAM image

The pipeline starts by feeding an image to vision models and getting a predicted result on the image. After that, different methods to extract models' attention, including CAM, LayerCAM, and more, are utilized to get an attention map of vision models on the image. Then, we apply a more general version of the *sigmoid* function to the attention map and get a mask for each image. The activation function is illustrated in Equation 1, where v_{xy}, ranging from 0 to 1, is the value of the attention map at position (x, y), indicating the importance of the pixel, and M_{xy} is the activated value at position (x, y).

$$M_{xy} = \frac{1}{1 + \exp\left(\alpha \cdot (\beta - v_{xy})\right)} \tag{1}$$

In the equation, the values of $v_{xy} > \beta$ are scaled closer to 1 to highlight important regions, while $v_{xy} < \beta$ gradually decrease toward 0, reflecting reduced importance. Meanwhile, α controls the transition speed. The higher α, the more sudden the transition from blacked-out to visible.

After achieving the mask, we apply it to the original image to hide regions with less attention according to the CAM-based method. This process is formulated in Equation 2, where we multiply each pixel in the original image I by the corresponding value in the calculated mask M in Equation 1 to achieve the final masked image A.

$$A_{xy} = I_{xy} \cdot M_{xy} \tag{2}$$

The main reason we use the masked image instead of the heatmap overlay to explain the vision model's attention is to prevent degrading the quality of the image, which can negatively affect the results of VLMs. Using a heatmap overlay can hide away important features of the object(s), thus reducing VLM's ability to understand the attention regions and lowering its accuracy. By blacking out the areas without the model's focus and maintaining the remaining areas, we will not sacrifice the image quality on attended objects while ensuring

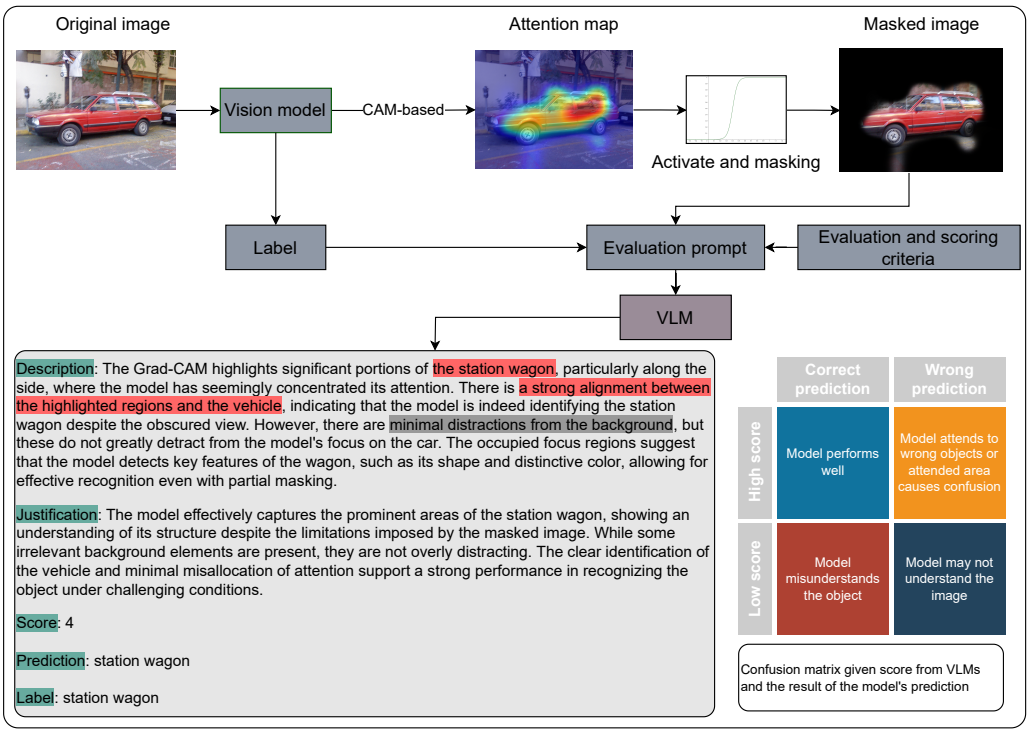

Figure 1: The pipeline evaluates vision models' ability to understand an image. The VLM model can describe, justify, and score the input image and the corresponding attention map. In the description, the model's interpretation of positive objects is highlighted in red, while gray illustrates the negative description.

that VLMs can only see and focus on the main regions they need for the evaluation. Furthermore, the attention of the vision model should justify and provide sufficient evidence to explain its prediction. The lack of evidence to recognize and distinguish objects in the attended regions might suggest an existing problem with the vision model.

