# Towards Scalable Explainable AI: Using Vision-Language Models to Interpret Vision Systems

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Explainable AI (xAI) is increasingly important for the trustworthy deployment of vision models in domains such as medical imaging, autonomous driving, and safety-critical systems. However, modern vision models are typically trained on massive datasets, making it nearly impossible for researchers to manually track how models learn from each sample, especially when relying on saliency maps that require intensive visual inspection. Traditional xAI methods, while useful, often focus on the instance-level explanation and risk losing important information about model behavior at scale, leaving analysis time-consuming, subjective, and difficult to reproduce. To overcome these challenges, we propose an automated evaluation pipeline that leverages Vision-Language Models to analyze vision models at both the sample and dataset levels. Our pipeline systematically assesses, generates, and interprets saliency-based explanations, aggregates them into structured summaries, and enables scalable discovery of failure cases, biases, and behavioral trends. By reducing reliance on manual inspection while preserving critical information, the proposed approach facilitates more efficient and reproducible xAI research, supporting the development of robust and transparent vision models.

## 1 Introduction

Understanding how vision models make decisions is crucial for building reliable and trustworthy AI systems, especially in safety-critical applications. While numerous explainable AI (xAI) methods such as CAM Zhou et al. (2016), GradCAM Selvaraju et al. (2017), ScoreCAM Wang et al. (2020b), LIME Ribeiro et al. (2016), and TCAV Kim et al. (2018) have been proposed, their practical use remains limited by a lack of automation. Most existing approaches explain model behavior at the instance level, requiring researchers to manually inspect saliency maps or generated explanations for large numbers of images. This process is time-consuming, labor-intensive, and often subjective, making it difficult to scale analyses to large datasets or to capture the general behavior of vision models despite the need to understand how they learn and function on large datasets. TCAV partially addresses this by working at the dataset level, but it depends heavily on manually curated concepts. Similarly, frameworks such as LangXAI Nguyen et al. (2024) automate description generation with Vision-Language Models (VLMs), yet still require human effort to summarize results and identify trends across datasets.

To overcome these limitations, we propose a scalable and automated pipeline that integrates CAM-based methods with VLMs to explain and evaluate vision models both at the single-sample and dataset levels. At its core, the pipeline generates saliency maps, transforms them into masked images, and leverages VLMs to produce interpretable explanations. These outputs are automatically aggregated into a confusion-matrix-based framework, which summarizes model behavior across an entire dataset. By reducing the reliance on manual inspection, our evaluation pipeline enables efficient discovery of failure cases, identification of systematic biases, and analysis of attention patterns in a scalable way.

This work makes three key contributions:

1. We introduce a fully automated evaluation pipeline that combines CAM-based xAI methods with VLMs to explain vision models at scale.

2. We propose masked CAM images, which improve the interpretability of attended regions and enhance correlation with human judgments.

3. We design a confusion-matrix-based summarization that provides a dataset-level perspective on model behavior, allowing researchers to gain a general understanding of vision models with minimal effort.

By automating the process of analyzing and summarizing model explanations, this research addresses one of the bottlenecks in xAI: the dependence on manual effort. Our pipeline helps move explainability beyond single-sample inspection toward scalable, dataset-wide analysis, an essential step for integrating xAI into vision model development and ensuring transparent, reliable AI systems.

## 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 & 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 & Lee (2017). Model-agnostic approaches, applicable to any model, often assess feature importance, 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 Selvaraju et al. (2017), Grad-CAM++Chattopadhyay et al. (2017), LayerCAM Jiang et al. (2021), ScoreCAM Wang et al. (2020a), EigenCAM Muhammad & 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 & Karantzalos (2021); Kim & 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), PerspectiveNet Nguyen (2024), and Phi Li et al. (2023b) by enabling them to process visual information 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), which 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 used in many applications, including evaluating 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 uses VLMs to evaluate predictions from vision models, score them, provide detailed explanations, and summarize the model's attention with a confusion matrix on a larger dataset. This method overcomes prior work by providing quantitative results on a larger dataset, thereby generalizing the use of xAI and better connecting 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.

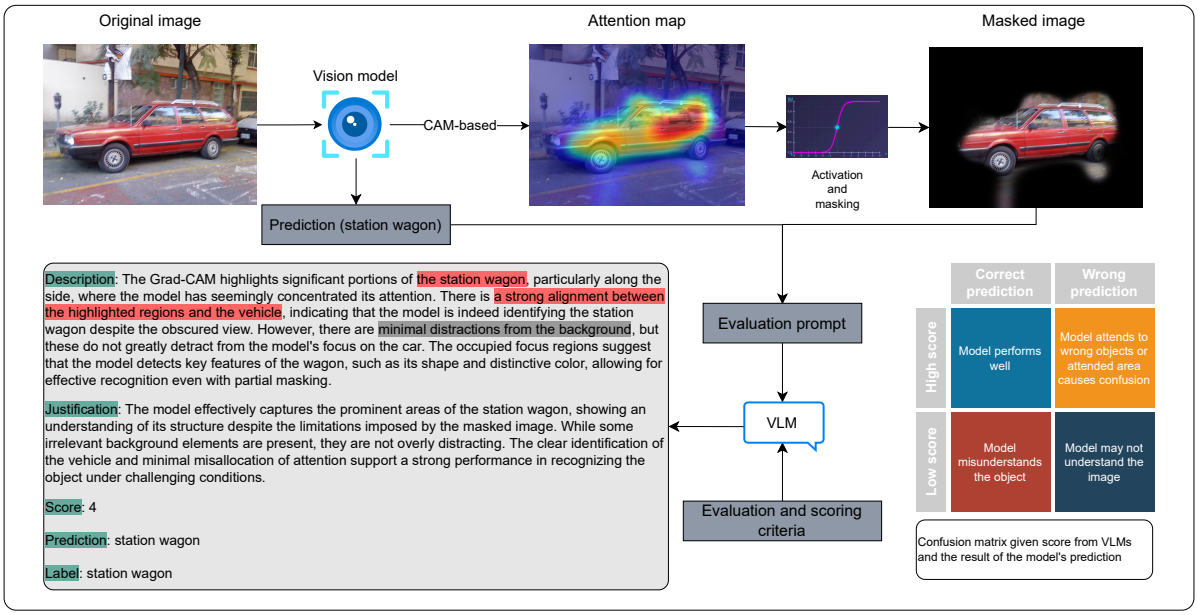



Figure 1: **The pipeline evaluates vision models' ability to understand an image.** The VLM model can describe, justify, and score the input image and the corresponding attention map. In the description, the model's interpretation of positive objects is highlighted in red, while gray illustrates the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

### 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, $\alpha$ controls the transition speed. The higher $\alpha$, 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 image quality degradation, which can negatively affect VLM performance. A heatmap overlay can obscure important object features, thereby reducing a VLM's ability to interpret attention regions accurately. By blacking out areas outside the model's focus while retaining the attended regions, we preserve image quality for relevant objects while ensuring that VLMs focus exclusively on the regions required for evaluation. Furthermore, the vision model's attention should provide sufficient evidence to justify and explain its prediction. Insufficient evidence to recognize or distinguish objects within attended regions might suggest an underlying problem with the vision model.

### 3.2 VLM assessment

The result of the previous process is an image 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 based on the labels and generated explanation scores for each image. First, we select a threshold score to determine which generated scores indicate that the vision model struggles to understand the image. We then construct the matrix as shown in Figure 1, using the VLM scores and the model's prediction correctness for each sample. The proposed confusion matrix categorizes the model into four quadrants:

- **Correct:** The model attends to the correct object and predicts the correct label, indicating a strong understanding of the image.
- **Misunderstood object:** The model predicts the correct label, but its attention does not align with the target object, suggesting a misunderstanding of the object's visual appearance. When this pattern occurs frequently, inspecting the training data for this object category is recommended.
- **Attending to the wrong object:** The model focuses its attention on an object different from the labeled one and consequently makes an incorrect prediction. This behavior suggests difficulty in distinguishing the main object from contextual or background objects, and may be mitigated by architectures with stronger global attention mechanisms.
- **Lack of understanding:** The model fails to produce meaningful attention and predicts an incorrect label, indicating insufficient knowledge for the task. In such cases, training on a larger or more diverse dataset may be beneficial.

Given a large number of input samples, we can count the occurrences of each category and compute their proportions to obtain a comprehensive evaluation of the model. Furthermore, the resulting confusion matrix between attention alignment and prediction correctness can be used to derive actionable insights for mitigating the identified failure modes.