3.2 VLM assessment

The result of the previous process is an image that is largely blacked out, except for areas the model considered important in its output. The masked image and the predicted label of the model are then fed to a VLM for evaluation and scoring. In the pipeline, VLMs are asked to find the relevance between the vision model's prediction and the visible object(s) in the masked image, and then explain further. Finally, VLMs score every pair of masked pictures and labels to quantify the model's ability.

3.3 Evaluation metrics

This section defines a confusion matrix for this pipeline, as we have a label and a generated explanation score for each image. First, we will select a threshold score to decide which generated score shows that the vision model has a problem in understanding images. After that, we build the matrix as shown in Figure 1, which depends on the VLM scores and the correctness of the model on each sample. The proposed confusion matrix shows four stages of the model:

- Correct: The model focuses on the correct object and predicts the object correctly, indicating a strong understanding of the image.
- Misunderstood object: The model prediction is correct, but its attention does not align with the object, indicating a misunderstanding of the appearance of the target.
- Attend to wrong object: The model's attention is correct, but its prediction is wrong, showing that the model focuses on another object, not the labeled one.
- Lack of understanding: The model cannot explain its attention and its prediction is incorrect, showing that the model does not have enough knowledge for the task.

Given many input samples, we can count and compute the percentage of each stage and obtain a comprehensive review of the model.

4 Experiment

We evaluated the pipeline's trustworthiness with four experiments to assess the VLMs' output (descriptions, scores), hyperparameter selection, and the usage of masked CAM and CAM images. The last one assesses our confusion matrix in predicting problems of trained vision models. The scoring system ranges from zero (random attention) to five (perfect attention), and saliency maps are extracted from the last layer as in the GradCAM paper.

4.1 Human evaluation

The first experiment compares the pipeline with two authors by collecting 200 ImageNet (Russakovsky et al., 2015) images, using ResNet18, MaxViT and GradCAM to extract saliency maps, manually scoring, and taking average scores. Then those scores are compared with the VLM scores using the Pearson correlation (PC). The results show that when using masked CAM images, the correlation between GPT-4o-mini and the annotators is 0.54, and between the Gemini-1.5-flash and the annotators is 0.50. Meanwhile, GPT-4o-mini can achieve 0.53 in Pearson's correlation when using original CAM images, while Gemini can achieve 0.41, lower than masked CAM images. Lastly, the correlation between the two annotators is 0.71. Table 1 shows the Pearson correlation (PC).

Next, two authors checked the VLMs' output (200 samples) to verify the quality of the generated text for CAM and masked CAM images. In this experiment, they read the VLMs' output and decide whether those texts are acceptable. An output is unacceptable if the VLMs provide incorrect information, do not match the predicted object, or the score is not aligned with the justification and description. The result shows that 85.5% of the GPT-4o-mini's generated samples on the masked CAM images are correct, while this rate in the Gemini-1.5-flash is 79.5%. Meanwhile, results on the original CAM image show a lower rate; Gemini-1.5-flash achieves 54.5% and GPT-4o-mini achieves 75.5%. This indicates that Gemini-1.5-flash benefits more from using masked CAM images than GPT-4o-mini. The experiment result is reported in Table 1, denoted by AR, which is short for acceptance rate.

The third experiment, which uses the same method and data as the first experiment, measures the framework-human correlation with different hyperparameters. The result in Table 2 shows that the combination of $\alpha = 25, \beta = 0.6$ achieves the highest correlation with 0.64, and all selected combinations are better than using the original CAM, which does not have hyperparameters. For this experiment, only Gemini-1.5-flash is used.

	Gemini-1.5-flash	GPT-4o-mini
Masked image	0.50 - 79.5%	0.54 - 85.5%
CAM image	0.41 - 54.5%	0.53 - 75.5%

Table 1: Comparison between masked CAM images ($\alpha = 25, \beta = 0.4$) and CAM image. The results are shown as $PC - AR$, PC is the Pearson correlation between VLMs' scores and humans' scores, and AR is the acceptance rate of VLMs' generated text.