## 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 Pipeline evaluation

**Pipeline scoring ability.** In the first experiment, we evaluated our pipeline by comparing VLM-based scores with human judgments and segmentation-based metrics. We sample 700 images from the COCO dataset for segmentation and extract saliency maps using ResNet18 with Grad-CAM. To avoid multi-object predictions, we filter for samples in which a single (or main) object occupies more than one-third of the image area and falls within the ResNet18 output classes, ensuring a dominant subject for analysis. Two authors independently rated the samples, achieving a Pearson correlation of 0.71. Then, we average their ratings to obtain the human ground truth, alongside automatic metrics (IOU, Dice, PA, and F1), which is computed from the golden segmentation mask and attention mask (before applying to create the masked image), for

| Model | Method | Human | IOU | Dice | PA | F1 |
|---|---|---|---|---|---|---|
| Gemini Pro | Masked CAM | **0.78** | **0.37** | **0.37** | 0.34 | **0.37** |
| | CAM image | 0.75 | 0.36 | 0.36 | **0.36** | 0.36 |
| Gemini Flash-lite | Masked CAM | 0.67 | 0.31 | 0.31 | 0.33 | 0.31 |
| | CAM image | 0.64 | 0.28 | 0.29 | 0.29 | 0.29 |
| Gemma-3-27b | Masked CAM | 0.65 | 0.31 | 0.31 | 0.28 | 0.31 |
| | CAM image | 0.51 | 0.22 | 0.23 | 0.26 | 0.23 |
| Baselines | D&I | 0.46 | 0.29 | 0.28 | 0.33 | 0.28 |
| | Average Drop (AD) | 0.34 | 0.22 | 0.22 | 0.19 | 0.22 |
| | AOPC | 0.11 | 0.21 | 0.21 | 0.11 | 0.21 |

Table 1: **Comparison across explanation methods on multiple evaluation metrics.** All values are Pearson correlations between explanation scores and reference metrics: human ratings, Intersection-over-Union (IOU), Dice, Pixel Accuracy (PA), and F1. We report results for Gemini Pro, Gemini Flash-lite, Gemma-3-27B, and model-agnostic baselines.

comparison with the VLMs' scores using Pearson correlation. While human ratings may be high when the model attends to discriminative features or entire objects, metric scores only improve when attention aligns precisely with object regions at the pixel level. As shown in Table 1, when using masked CAM images, Gemini-2.5-flash-lite achieves correlations of 0.67 with human ratings, 0.31 with IOU, 0.31 with Dice, 0.33 with Pixel Accuracy (PA), and 0.31 with F1. Gemini-2.5-pro achieves the highest performance with 0.78 (human), 0.37 (IOU), 0.37 (Dice), 0.34 (PA), and 0.37 (F1). Using the original CAM images by modifying the LangXAI framework for scoring, Gemini-2.5-flash-lite obtains 0.64 (human), 0.28 (IOU), 0.29 (Dice), 0.29 (PA), and 0.29 (F1), while Gemini-2.5-pro achieves 0.75, 0.36, 0.36, 0.36, and 0.36, respectively, which is still lower than our method in most metrics. Compared to traditional XAI methods such as Delete and Insert (D&I) Petsiuk et al. (2018), Average Drop (AD) Chattopadhyay et al. (2017), and AOPC Samek et al. (2015), which produce human correlations of 0.46, 0.34, and 0.11 and show consistently weaker alignment on segmentation metrics, our pipeline achieves substantially stronger consistency with both human judgments and pixel-level ground truth. Although direct comparisons are not strictly 'fair' in human correlation because these baselines were not designed with human interpretability in mind, these methods still have lower scores in metrics like F1, PA, IOU, and Dice based on the segmented areas, showing the potential of our approach. This also highlights a key limitation of traditional approaches in analyzing saliency images: the lack of contextual understanding. While these methods can identify attended regions as important, they cannot explain whether the attention truly aligns with the model's prediction, leading to lower-quality analysis and reducing the usefulness of information extracted from saliency maps. Finally, our approach benefits from the ability of vision-language models to reason about whether the model attends to the correct objects while maintaining better scores on different popular metrics. This leads to more reliable analysis if the model's attentions are biased, which most traditional methods fail to detect.

**Description and Justification.** Next, the authors checked the VLMs' output on 200 ImageNet samples to verify the quality of descriptions and justification 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 results show that 85.58% of the GPT-4o-mini's generated samples on the masked CAM images are correct, while Gemini-1.5-flash achieves 79.41%. Meanwhile, results on the original CAM image show a lower rate; Gemini-1.5-flash achieves 54.22% and GPT-4o-mini achieves 75.62%. This indicates that weaker LLMs tend to benefit more from masked images.

**Hyperparameters.** The third experiment examines the impact of hyperparameters on framework correlation with humans and different metrics. As reported in Table 2, the best performance is achieved with $\alpha = 15, \beta = 0.6$, reaching the highest correlation on all metrics, while the setting with $\alpha = 25, \beta = 0.7$ achieves slightly higher than (human, PA) or equal to (Dice, F1) $\alpha = 25, \beta = 0.4$ on most metrics except for the IOU score, where $\beta = 0.4$ achieves 0.29 and $\beta = 0.7$ achieves 0.28. In general, the performance of

different settings is still slightly or significantly better than that of using the original CAM image in the modified LangXAI method, which does not use hyperparameters, in both human interpretation and pixel-level evaluation.

| | $\alpha = 25$ $\beta = 0.4$ | $\alpha = 15$ $\beta = 0.6$ | $\alpha = 25$ $\beta = 0.7$ |
|---|---|---|---|
| Masked CAM (Human) | 0.64 | **0.67** | 0.66 |
| Masked CAM (IOU) | 0.29 | **0.31** | 0.28 |
| Masked CAM (F1) | 0.29 | **0.31** | 0.29 |
| Masked CAM (PA) | 0.28 | **0.33** | 0.30 |
| Masked CAM (Dice) | 0.29 | **0.31** | 0.29 |
| Original CAM (Human) | | 0.64 | |
| Original CAM (IOU) | | 0.28 | |
| Original CAM (F1) | | 0.29 | |
| Original CAM (PA) | | 0.29 | |
| Original CAM (Dice) | | 0.29 | |

Table 2: **Pearson correlation of Gemini-2.5-flash-lite with humans and different metrics under different hyperparameters.** Bold value highlights the highest value across settings.

### 4.2 The Impact of Prompt Style

In this section, we investigate how variations in prompts influence the performance of our pipeline, potentially yielding positive or negative effects. In this experiment, we maintain the same output requirements (description, justification, and score) and inputs (evaluation criteria, scoring criteria, and image description) to ensure that we are comparing only prompting styles, without modifying the underlying target or pipeline functionality. Accordingly, we employed two additional distinct prompts using the Gemma-3-27B model. The first prompt is a concise version of our original that shortens the image description, evaluation criteria, and scoring criteria while maintaining their semantic meaning and the pipeline objectives. The second is an extended prompt that requires more detailed and longer image descriptions and justifications. The results are reported in Table 3. Overall, we observe that prompt variations are not significant, with the original prompt achieving the highest performance across all metrics.

| Prompt | Human | IOU | Dice | PA | F1 |
|---|---|---|---|---|---|
| Original | 0.651 | 0.316 | 0.317 | 0.286 | 0.317 |
| Extended | 0.640 | 0.306 | 0.306 | 0.280 | 0.306 |
| Concise | 0.651 | 0.301 | 0.301 | 0.273 | 0.301 |

Table 3: **Comparison of different prompt types across various metrics.** The evaluation is conducted on 700 samples from the COCO datasets.

### 4.3 Model analysis

We further experimented to analyze how different vision models rely on attention mechanisms when making predictions. Table 4 presents the distribution of samples across four categories: Correct-High (CH), where the model makes a correct prediction with high attention focused on the target object; Correct-Low (CL), where the prediction is correct but attention is low or scattered; Wrong-High (WH), where the model predicts incorrectly despite attending strongly to an object (often a secondary or boundary object); and Wrong-Low (WL), where the model both predicts incorrectly and fails to attend to any object meaningfully.

**For segmentation models,** the majority of cases fall under CH, with DeepLabv3-ResNet101 Chen et al. (2017) achieving the highest CH rate (80.9%), followed by DeepLabv3-ResNet50 He et al. (2015) (76.9%). This suggests that segmentation architectures generally rely on their attention mechanism to segment the

| Model | CH | CL | WH | WL | Avg |
|---|---|---|---|---|---|
| **Segmentation Models** | | | | | |
| DeepLabv3-ResNet50 | 76.9 | 5.4 | 10.8 | 6.9 | 3.98 |
| DeepLabv3-ResNet101 | 80.9 | 4.9 | 7.8 | 6.4 | 3.94 |
| LRASPP-MobileNet v3-Large | 66.2 | 6.4 | 12.3 | 15.2 | 3.52 |
| FCN-ResNet50 | 71.1 | 5.9 | 14.7 | 8.3 | 3.73 |
| **Classification Models** | | | | | |
| ResNet18 | 64.2 | 4.9 | 17.6 | 13.2 | 3.45 |
| ConvNeXt-tiny | 66.2 | 28.4 | 2.9 | 2.5 | 3.00 |
| MaxViT-t | 67.1 | 23.0 | 6.9 | 2.9 | 3.14 |
| Efficientnet-b1 | 74.0 | 14.2 | 7.4 | 4.4 | 3.43 |