	$\alpha = 25$ $\beta = 0.4$	$\alpha = 15$ $\beta = 0.6$	$\alpha = 25$ $\beta = 0.7$
Masked CAM	0.50	0.64	0.63
Original CAM		0.41	

Table 2: Framework-human correlation results (Pearson Correlation) for Gemini-1.5-flash using different hyperparameters and CAM types.

4.2 Failed models evaluation

We trained 31 models to classify cats and dogs (Elson et al., 2007) in two scenarios: normal training and training with cat images marked by a red dot on the top right, introducing a biased attention mechanism. Further examples of the training datasets are provided in Section A.2. We then collected 20 images from the training set and computed the confusion matrix as proposed for each model. Next, we determined the percentage of incorrect predictions err (wrong predictions or low VLM scores). The correlation between this percentage and the type of training (normal or biased) is -0.70, indicating that the higher err, the more likely the model is trained on the biased dataset. This experiment shows the trustworthiness of the proposed matrix in understanding and detecting models' problems.

5 Conclusion

This paper proposed a novel framework to integrate CAM visualizations with VLM to explain vision models. The pipeline can be easily integrated into the evaluation process to provide more details, including text-based explanations, scores, and a confusion matrix. This pipeline's specialty is that it can provide assessments for both the sample-level and dataset-level, allowing researchers to understand the general and detailed model's performance.

Limitations

Despite being scalable and helpful in detecting scenarios where vision models behave incorrectly, the pipeline still contains some limitations, including the dependence on VLM and the quality of the prompt to generate a correct description with a suitable score for each sample. Furthermore, the pipeline only utilizes CAM-based methods (and RISE, as we can extract attention regions from them) to extract the attention regions, but not methods like finding the decision boundary and some other xAI visualization techniques.

Potential risk

The quality of the generated descriptions is highly dependent on the performance of the VLM. Therefore, the pipeline should be used only as a supporting tool, with the researcher remaining the primary decision maker in the analysis.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenhang Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, K. Lu, and 31 others. 2023. Qwen technical report. *ArXiv*, abs/2309.16609.

Katarzyna Borys, Yasmin Alyssa Schmitt, Meike Nauta, Christin Seifert, Nicole Krämer, Christoph M Friedrich, and Felix Nensa. 2023. Explainable ai in medical imaging: An overview for clinical practitioners–saliency-based xai approaches. *European journal of radiology*, 162:110787.

Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. 2017. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6562–6595. PMLR.

Xiangxiang Chu, Jianlin Su, Bo Zhang, and Chunhua Shen. 2024. Visionllama: A unified llama backbone for vision tasks. In *European Conference on Computer Vision*, pages 1–18. Springer.

Jeremy Elson, John (JD) Douceur, Jon Howell, and Jared Saul. 2007. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc.

Thomas FEL, Agustin Martin Picard, Louis Béthune, Thibaut Boissin, Julien Colin, David Vigouroux, Remi Cadene, and Thomas Serre. 2022. CRAFT: explaining using concepts from recursive activation factorization.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32.

David Gunning and David Aha. 2019. Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58.

Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259.

Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. 2021. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888.

Ioannis Kakogeorgiou and Konstantinos Karantzalos. 2021. Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 103:102520.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and 1 others. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Hong-Sik Kim and Inwhee Joe. 2022. An xai method for convolutional neural networks in self-driving cars. *PLoS one*, 17(8):e0267282.

Matthias Kümmerer, Lucas Theis, and Matthias Bethge. 2014. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

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.

Yuan-Fang Li, Sébastien Bubeck, Ronen Eldan, Allison Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. *ArXiv*, abs/2309.05463.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Mohammed Bany Muhammad and Mohammed Yeasin. 2020. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE.

Truong Thanh Hung Nguyen, Tobias Clement, Phuc Truong Loc Nguyen, Nils Kemmerzell, Van Binh Truong, Vo Thanh Khang Nguyen, Mohamed Abdelaal, and Hung Cao. 2024. Langxai: Integrating large vision models for generating textual explanations to enhance explainability in visual perception tasks. *arXiv preprint arXiv:2402.12525*.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2015. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S Ryoo, and Tsung-Yu Lin. 2024. Learning to localize objects improves spatial reasoning in visual-llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12977–12987.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

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.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020a. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 111–119.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020b. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25.

Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2015. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1265–1274.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929.