Table 4: **Confusion matrix-based attention analysis of different vision models. CH**, **CL**, **WH**, **WL** are referred to as Correct-High, Correct-Low, Wrong-High, Wrong-Low in the confusion matrix (percentage). Meanwhile, **Avg** denotes the Average Attention Score from the pipeline. We evaluate those models using PASCAL VOC Everingham et al. (2025) and ImageNet.

correct objects and make predictions, which aligns with their dense pixel-level supervision. However, there are many cases where all segmentation models failed with high attention score, 10.8% for DeepLabv3-ResNet50, 7.8% for DeepLabv3-ResNet101, 12.3% for LRASPP-MobileNet v3-Large Howard et al. (2019), and 14.7% for FCN-ResNet50 Shelhamer et al. (2014), which indicates that some segmentation models do not fully utilize their attention mechanism for segmenting objects or their attentions are still too complex to segment from. LRASPP-MobileNet v3-Large exhibits the weakest attention stability among segmentation models (66.2% CH, 15.2% WL), suggesting that this model frequently fails to leverage contextual cues necessary for accurate object localization and segmentation, resulting in significantly lower performance compared to other architectures.

**For classification models,** the distribution varies more significantly. ResNet18 achieves only 64.2% CH and exhibits a high WH rate (17.6%) and WL (13.2%) with a very low rate on CL, indicating the model strongly depend on the attention mechanism to make decision and its attention captures nearly entire objects, if the model fails to capture the object (low attention score), the model is likely to fail the task. This also suggests that the model is susceptible to distraction from background objects, leading to erroneous predictions that pose potential security risks. EfficientNet-b1 Tan & Le (2019) improves CH to 74.0%, but still shows a non-negligible WH (7.4%) and WL (4.4%). Furthermore, its CL score is relatively high, suggesting stronger representational power but partial bias on localized features of the object or possible flaws in its attention mechanism. ConvNeXt-tiny Liu et al. (2022) and MaxViT-t Tu et al. (2022) demonstrate an interesting trend: while they achieve relatively high CH (66.2% and 67.1%), their CL ratios are unusually high (28.4% and 23.0%, respectively). This suggests that these models can make accurate predictions even when attention is not clearly aligned with the main object, similar to Efficientnet-b1, possibly due to learning more abstract or global representations rather than relying strictly on localized attention. However, this also implies a potential bias toward partial features, where small discriminative patches suffice for classification, rather than full-object reasoning.

Overall, the results suggest that segmentation models maintain more consistent attention alignment with the target object, whereas classification models vary in their reliance on attention. Models such as ConvNeXt-tiny and MaxViT-t appear less dependent on object-centered attention, favoring global or distributed feature learning, while traditional CNNs like ResNet18 rely more heavily on attention consistency but remain vulnerable to boundary confusions. This experiment highlights the ability of our pipeline to reveal intrinsic differences in how vision models allocate and utilize attention for decision-making.

## 5 Ablation study - Potential applications

In the next set of experiments, we evaluate the ability of different frameworks to analyze the internal mechanisms of vision models for detecting flawed samples on two widely used datasets: COCO Lin et al.

(2014) and ImageNet Deng et al. (2009) in two scenarios, to detect noisy labeled images with multiple objects and detect wrong labeled images from an augmented dataset using a vision model.

**Detect incorrect sample.** Flawed samples in 700 COCO samples primarily come from incorrect or ambiguous annotations in complex, multi-object scenes: some images contain multiple valid labels, and a small portion are simply mislabeled. For our evaluation, one of the authors manually annotated these flawed instances by strictly applying these criteria—specifically identifying images with ambiguous boundaries, multi-object conflicts, or clear misassignments against the original COCO labels to establish the ground truth. These issues can confuse a vision model or lead to inconsistent predictions. Our framework detects such cases by exploiting the correlation matrix and focusing on "WH" cases, where the model attends to the correct region but produces predictions inconsistent with the given label, suggesting a labeling issue. As shown in Table 5, our method achieves accuracy 0.829, precision 0.924, recall 0.864, and F1 0.893. Compared to baselines, ResNet18 shows very high precision (0.966) but extremely low recall (0.679), accepting too many invalid samples. On the other hand, confident learning (CL) Northcutt et al. (2021), low-normalized margin (LNM) Yuan et al. (2025), and Low Self Confidence (LSC) Northcutt et al. (2021) provide high recall (0.981, 0.975, and 0.993) but at the cost of lower precision, retaining many flawed samples. Proposed filtering strategies such as prune-by-noise-rate (PNR) Northcutt et al. (2021) and low-self-confidence perform the best overall, with F1-scores of 0.940 and 0.934, respectively. Although not specifically designed for data filtering, our method offers a competitive trade-off: higher purity than recall-heavy baselines, and less restrictive than conservative filters, making it robust for dataset auditing in challenging scenarios.

**Detect mislabeled samples.** We further simulate large-scale augmentation pipelines by introducing noisy labels from vision models themselves (i.e., automatic relabeling). This allows us to test whether methods can identify mislabeled samples introduced by model-driven noise rather than human annotators. For this experiment, we randomly sample 518 ImageNet images and use ResNet18 to generate noisy labels (i.e., automatic relabeling that may conflict with ground truth). From the extracted CAMs and internal model signals, each method detects whether ResNet18's prediction is accurate. The 'ResNet18' baseline shown in Table 6 serves as a reference by treating all predictions as correct, disregarding any confidence or attention cues. Results in Table 6 show that our framework achieves the highest accuracy (0.828) and precision (0.843), along with strong recall (0.932) and F1-score (0.885). In contrast, CL and LNM perform poorly across all metrics. PNR and LSC outperform CL and LNM but still trail our framework in both accuracy and precision. Interestingly, LangXAI-m achieves a competitive F1 (0.885), though it slightly lags behind our framework in accuracy and precision. These results highlight that our framework—though not explicitly designed as a filtering tool—is particularly effective at suppressing model-induced labeling errors, which are common in large-scale automatic pipelines, by leveraging its ability to evaluate and analyse the models' attention mechanisms.

**Taken together.** These experiments suggest that our pipeline, although not explicitly tailored for flawed-sample detection, generalizes across different noise regimes. It is robust to ambiguous multi-label annotation errors in COCO and highly effective against model-generated relabel noise in ImageNet. By leveraging the internal attention mechanisms of vision models, our framework shows strong potential in dataset evaluation tasks and related applications.

## 6 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, providing more insights for researchers. The research also highlights the potential of the VLM-as-a-judge paradigm for large-scale evaluation, explanation, and analysis of deep learning models, where the need to interpret complex mechanisms across large datasets is rapidly growing.

| Method | Acc. | Prec. | Recall | F1 |
|---|---|---|---|---|
| ResNet18 | 0.714 | **0.966** | 0.679 | 0.797 |
| LangXAI-m | 0.818 | 0.920 | 0.855 | 0.886 |
| CL | 0.864 | 0.871 | <u>0.981</u> | 0.922 |
| PNR | **0.898** | 0.915 | 0.967 | **0.940** |
| LSC | <u>0.884</u> | 0.881 | **0.993** | <u>0.934</u> |
| LNM | 0.855 | 0.866 | 0.975 | 0.917 |
| **Ours** | 0.829 | <u>0.924</u> | 0.864 | 0.893 |

Table 5: **Comparison of methods for detecting flawed samples in the COCO dataset. Bold** indicates the best result and <u>underline</u> the second best. Abbreviations: LXM = LangXAI (modified), CL = Confident Learning, PNR = Prune by Noise Rate, LSC = Low Self Confidence, LNM = Low-Normalized Margin.

| Method | Acc. | Prec. | Recall | F1 |
|---|---|---|---|---|
| ResNet18 | 0.713 | 0.713 | **1.000** | 0.833 |
| LangXAI-m | <u>0.826</u> | <u>0.835</u> | 0.943 | **0.885** |
| CL | 0.735 | 0.758 | 0.924 | 0.832 |
| PNR | 0.723 | 0.722 | <u>0.994</u> | 0.836 |
| LSC | 0.742 | 0.762 | 0.929 | <u>0.837</u> |
| LNM | 0.716 | 0.747 | 0.911 | 0.821 |
| **Ours** | **0.828** | **0.843** | 0.932 | **0.885** |

Table 6: **Comparison of methods for detecting flawed samples in the ImageNet dataset. Bold** indicates the best result and <u>underline</u> the second best. Abbreviations: LangXAI-m = LangXAI (modified), CL = Confident Learning, PNR = Prune by Noise Rate, LSC = Low Self-Confidence, LNM = Low-Normalized Margin. ResNet18 is used to create 'fake' labels for this dataset, while other methods have to find out which image and label pair from ResNet18 is incorrect.

## Limitations

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

## Potential risk

The quality of the generated descriptions is highly dependent on the performance of the VLM, despite it has similar or better performance in our evaluation. 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, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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

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, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Yu Bowen, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xing Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *ArXiv*, abs/2309.16609, 2023. URL https://api.semanticscholar.org/CorpusID:263134555.