A Appendix

A.1 Examples of Model's Evaluation

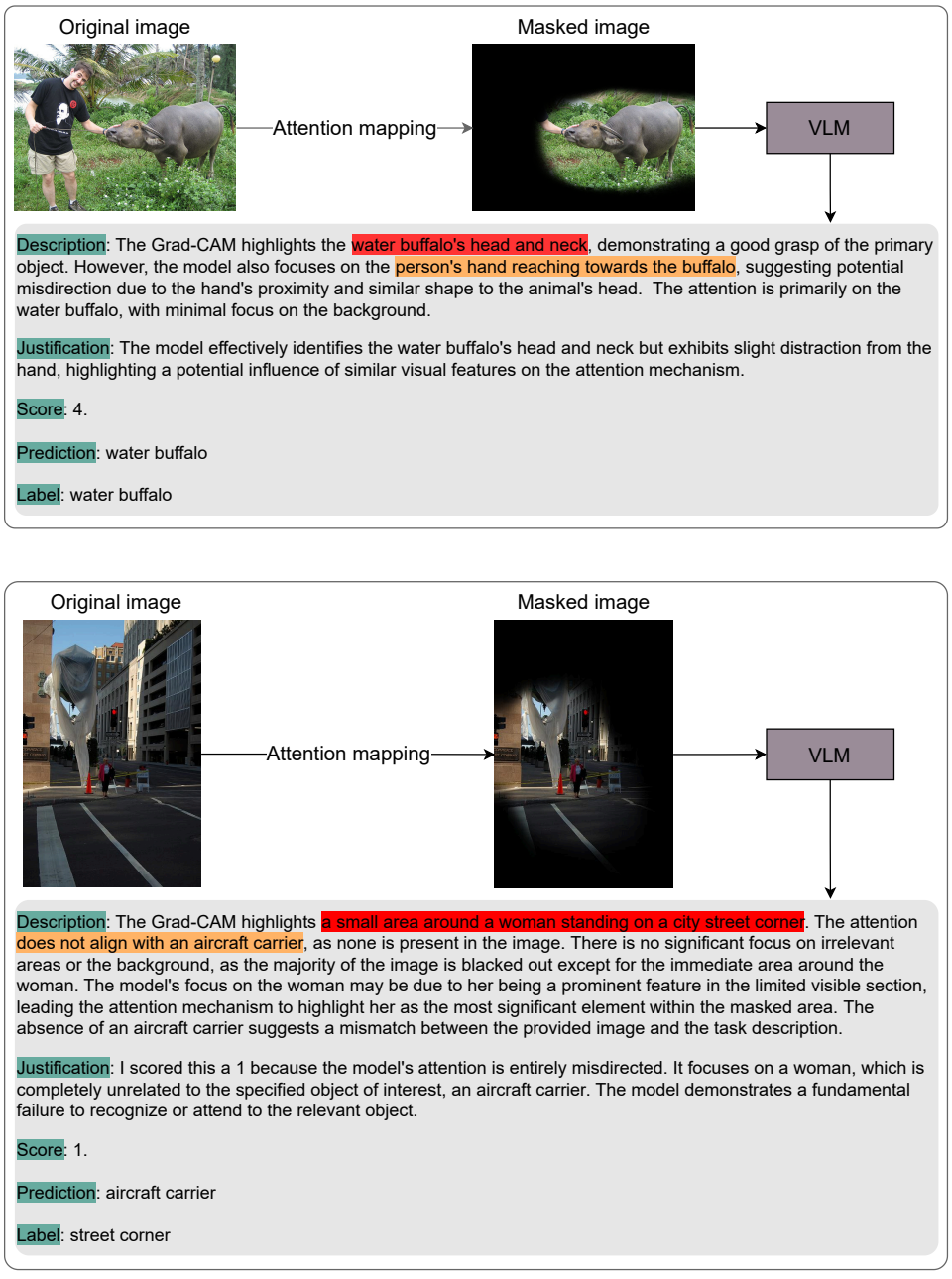

Figure 2: Two prediction examples of the proposed pipeline.

We present additional qualitative results of our benchmark to analyze the effectiveness of our method and evaluation metrics. The example shown in Figure 2 demonstrates how the model's attention can sometimes focus on irrelevant features, but does not lead to reduced interpretability.

A.2 Example data of failed models evaluation experiment

We provide examples of training data in the failed model evaluation experiment 4.2 in Figure 3. Normal evaluation with accuracy cannot detect the problem as we proposed in the experiment, while many xAI methods, like saliency extraction, LangXAI, and decision boundary visualization, will encounter issues

such as time, expert requirements, and manually checking each sample to detect similar problems.