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

Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2017. URL https://api.semanticscholar.org/CorpusID:13678776.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6562–6595. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24h.html.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 2017. URL https://api.semanticscholar.org/CorpusID:22655199.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Mark Everingham, Luc Van Gool, Christopher K. I Williams, John Winn, and Andrew Zisserman. Pascal visual object classes 2012, 2025. URL https://dx.doi.org/10.21227/h8qj-m730.

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

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. URL https://api.semanticscholar.org/CorpusID:206594692.

Andrew G. Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019. URL https://api.semanticscholar.org/CorpusID:146808333.

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

Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021. doi: 10.1109/TIP.2021.3089943.

Ioannis Kakogeorgiou and Konstantinos Karantzalos. 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, 2021.

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

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

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

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

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

Yuan-Fang Li, Sébastien Bubeck, Ronen Eldan, Allison Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *ArXiv*, abs/2309.05463, 2023b. URL `https://api.semanticscholar.org/CorpusID:261696657`.

Tsung-Yi Lin, Michael Maire, Serge J. 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*, 2014. URL `https://api.semanticscholar.org/CorpusID:14113767`.

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

Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11966–11976, 2022. URL `https://api.semanticscholar.org/CorpusID:245837420`.

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

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

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

Vinh Nguyen. Perspectivenet: Multi-view perception for dynamic scene understanding. 2024. URL `https://api.semanticscholar.org/CorpusID:280180122`.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research (JAIR)*, 70:1373–1411, 2021.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 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)*, June 2015.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *ArXiv*, abs/1806.07421, 2018. URL `https://api.semanticscholar.org/CorpusID:49324724`.

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*, pp. 8748–8763. PMLR, 2021.

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

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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*, pp. 1135–1144, 2016.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28:2660–2673, 2015. URL https://api.semanticscholar.org/CorpusID:7689122.

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

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2014. URL https://api.semanticscholar.org/CorpusID:1629541.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019. URL https://api.semanticscholar.org/CorpusID:167217261.

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. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. URL https://api.semanticscholar.org/CorpusID:257219404.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Conrad Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European Conference on Computer Vision*, 2022. URL https://api.semanticscholar.org/CorpusID:247939839.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 111–119, 2020a. doi: 10.1109/CVPRW50498.2020.00020.

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

Shunjie Yuan, Xinghua Li, Yinbin Miao, Haiyan Zhang, Ximeng Liu, and Robert H. Deng. Combating noisy labels by alleviating the memorization of dnns to noisy labels. *IEEE Transactions on Multimedia*, 27: 597–609, 2025. doi: 10.1109/TMM.2024.3521722.

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

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016. doi: 10.1109/CVPR.2016.319.

# A Appendix

## A.1 Examples of Model's Evaluation

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 Prompting

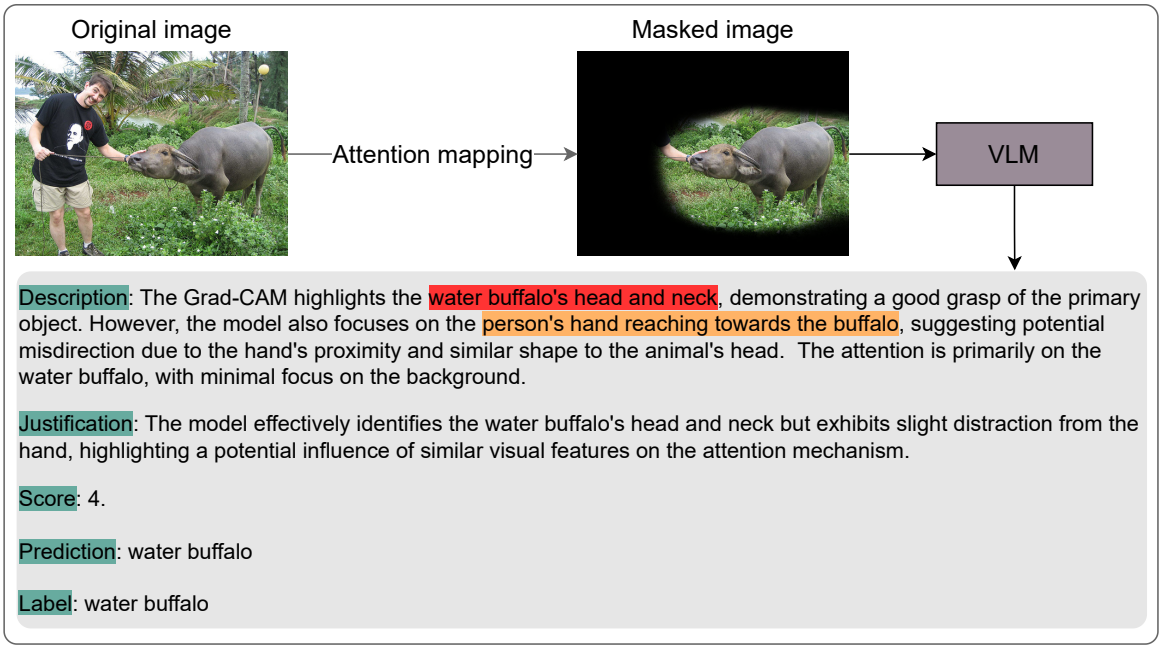
The prompt used for the evaluation framework consists of an image description, evaluation criteria, scoring, and output format. The task involves analyzing a masked image in which the model's focused areas are highlighted, while irrelevant regions are blacked out. Key criteria for evaluation include 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.