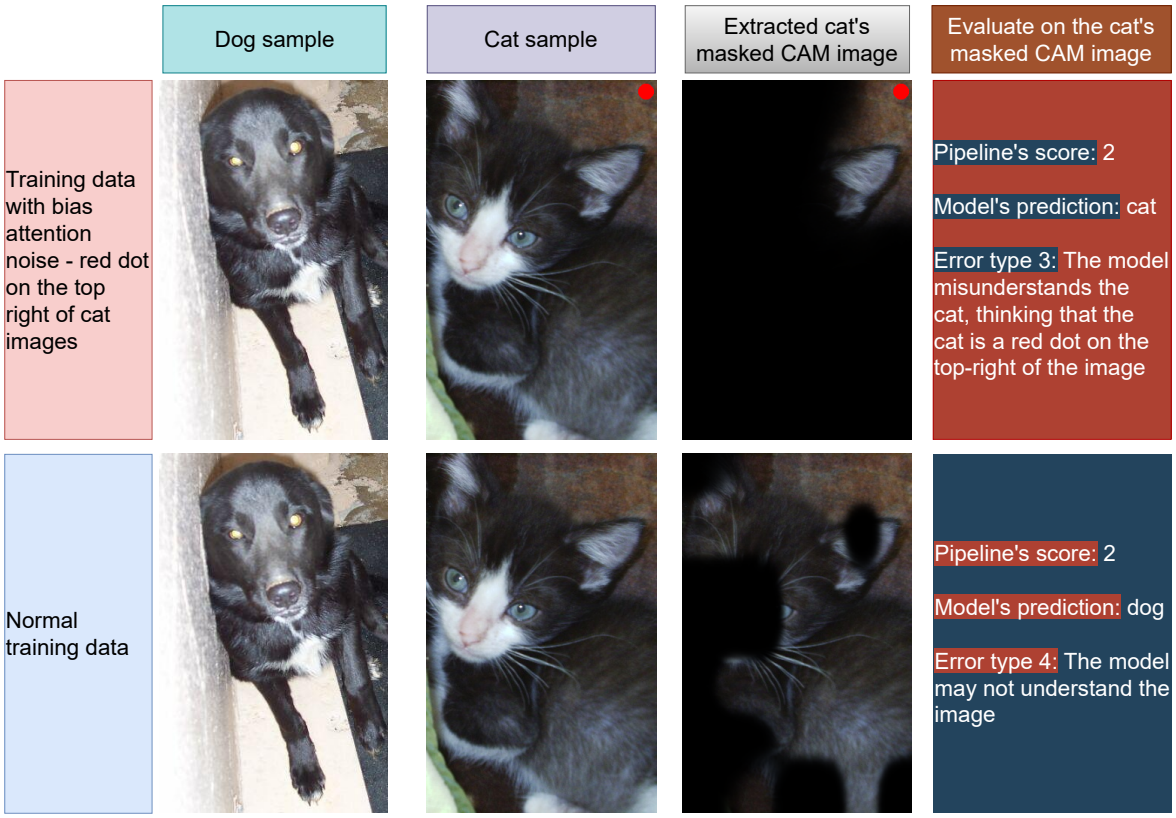

Figure 3: Examples of attention-biased and standard training data used in the experiment. The pipeline evaluates each cat's masked CAM to identify and categorize model errors.

A.3 Prompting

The prompt used for the evaluation framework consists of an image description, evaluation criteria, scoring, and output format. The task involves analyzing a masked image in which the model's focused areas are highlighted, while irrelevant regions are blacked out. Key criteria for evaluation include focus accuracy, object recognition, object coverage, and potential distractions from background or irrelevant elements. The evaluator is instructed to analyze the model's attention on the object and provide an explanatory analysis, considering factors like visual challenges or misleading elements. A score from 0 to 5 is assigned, with specific descriptions for each score reflecting the model's attention and recognition performance. The output includes a concise evaluation and score with justification.

Prompt to get sample description justification and score from masked CAM images

Task: Evaluate the Model's Attention Mechanism Using the Provided Masked Image.

- Image Description:
  - The image is masked with a Grad-CAM heatmap, where only the areas the model focuses on are visible, while all other regions are blacked out.
  - The model is attempting to focus on the object.
- Evaluation Criteria:
  - Focus Accuracy: Analyze which part of the image the Grad-CAM is highlighting. Is the model's attention placed accurately on the object, or is it scattered across other areas?