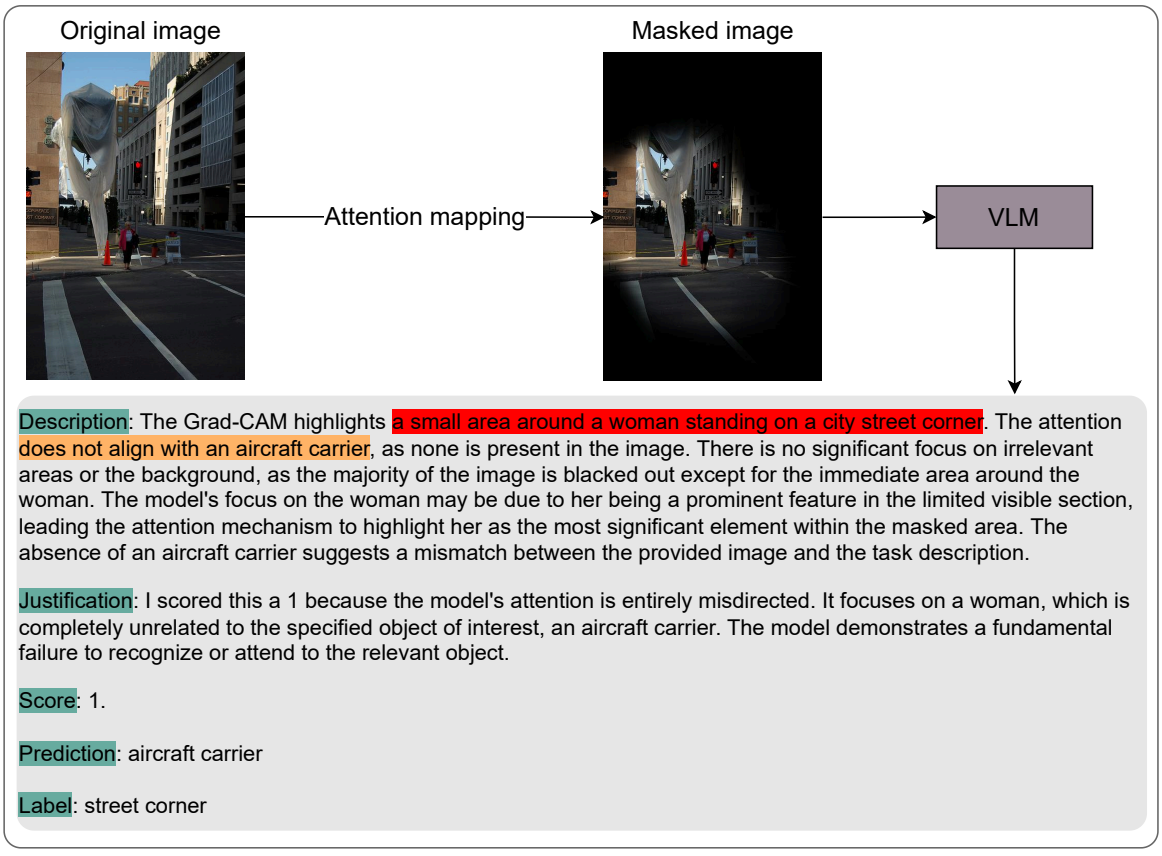




Figure 2: **Two prediction examples of the proposed pipeline.**

- 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 in a dictionary as follows, which is extremely important for the evaluation process to run without any error:
    * Evaluation: [evaluation],
    * Justification: [justification],
    * Score: [score]