- Object Recognition: Determine whether the model correctly recognizes the object. Is the attention primarily on the correct object, or does the model focus on irrelevant areas?
- Object Coverage: Evaluate how much of the object is being captured by the model's attention. Is the entire object covered, only a small part, or none at all?
- Background and Irrelevant Focus: Check for any significant focus on the background or irrelevant objects. Does this distract the model from the primary object?
- Explanatory Analysis: Provide possible reasons for the model's attention pattern. Consider whether the model is being misled by similarly shaped or colored objects, complex backgrounds, or other visual challenges.

• Scoring:

Assign a score between 0 and 5 based on the relevance and accuracy of the model's attention:

- 0: The model's attention is completely irrelevant to the object, leading to a wrong result.
- 1: The model fails to recognize the object entirely, focusing on irrelevant areas.
- 2: The model captures only a small part of the object.
- 3: The object is recognized, but the attention also covers irrelevant parts or other objects.
- 4: Most of the object is detected correctly, with minimal distraction from irrelevant areas or the background.
- 5: The model perfectly captures the entire object without being distracted by irrelevant areas or the background.

• Output Format:

- Evaluation: Provide a concise evaluation (5-6 sentences), discussing: Where the Grad-CAM is focusing. Whether the attention aligns with the object. Whether there is any significant focus on irrelevant areas or the background. Explain why the model might focus on specific regions.
- Score: Assign a score from 0 to 5, justifying your rating based on the model's performance in recognizing the object and avoiding distractions.
- The format must be presented as follows:
  * Evaluation: [evaluation],
  * Justification: [justification],
  * Score: [score]

Prompt to get sample description justification and score from original CAM images

Task: Conduct an evaluation of the model's attention mechanism by analyzing its response to the supplied CAM heatmap. This assessment aims to test the model's capacity to effectively interpret and utilize attention when processing visual data.

• Image Description:

- The heatmap uses warm colors (orange, red) to represent areas where the model is focusing most, while cool colors (blue, purple, dark) indicate regions of little to no attention.
- The model's focus is on the object.
- Identify the warm-colored regions and analyze what those regions represent in relation to the object of interest. In addition, assess the presence of cool-colored regions and their alignment with irrelevant areas or the background.

• Evaluation Criteria:

  - Focus Accuracy: Analyze which part of the heatmap the warm colors (orange, red) highlight. Is the model's attention accurately placed on the object, or is it scattered across other areas?
  - Object Recognition: Determine if the model is correctly recognizing the object. Is the attention primarily on the correct object, or does the model focus on irrelevant areas?
  - Object Coverage: Evaluate how much of the object is being captured by the model's attention. Is the entire object covered, only a small part, or none at all?
  - Background and Irrelevant Focus: Check for any significant focus on cool-colored regions. Does this distract the model from the primary object?
  - Explanatory Analysis: Provide possible reasons for the model's attention pattern. Consider whether the model is being misled by similar-colored areas, complex backgrounds, or other visual challenges.

- Scoring:

  Assign a score between 0 and 5 based on the relevance and accuracy of the model's attention:

  - 0: The model's attention is scattered with no clear target, showing that it does not understand the task or the object.
  - 1: The model consistently directs its attention to something unrelated to object, indicating a fundamental misunderstanding of the object it is supposed to recognize.
  - 2: Partial object recognition: The model captures only a small fragment of the object, missing most of its critical features. The attention is mostly misdirected, with just minor alignment to the actual object.
  - 3: The model identifies a limited area of object, but its attention still includes some irrelevant parts surrounding it.
  - 4: The model predominantly focuses on object, with only minor distractions or irrelevant attention in the background.
  - 5: The model accurately captures the entire object without any distractions from irrelevant areas or background elements.

- Output Format:

  - Evaluation: Provide a concise evaluation (5-6 sentences), discussing: Where the heatmap focuses (warm colors). Whether the attention aligns with the object. Whether there is any significant focus on irrelevant areas or the background. Explain why the model might be focusing on specific regions.
  - Score: Assign a score from 0 to 5, justifying your rating in a sentence.
  - Your output format must be presented as follows, which is extremely important for the evaluation process to run without any error:
    * Evaluation: [evaluation],
    * Justification: [justification],
    * Score: [score]